



Universidade de Coimbra
Faculdade de Ciências e Tecnologia

Recuperação de falhas em redes MPLS - uma proposta

Luísa Maria Garcia Jorge

Coimbra
2007

Faculdade de Ciências e Tecnologia da Universidade de Coimbra

Recuperação de falhas em redes MPLS

Uma proposta

Luísa Maria Garcia Jorge

Dissertação apresentada à Faculdade de Ciências e Tecnologia da Universidade de Coimbra para obtenção do grau de Doutor em Engenharia Electrotécnica, especialidade de Telecomunicações e Electrónica sob orientação da Doutora Teresa Martinez dos Santos Gomes

Coimbra – Portugal
2007

Abstract

The consequences of service interruption can be severe due to the number of services and applications which depend of communication networks. A failure does not have the same impact in all types of traffic. Real time applications are usually more affected because they are unable to recover using retransmission. In fact communication networks are not immune to failure however great effort has been made in order to increase their dependability.

Routing mechanisms with teletraffic engineering using *MultiProtocol Label Switching* (MPLS) has been growing and has spurred the development of MPLS based recovery mechanisms. Recently many recovery schemes have been proposed for MPLS networks, which take into account recovery time, protection bandwidth, and also the required additional state information. This dissertation is focused on MPLS based recovery schemes.

A survey of protection schemes and their characteristics was made in the context of MPLS networks. A relevant issue of most of these schemes is how to effectively achieve a high degree of protection bandwidth sharing. In this context a proposal for implementing minimal allocation of protection bandwidth, was made. Routing and protection are interleaved and in this work three algorithms are described which can be used to obtain active and recovery paths. The first algorithm is an improved version for an on-line routing algorithm of locally protected paths, the second an algorithm for obtaining the minimal cost pair of disjoint paths with dual arc costs, and the third algorithm obtains the set of all such disjoint path pairs.

The main contribution of this work is a proposal of a new recovery system based on a multi-scheme approach which uses, in an articulate manner, different recovery schemes. The goal of this proposal is to improve the performance of resilient communication networks, particularly regarding differentiated recovery according to traffic classes.

The proposed recovery system was tested using simulation. For this purpose a simulator was build using OMNeT++. This simulation platform was built modularly, having in mind its adaptability and flexibility. It allows several configurations of considered schemes and also the selection method and underlying rules. Some simulation tests were performed and show the pertinence of the proposed recovery system.

Resumo

O número de serviços e aplicações que actualmente dependem das redes de comunicações é bastante elevado, pelo que as consequências de uma interrupção de serviço podem ser drásticas. Os efeitos de uma falha não se distribuem uniformemente por todos os tipos de tráfego, afectando principalmente certas aplicações, tais como as aplicações em tempo real, que não podem facilmente recuperar o tráfego perdido através da sua retransmissão. Como na realidade as redes de comunicações não são imunes a falhas, os esforços para conseguir redes com uma elevada confiabilidade têm sido consistentemente elevados.

Mecanismos de encaminhamento de tráfego através de Engenharia de Tráfego usando MPLS têm tido bastante aceitação, o que tem fomentado o desenvolvimento da recuperação baseada no MPLS. Nos últimos anos têm sido propostos vários esquemas de recuperação de falhas em redes MPLS considerando vários aspectos incluindo, nomeadamente, o tempo de recuperação, a largura de banda adicional necessária para a recuperação e a informação de estado adicional requerida. Esta dissertação enquadra-se nesta temática, pretendendo apresentar uma proposta para a recuperação em redes MPLS que tenha em consideração as características do tráfego a transportar bem como as características dos esquemas a utilizar para recuperar este tráfego.

Foi feita uma resenha comparativa de esquemas de recuperação em redes MPLS, em função das suas características mais relevantes. Alguns problemas em aberto no respeitante a estes esquemas, incluem mecanismos de partilha de largura de banda, pelo que é apresentada uma proposta de reserva mínima de largura de banda na protecção local, utilizando partilha de largura de banda. Nesta dissertação são apresentados estudos sobre três algoritmos usados na determinação dos caminhos activos e de recuperação. O primeiro é uma versão melhorada de um algoritmo de encaminhamento para protecção local com reserva exacta, o segundo é um algoritmo para obtenção de um par de caminhos disjuntos de custo mínimo numa rede com custos duais nos arcos e o terceiro calcula o conjunto de todos esses pares de custo mínimo.

Finalmente, e como contribuição principal foi proposto um novo sistema de recuperação baseado numa metodologia de recuperação multi-esquema que consiste em aplicar esquemas de recuperação diferentes em função de vários critérios. Com este novo sistema pretende-se melhorar o desempenho de redes de comunicações resilientes, nomeadamente na recuperação diferenciada dos vários serviços presentes na rede usando para tal engenharia de tráfego, tendo em consideração as várias classes de serviço.

Este sistema foi implementado e sujeito a testes mediante a construção de um simulador no ambiente simulacional OMNeT++. Este simulador foi construído de raiz, sendo uma ferramenta adaptável que permite diversas configurações e

adaptações em termos de esquemas a usar e no métodos de selecção de esquemas por condições. Alguns testes foram realizados com o simulador, quer como prova de conceito quer como mecanismo de teste de um conjunto de hipóteses, e os seus resultados são aqui apresentados.

Índice

| | |
|---|-------------|
| Abstract | iii |
| Resumo | v |
| Lista de Figuras | xi |
| Lista de Tabelas | xvii |
| Agradecimentos | xxv |
| 1 Introdução | 1 |
| 1.1 Enquadramento | 1 |
| 1.2 Objectivos e contribuições do trabalho | 3 |
| 1.3 Organização do documento | 4 |
| 2 Recuperação em MPLS | 7 |
| 2.1 Conceitos básicos da recuperação | 7 |
| 2.2 Princípios do funcionamento do MPLS | 9 |
| 2.3 Recuperação em redes MPLS | 11 |
| 2.3.1 Princípios de recuperação no MPLS | 13 |
| 2.3.2 Ciclos do processo de recuperação do MPLS | 24 |
| 2.3.3 Critérios de comparação de esquemas de recuperação | 29 |
| 2.3.4 Os vários significados de recuperação global e local - clarificando a notação | 31 |
| 2.3.5 A qualidade de serviço e a engenharia de tráfego no MPLS | 32 |
| 2.4 Esquemas de recuperação no MPLS | 39 |
| 2.4.1 Protecção por comutação | 40 |
| 2.4.2 Reencaminhamento | 61 |
| 2.4.3 Conclusões gerais acerca dos esquemas analisados | 67 |
| 2.5 Conclusões | 74 |
| 3 Partilha de Largura de Banda de Protecção | 75 |
| 3.1 Considerações sobre a partilha de LB de protecção | 75 |
| 3.1.1 Notação | 76 |
| 3.1.2 Um método para reserva mínima de LBP na protecção local | 78 |
| 3.2 Um algoritmo para protecção local com reserva exacta | 79 |
| 3.2.1 Visão geral do algoritmo em Kodialam e Lakshman (2001) | 80 |
| 3.2.2 Algoritmos original e melhorado | 82 |
| 3.2.3 Ilustração da pertinência das modificações propostas | 88 |

| | | |
|----------|--|------------|
| 3.2.4 | Resultados Simulacionais | 105 |
| 3.3 | Caminhos disjuntos de custo mínimo com custos duais nos arcos | 110 |
| 3.3.1 | Partilha de Largura de Banda na Protecção Global | 110 |
| 3.3.2 | Um algoritmo para obter um par de caminhos disjuntos de custo mínimo com custos duais nos arcos | 112 |
| 3.3.3 | Um algoritmo para obter o conjunto de pares de caminhos disjuntos de custo mínimo com custos duais nos arcos | 116 |
| 3.4 | Conclusões | 121 |
| 4 | Um Novo Sistema de Recuperação | 123 |
| 4.1 | Estrutura do sistema | 123 |
| 4.2 | Um exemplo de aplicação | 128 |
| 4.3 | Conclusões | 133 |
| 5 | Aspectos da Implementação | 135 |
| 5.1 | Breve descrição do ambiente de simulação utilizado – OMNeT++ | 135 |
| 5.2 | Descrição geral do funcionamento do programa | 137 |
| 5.2.1 | Descrição da topologia e ficheiros NED | 137 |
| 5.2.2 | Descrição da saídas e ficheiros que as suportam | 138 |
| 5.2.3 | Descrição dos ficheiros de configuração | 145 |
| 5.3 | Funcionamento da simulação por troca de mensagens | 147 |
| 5.4 | Projecto do programa de simulação | 152 |
| 5.4.1 | Componentes nucleares (módulo base) | 159 |
| 5.4.2 | Suporte para múltiplos esquemas de recuperação | 160 |
| 5.4.3 | Suporte para a escolha dinâmica do esquema de recuperação | 162 |
| 5.4.4 | Estruturas para armazenamento de Informação | 164 |
| 5.4.5 | Estruturas para recolha de dados e estruturas auxiliares | 164 |
| 5.5 | Opções tomadas no novo sistema de recuperação | 167 |
| 6 | Análise Experimental | 171 |
| 6.1 | Topologias das redes | 171 |
| 6.2 | Características gerais às várias experiências | 173 |
| 6.3 | Medidas de desempenho e Intervalos de Confiança | 176 |
| 6.4 | Pressupostos das experiências | 177 |
| 6.4.1 | Estratégia para a construção de tabelas | 177 |
| 6.4.2 | Hipóteses a testar | 178 |
| 6.4.3 | Objectivos | 178 |
| 6.5 | Organização das experiências | 180 |
| 6.5.1 | Construção de uma tabela “mono-esquema” | 181 |
| 6.5.2 | Construção de uma tabela de esquema por CT | 189 |
| 6.5.3 | Construção de uma tabela por comprimento esperado de CA | 195 |
| 6.5.4 | Construção de uma tabela por carga da rede | 201 |
| 6.5.5 | Várias tabelas e combinação de condições | 212 |
| 6.6 | Análise final | 221 |

| | | |
|----------|--|------------|
| 7 | Conclusões | 223 |
| 7.1 | Síntese | 223 |
| 7.2 | Principais resultados do trabalho realizado | 224 |
| 7.3 | Perspectivas de trabalho futuro | 225 |
| A | Estrutura do programa de simulação | 229 |
| A.1 | Estruturas para armazenamento de dados | 229 |
| A.1.1 | Informação dos Nós da rede e dos Arcos da rede | 229 |
| A.1.2 | Informação sobre Caminhos | 230 |
| A.1.3 | Informação dos pedidos | 235 |
| A.1.4 | Informação da rede e do seu estado | 235 |
| A.2 | Componentes nucleares | 235 |
| A.2.1 | Implementação da máquina de estados | 236 |
| A.2.2 | Componentes usados na implementação dos BCM | 239 |
| A.2.3 | Estatísticas | 240 |
| A.3 | Componentes específicos do novo sistema de recuperação | 243 |
| A.4 | Componentes associados aos esquemas de recuperação | 244 |
| A.5 | Componentes auxiliares | 248 |
| B | O simulador - Ao nível do pacote | 251 |
| B.1 | O INET | 251 |
| B.2 | Alterações/adições no INET | 252 |
| B.3 | Alterações/adições efectuadas no INET no desenvolvimento do nosso modelo | 254 |
| B.3.1 | Alteração interna para permitir a simulação de falhas de ramos | 254 |
| B.3.2 | Alteração para permitir a criação de pedidos RSVP não <i>scripted</i> | 255 |
| B.3.3 | Geração de pedidos | 255 |
| B.3.4 | LSR com tratamento diferenciado por classe | 255 |
| C | Tipos de tabelas usadas nas experiências | 257 |
| D | Resultados adicionais | 263 |
| D.1 | Construção de uma tabela “mono-esquema” - resultados adicionais | 263 |
| D.2 | Construção de uma tabela de esquema por CT – resultados adicionais | 268 |
| D.3 | Construção de uma tabela por comprimento esperado de CA – resultados adicionais | 270 |
| D.4 | Construção de uma tabela por carga da rede – resultados adicionais | 273 |
| D.5 | Várias tabelas e combinação de condições – resultados adicionais . | 275 |
| E | Cálculo de Frequências Variáveis | 281 |
| F | Intervalos de confiança | 283 |
| G | Lista de Acrónimos e Siglas | 287 |

Lista de Figuras

| | | |
|------|--|----|
| 2.1 | Etiquetas e LSP. | 10 |
| 2.2 | Rede que exemplifica alguns dos conceitos básicos. | 12 |
| 2.3 | Exemplo do modelo de reparação global. | 16 |
| 2.4 | Exemplo do modelo de reparação local. | 17 |
| 2.5 | Exemplo de um PPG e de um <i>bypass tunnel</i> | 19 |
| 2.6 | Exemplo de detecção e notificação de falhas. | 22 |
| 2.7 | Modelo do ciclo de recuperação do MPLS. | 26 |
| 2.8 | Modelo do ciclo de reversão do MPLS. | 27 |
| 2.9 | Modelo do ciclo de reencaminhamento dinâmico. | 28 |
| 2.10 | Exemplo do modelo de reparação global. | 31 |
| 2.11 | Diagrama do modelo MAM. | 38 |
| 2.12 | Diagrama do modelo RDM. | 38 |
| 2.13 | Diagrama do modelo MAR. | 39 |
| 2.14 | Diagrama MPLS DS-TE. | 39 |
| 2.15 | Rede com exemplo de caminhos. | 40 |
| 2.16 | Exemplo de uma RNT. | 42 |
| 2.17 | <i>Bypass tunnel</i> para protecção de um ramo. | 44 |
| 2.18 | Exemplo de um " <i>p-cycle</i> " e ramos <i>straddling</i> | 45 |
| 2.19 | Exemplo do mecanismo FSM. | 56 |
| 2.20 | Exemplo do mecanismo PSM. | 57 |
| 2.21 | Exemplo com os vários caminhos de protecção de um LSP no <i>One-to-one backup</i> | 59 |
| 2.22 | Exemplo de caminhos de protecção no <i>One-to-one backup</i> | 60 |
| 2.23 | Exemplo de <i>bypass tunnels</i> na recuperação <i>Facility backup</i> | 61 |
| 2.24 | Diagrama de fluxo do esquema proposto em Yoon <i>et al.</i> (2001). | 63 |
| 2.25 | Exemplo de expansão do âmbito da recuperação. | 65 |
| 2.26 | Rede exemplo com três caminhos de trabalho estabelecidos. | 66 |
| 3.1 | Rede para ilustrar a estrutura usada na reserva exacta. | 79 |
| 3.2 | Estrutura activo-protecção. | 80 |
| 3.3 | Estrutura protecção-activo. | 80 |
| 3.4 | Algoritmo LOCAL_EDGE_DISJOINT (s, t) (original e modificado). | 83 |
| 3.5 | Algoritmo ALT_PATH_COST (k, j) (original e modificado). | 83 |
| 3.6 | Algoritmo SHORT_PRED_PATH (k, u, j) (original e modificado). | 84 |
| 3.7 | Rede exemplo usada para ilustrar as correcções e alterações propostas ao algoritmo. | 88 |
| 3.8 | Iterações principais do algoritmo LOCAL_EDGE_DISJOINT (s, t) - versão original. | 96 |

| | | |
|------|--|-----|
| 3.9 | Iterações principais do algoritmo LOCAL_EDGE_DISJOINT (s, t) - versão alterada. | 97 |
| 3.10 | Iterações principais do algoritmo LOCAL_EDGE_DISJOINT (s, t) - versão original. | 101 |
| 3.11 | Iterações principais do algoritmo LOCAL_EDGE_DISJOINT (s, t) - versão alterada. | 101 |
| 3.12 | Topologia da rede (adaptada de Kodialam e Lakshman, 2000). | 105 |
| 3.13 | Pedidos aceites, usando tráfego incremental numa rede de capaci- dade infinita. | 106 |
| 3.14 | Pedidos aceites, usando tráfego incremental numa rede de capaci- dade infinita, para melhor visualização do comportamento de LED. | 107 |
| 3.15 | Largura de banda usada por CA e CR, numa rede de capacidade infinita com tráfego incremental. | 107 |
| 3.16 | Probabilidade de rejeição de pedidos, numa rede de capacidade finita com tráfego dinâmico. | 108 |
| 3.17 | Percentagem de soluções sub-óptimas, em rede com soluções sub- óptimas para $m = 4n, 6n$ e gamas $\bar{\mathcal{E}}$ | 114 |
| 3.18 | Tempo médio de CPU por par de nós para $n = 200, 500, 800$ e $m = 3n, 4n, 6n$ | 115 |
| 3.19 | Desvio normalizado máximo associado às soluções sub-óptimas, para $m = 4n, 6n$ e gamas $\bar{\mathcal{Z}}$ | 115 |
| 3.20 | Tempo médio de CPU por par de nós para $n = 100, 200, 400, 800,$ $1600, 3200$ e $m = 10\%, 20\%$ | 117 |
| 3.21 | Percentagem de soluções sub-óptimas para redes com soluções sub-óptimas para $m = 3n$ e $m = 4n$ | 119 |
| 3.22 | Tempo médio de CPU por par de nós (incluindo redes com solu- ções sub-óptimas) para $m = 3n$ e $m = 4n$ | 120 |
| 3.23 | Dimensão dos conjuntos óptimos para gamas $\bar{\mathcal{E}}$ para $m = 3n$ e $m = 4n$ | 121 |
| 4.1 | Diagrama principal. | 124 |
| 5.1 | Fila de acontecimentos futuros - um estado possível. | 148 |
| 5.2 | Diagrama de fluxo da máquina de estados. | 150 |
| 5.3 | O percurso de quatro pedidos. | 152 |
| 5.4 | Diagrama das classes mais significativas definidas na implementa- ção. | 153 |
| 5.5 | Componentes nucleares. | 154 |
| 5.6 | Componentes associados aos esquemas de recuperação. | 154 |
| 5.7 | Componentes específicos do novo sistema. | 155 |
| 5.8 | Estruturas para armazenamento de dados. | 156 |
| 5.9 | Componentes de recolha de dados. | 157 |
| 5.10 | Componentes auxiliares. | 158 |
| 6.1 | Rede KL-15. | 172 |
| 6.2 | Rede COST 239. | 172 |

| | | |
|------|---|-----|
| 6.3 | Bloqueio no estabelecimento para CT1, em função do factor de carga (rede KL-15) - “mono-esquema” versus “multi-esquema”. | 191 |
| 6.4 | Número de ramos do CA para CT2, em função do factor de carga (rede KL-15) - “mono-esquema” versus “multi-esquema”. | 193 |
| 6.5 | Bloqueio no estabelecimento para CT3, em função do factor de carga (rede KL-15) - “mono-esquema” versus “multi-esquema”. | 194 |
| 6.6 | Número de ramos do CA para CT3, em função do factor de carga (rede KL-15) - “mono-esquema” versus “multi-esquema”. | 195 |
| 6.7 | Bloqueio no estabelecimento para CT1 e CT3, em função do factor de carga - “mono-esquema” versus misto por comprimento de CA (PG3-PL3). | 197 |
| 6.8 | Média para todos os ramos da LB gasta, para CT1 e CT3, em função do factor de carga - “mono-esquema” versus misto por comprimento de CA (PG1-PL3). | 198 |
| 6.9 | Média para todos os ramos da LB gasta, por todos os CT, em função do factor de carga - “mono-esquema” versus misto por comprimento de Caminho Activo (<i>Active Path</i>) (CA) (PG1-PL3). | 199 |
| 6.10 | Número de ramos do CA, para CT1 e CT3, em função do factor de carga - “mono-esquema” versus misto por comprimento de CA (PG1-PL1). | 199 |
| 6.11 | Número de CR por pedido, para CT3, em função do factor de carga - “mono-esquema” versus misto por comprimento de CA (PG1-PL3). | 200 |
| 6.12 | Bloqueio no estabelecimento para CT3, em função do factor de carga - “mono-esquema” versus misto por carga da rede (PL3-RL5). | 204 |
| 6.13 | Número de pedidos tentados versus estabelecidos, por esquema, para CT3, em função do factor de carga (rede KL-15) - misto por carga da rede (PL3-RL5). | 205 |
| 6.14 | Bloqueio no estabelecimento para CT3, em função do factor de carga - “mono-esquema” versus misto por carga da rede (PL3-PL2). | 206 |
| 6.15 | Número de ramos do CA, para CT3, em função do factor de carga (rede KL-15) - “mono-esquema” versus misto por carga da rede (PL3-PL2). | 207 |
| 6.16 | Número de pedidos tentados versus estabelecidos, por esquema, para CT3, em função do factor de carga (rede KL-15) - misto por carga da rede (PL3-PL2). | 208 |
| 6.17 | Bloqueio no estabelecimento para CT3, em função do factor de carga - “mono-esquema” versus misto por carga da rede (PL3-PG3). | 209 |
| 6.18 | Número de ramos do CA, para CT3, em função do factor de carga (rede KL-15) - “mono-esquema” versus misto por carga da rede (PL3-PG3). | 210 |
| 6.19 | Número de pedidos tentados versus estabelecidos, por esquema, para <i>Class Type</i> (CT)3, em função do factor de carga (rede KL-15) - misto por carga da rede (PL3-PG3). | 211 |
| 6.20 | Bloqueio no estabelecimento para CT1, em função do factor de carga (rede KL-15) - “mono-esquema” versus exemplo. | 215 |

| | | |
|------|---|-----|
| 6.21 | Desconexão na falha para CT1, em função do factor de carga (rede KL-15) - “mono-esquema” versus exemplo. | 216 |
| 6.22 | Número de ramos do CA para CT2, em função do factor de carga (rede KL-15) - “mono-esquema” versus exemplo. | 217 |
| 6.23 | Bloqueio no estabelecimento para CT2, em função do factor de carga (rede KL-15) - “mono-esquema” versus exemplo. | 218 |
| 6.24 | Bloqueio no estabelecimento para CT3, em função do factor de carga (rede KL-15) - “mono-esquema” versus exemplo. | 219 |
| 6.25 | Número de ramos do CA para CT3, em função do factor de carga (rede KL-15) - “mono-esquema” versus exemplo. | 219 |
| 6.26 | Número de pedidos tentados versus estabelecidos, por esquema, para CT3, em função do factor de carga no exemplo (rede KL-15). | 221 |
| | | |
| A.1 | Diagrama de herança – No e VECNos. | 230 |
| A.2 | Diagrama de herança – ARCO e VECARCOS. | 231 |
| A.3 | Diagrama de herança – CAMINHO. | 232 |
| A.4 | Diagrama de classes – CAMINHOACTIVO. | 233 |
| A.5 | Diagrama de classes – CAMINHOSRECUPERACAO. | 234 |
| A.6 | Diagrama de classes – INFORMACAOREDE, REGISTARESERVAS e RESERVARSVP. | 236 |
| A.7 | Diagrama de classes – componentes nucleares. | 237 |
| A.8 | Diagrama de herança – classe MSGBASE. | 238 |
| A.9 | Diagrama de herança – MODELORESTRICAOARGURABANDA. | 239 |
| A.10 | Diagrama de herança – classe MBSTATBASE. | 242 |
| A.11 | Diagrama de herança – classe IMBASE e derivadas. | 244 |
| A.12 | Diagrama de classes – classe ESQUEMARECUPERACAO e associadas. | 245 |
| | | |
| D.1 | Desconexão na falha para CT1, em função do factor de carga (rede KL-15) – “mono-esquema” versus “multi-esquema”. | 269 |
| D.2 | Bloqueio no estabelecimento para CT1, em função do factor de carga (rede COST 239) – “mono-esquema” versus “multi-esquema”. | 269 |
| D.3 | Número de ramos do CA para CT2, em função do factor de carga (rede COST 239) – “mono-esquema” versus “multi-esquema”. | 270 |
| D.4 | Bloqueio no estabelecimento para CT3, em função do factor de carga (rede COST 239) – “mono-esquema” versus “multi-esquema”. | 271 |
| D.5 | Número de ramos do CA para CT3, em função do factor de carga (rede COST 239) – “mono-esquema” versus “multi-esquema”. | 271 |
| D.6 | Bloqueio na falha para CT1, em função do factor de carga (rede COST 239) – “mono-esquema” versus “multi-esquema”. | 272 |
| D.7 | Bloqueio no estabelecimento para CT1 e CT3 – “mono-esquema” versus misto por comprimento de CA (PG1-PL1). | 272 |
| D.8 | Média para todos os ramos da LB gasta, para CT1 e CT3 – “mono-esquema” versus misto por comprimento de CA (PG3-PL1). | 273 |
| D.9 | Média para todos os ramos da LB gasta, por todos os CT – “mono-esquema” versus misto por comprimento de CA (PG3-PL1). | 273 |

| | |
|---|-----|
| D.10 Bloqueio no estabelecimento para CT1, em função do factor de carga – “mono-esquema” versus misto por carga da rede (PL3-RL5). | 274 |
| D.11 Número de pedidos tentados versus estabelecidos, por esquema, para CT1, em função do factor de carga (rede KL-15) – misto por carga da rede (PL3-RL5). | 274 |
| D.12 Bloqueio no estabelecimento para CT1, em função do factor de carga – “mono-esquema” versus misto por carga da rede (PL3-PL2). | 274 |
| D.13 Número de ramos do CA, para CT1, em função do factor de carga (rede KL-15) – “mono-esquema” versus misto por carga da rede (PL3-PL2). | 275 |
| D.14 Número de pedidos tentados versus estabelecidos, por esquema, para CT1, em função do factor de carga (rede KL-15) – misto por carga da rede (PL3-PL2). | 275 |
| D.15 Bloqueio no estabelecimento para CT1, em função do factor de carga – “mono-esquema” versus misto por carga da rede (PL3-PG3). | 275 |
| D.16 Número de ramos do CA, para CT1, em função do factor de carga (rede KL-15) – “mono-esquema” versus misto por carga da rede (PL3-PG3). | 276 |
| D.17 Número de pedidos tentados versus estabelecidos, por esquema, para CT1, em função do factor de carga (rede KL-15) – misto por carga da rede (PL3-PG3). | 276 |
| D.18 Bloqueio no estabelecimento para CT1, em função do factor de carga (rede COST 239) – “mono-esquema” versus exemplo. | 277 |
| D.19 Bloqueio na falha para CT1, em função do factor de carga (rede COST 239) – “mono-esquema” versus exemplo. | 277 |
| D.20 Número de ramos do CA para CT2, em função do factor de carga (rede COST 239) – “mono-esquema” versus exemplo. | 277 |
| D.21 Bloqueio no estabelecimento para CT2, em função do factor de carga (rede COST 239) – “mono-esquema” versus exemplo. | 278 |
| D.22 Bloqueio no estabelecimento para CT3, em função do factor de carga (rede COST 239) – “mono-esquema” versus exemplo. | 278 |
| D.23 Número de ramos do CA para CT2, em função do factor de carga (rede COST 239) – “mono-esquema” versus exemplo. | 278 |
| D.24 Número de pedidos tentados versus estabelecidos, por esquema, para CT3, em função do factor de carga no exemplo (rede COST 239). | 279 |

Lista de Tabelas

| | | |
|------|--|----|
| 2.1 | Resumo de alguns princípios de recuperação. | 25 |
| 2.2 | Necessidades em termos de QoS. | 33 |
| 2.3 | Exemplo com quatro comportamentos DiffServ distintos. | 37 |
| 2.4 | Âmbito da recuperação em termos topológicos. | 68 |
| 2.5 | Protecção por Comutação. | 72 |
| 2.6 | Recuperação por Reencaminhamento (e pré-qualificado). | 73 |
| 3.1 | Estado da rede antes do estabelecimento de qualquer pedido. | 88 |
| 3.2 | Pedidos considerados no exemplo para a Correção 1. | 89 |
| 3.3 | Estado da rede após o estabelecimento do primeiro pedido. | 89 |
| 3.4 | Valores dos vectores δ e l numa invocação particular de SHORT_PRED_PATH (k, u, j) na versão original do algoritmo. | 90 |
| 3.5 | Valores em vectores λ após a execução da primeira iteração do código identificado como "ITERATIVE STEP" da versão original do algoritmo LOCAL_EDGE_DISJOINT (s, t). | 91 |
| 3.6 | Valores no vector λ^3 após a execução da segunda iteração do código identificado como "ITERATIVE STEP" da versão original do algoritmo LOCAL_EDGE_DISJOINT (s, t). | 91 |
| 3.7 | Valores no vector λ^4 quando LSR4 é etiquetado permanentemente. | 91 |
| 3.8 | Valores no vector λ^0 quando LSR0 é etiquetado permanentemente. | 92 |
| 3.9 | Valores que o vector λ^0 deveria ter quando LSR0 é etiquetado permanentemente. | 92 |
| 3.10 | Estado da rede antes do estabelecimento de qualquer pedido - Correção 5. | 92 |
| 3.11 | Pedidos considerados no exemplo - Correção 5. | 92 |
| 3.12 | Estado da rede após o estabelecimento dos dois primeiros pedidos - Correção 5. | 93 |
| 3.13 | Valores no vector λ^1 após a execução da primeira iteração do código identificado como "ITERATIVE STEP" da versão original do algoritmo LOCAL_EDGE_DISJOINT (s, t). | 93 |
| 3.14 | Pedidos considerados no exemplo para a Alteração 1. | 95 |
| 3.15 | Estado da rede após o estabelecimento do primeiro pedido. | 95 |
| 3.16 | Valores nos vectores λ^1 e λ^2 na 2ª iteração do algoritmo LOCAL_EDGE_DISJOINT (s, t). | 98 |
| 3.17 | Estado da rede após o estabelecimento do segundo pedido - versão original. | 98 |
| 3.18 | Estado da rede após o estabelecimento do segundo pedido - versão alterada. | 98 |

| | | |
|------|---|-----|
| 3.19 | Estado da rede após o estabelecimento do segundo pedido - com escolha de caminho de protecção alternativo. | 99 |
| 3.20 | Pedidos considerados no exemplo para a Alteração 2. | 99 |
| 3.21 | Estado da rede após o estabelecimento do primeiro pedido. | 99 |
| 3.22 | Estado da rede após o estabelecimento do segundo pedido. | 100 |
| 3.23 | Estado da rede após o estabelecimento do terceiro pedido. | 100 |
| 3.24 | Estado da rede após o estabelecimento do quarto pedido. | 100 |
| 3.25 | Estado da rede após o estabelecimento do quinto pedido - versão original. | 103 |
| 3.26 | Estado da rede após o estabelecimento do quinto pedido - versão alterada. | 103 |
| 3.27 | Pedidos considerados no exemplo para a Alteração 3a. | 103 |
| 3.28 | Estado da rede após o estabelecimento do primeiro pedido (ver Tabela 3.27). | 104 |
| 3.29 | Valores dos vectores δ e l numa invocação de SHORT_PRED_PATH (k, u, j) na versão original do algoritmo, no estabelecimento do segundo pedido. | 104 |
| 3.30 | Valores dos vectores δ e l numa invocação de SHORT_PRED_PATH (k, u, j) na versão modificada do algoritmo, no estabelecimento do segundo pedido. | 104 |
| 3.31 | Tráfego dinâmico com capacidade finita na rede. | 109 |
| 3.32 | Símbolos correspondentes a gamas de custos. | 113 |
| | | |
| 4.1 | Esquemas de recuperação. | 125 |
| 4.2 | Classes de serviço suportadas. | 128 |
| 4.3 | #A.10 – De acordo com o tráfego na rede. | 128 |
| 4.4 | #A.20 – De acordo com a garantia de LB e comprimento do LSP activo mínimo. | 129 |
| 4.5 | #A.99 - Regras aplicadas por omissão para a etapa A. | 129 |
| 4.6 | #B.99 - Tabela por omissão para a etapa B. | 130 |
| 4.7 | #C.10 - De acordo com a probabilidade de falha do CA. | 131 |
| 4.8 | #C.99 - De acordo com o comprimento do CA. | 131 |
| 4.9 | #F.10 - Mudanças no esquema de recuperação por reencaminhamento de acordo com a localização da falha. | 132 |
| 4.10 | #F.99 - Tabela por omissão para a etapa F. | 132 |
| | | |
| 5.1 | Saídas do programa de simulação - vectoriais. | 139 |
| 5.2 | Possíveis medidas obtidas a partir do ficheiro de vectores. | 140 |
| 5.3 | Saídas do programa de simulação - escalares. | 143 |
| 5.4 | Parâmetros de entrada comuns a todas as corridas. | 145 |
| 5.5 | Parâmetros de entrada específicos de cada corrida. | 146 |
| | | |
| 6.1 | Valores dos parâmetros de entrada comuns a todas as corridas. | 175 |
| 6.2 | Medidas usadas nas experiências. | 176 |
| 6.3 | Esquemas de recuperação implementados. | 181 |
| 6.4 | Tabela “mono-esquema”. | 182 |

| | | |
|------|---|-----|
| 6.5 | Resumo das características das tabelas “mono-esquema” | 183 |
| 6.6 | Bloqueio no estabelecimento para CT1 e CT3, rede KL-15 e factores de carga entre 1 e 4 - Tabelas “mono-esquema” - Protecção. | 184 |
| 6.7 | Bloqueio no estabelecimento para CT1 e CT3, rede KL-15 e factores de carga entre 2 e 6 - Tabelas “mono-esquema” - Reencaminhamento. | 184 |
| 6.8 | Desconexão na falha para CT1 e CT3, rede KL-15 e factores de carga entre 2 e 5 - Tabelas “mono-esquema”. | 185 |
| 6.9 | Número de ramos dos CA para CT1 e CT3, rede KL-15 e factores de carga entre 1 e 4 - Tabelas “mono-esquema” - Protecção. | 185 |
| 6.10 | Número de ramos dos CA para CT1 e CT3, rede KL-15 e factores de carga entre 1 e 5 - Tabelas “mono-esquema” - Reencaminhamento. | 186 |
| 6.11 | LB média dos CA para CT1 e CT3, rede COST 239 e factores de carga entre 1 e 4 - Tabelas “mono-esquema” - Protecção. | 186 |
| 6.12 | LB média dos CA para CT1 e CT3, rede COST 239 e factores de carga entre 1 e 4 - Tabelas “mono-esquema” - Reencaminhamento. | 187 |
| 6.13 | LB média para os CR de um CA para CT1 e CT3, rede COST 239 e factores de carga entre 1 e 4 - Tabelas “mono-esquema” - Protecção. | 187 |
| 6.14 | LB média para os CR de um CA para CT1 e CT3, rede COST 239 e factores de carga entre 1 e 4 - Tabelas “mono-esquema” - Reencaminhamento. | 188 |
| 6.15 | Classificação dos CT. | 190 |
| 6.16 | Esquemas a usar por cada CT. | 190 |
| 6.17 | Tabela “multi-esquema”. | 190 |
| 6.18 | Resumo das características das tabelas de esquema por CT. | 191 |
| 6.19 | Bloqueio no estabelecimento para CT1, rede KL-15 e factores de carga entre 1 e 4 - “mono-esquema” versus “multi-esquema”. | 192 |
| 6.20 | Número de ramos do CA para CT2, rede KL-15 e factores de carga entre 1 e 4 - “mono-esquema” versus “multi-esquema”. | 193 |
| 6.21 | Bloqueio no estabelecimento para CT3, rede KL-15 e factores de carga entre 1 e 4 - “mono-esquema” versus “multi-esquema”. | 194 |
| 6.22 | Número de ramos do CA para CT3, rede KL-15 e factores de carga entre 1 e 4 - “mono-esquema” versus “multi-esquema”. | 195 |
| 6.23 | Tabela por comprimento esperado de CA. | 196 |
| 6.24 | Resumo das características das tabelas por comprimento esperado de CA. | 196 |
| 6.25 | Bloqueio no estabelecimento para CT1 e CT3, rede COST 239, factores de carga entre 1 e 4 e tipos de tabela TT11, TT14 e TT84 (PG3-PL3). | 197 |
| 6.26 | Bloqueio no estabelecimento para CT1 e CT3, rede KL-15, factores de carga entre 1 e 4 e tipos de tabela TT11, TT14 e TT84 (PG3-PL3). | 198 |
| 6.27 | Média para todos os ramos da LB gasta, por todos os CT, para as redes COST 239 e KL-15, factores de carga entre 1 e 4 e tipos de tabela TT8, TT11 e TT82 (PG1-PL3). | 199 |
| 6.28 | Número de ramos do CA, para CT1 e CT3, rede COST 239, factores de carga entre 1 e 4 e tipos de tabela TT1, TT8 e TT81 (PG1-PL1). | 200 |

| | | |
|------|---|-----|
| 6.29 | Número de ramos do CA, para CT1 e CT3, rede KL-15, factores de carga entre 1 e 4 e tipos de tabela TT1, TT8 e TT81 (PG1-PL1). | 200 |
| 6.30 | Parâmetros de entrada adicionais (ou com valores diferentes) para a experiência desta secção. | 202 |
| 6.31 | Valores para λ_{high} e λ_{low} | 202 |
| 6.32 | Resumo das características das tabelas por carga da rede. | 203 |
| 6.33 | Bloqueio no estabelecimento para CT3, rede COST 239, factores de carga entre 1 e 3 e tipos de tabela TT121, TT131, TT133, TT141, TT151 e TT161 (PL3-RL5). | 204 |
| 6.34 | Bloqueio no estabelecimento para CT3, rede KL-15, factores de carga entre 1 e 3 e tipos de tabela TT121, TT131, TT133, TT141, TT151 e TT161 (PL3-RL5). | 205 |
| 6.35 | Número de pedidos tentados versus estabelecidos, por esquema, para CT3, rede KL-15, factores de carga entre 1 e 3 e tabela TT121. | 205 |
| 6.36 | Número de pedidos tentados versus estabelecidos, por esquema, para CT3, rede KL-15, factores de carga entre 1 e 3 e tabela TT141. | 206 |
| 6.37 | Bloqueio no estabelecimento para CT3, rede COST 239, factores de carga entre 1 e 3 e tipos de tabela TT123, TT131, TT134, TT143, TT153 e TT163 (PL3-PL2). | 206 |
| 6.38 | Bloqueio no estabelecimento para CT3, rede KL-15, factores de carga entre 1 e 3 e tipos de tabela TT123, TT131, TT134, TT143, TT153 e TT163 (PL3-PL2). | 207 |
| 6.39 | Número de ramos do CA, para CT3, rede KL-15, factores de carga entre 1 e 3 e tipos de tabela TT123, TT131, TT134 e TT143 (PL3-PL2). | 207 |
| 6.40 | Número de pedidos tentados versus estabelecidos, por esquema, para CT3, rede KL-15, factores de carga entre 1 e 3 e tabela TT123. | 208 |
| 6.41 | Número de pedidos tentados versus estabelecidos, por esquema, para CT3, rede KL-15, factores de carga entre 1 e 3 e tabela TT143. | 208 |
| 6.42 | Bloqueio no estabelecimento para CT3, rede COST 239, factores de carga entre 1 e 3 e tipos de tabela TT125, TT131, TT132, TT145, TT155 e TT165 (PL3-PG3). | 209 |
| 6.43 | Bloqueio no estabelecimento para CT3, rede KL-15, factores de carga entre 1 e 3 e tipos de tabela TT125, TT131, TT132, TT145, TT155 e TT165 (PL3-PG3). | 210 |
| 6.44 | Número de ramos do CA, para CT3, rede KL-15, factores de carga entre 1 e 3 e tipos de tabela TT125, TT131, TT132 e TT145 (PL3-PG3). | 210 |
| 6.45 | Número de pedidos tentados versus estabelecidos, por esquema, para CT3, rede KL-15, factores de carga entre 1 e 3 e tabela TT125. | 211 |
| 6.46 | Número de pedidos tentados versus estabelecidos, por esquema, para CT3, rede KL-15, factores de carga entre 1 e 3 e tabela TT145. | 211 |
| 6.47 | #A.99 - Regras aplicadas para a etapa A. | 212 |
| 6.48 | #B.10 - De acordo com o comprimento do CA. | 213 |
| 6.49 | #B.20 - De acordo com o modelo de recuperação e carga na rede. | 213 |
| 6.50 | #B.99 - Regras aplicadas por omissão para a etapa B. | 213 |
| 6.51 | Valores para λ_{high} e λ_{low} (exemplo do sistema). | 214 |

| | | |
|------|--|-----|
| 6.52 | Bloqueio no estabelecimento para CT1, rede KL-15, factores de carga entre 1 e 1,3 e tipos de tabela TT131, TT132, TT133, TT134, TT135 e TT201. | 216 |
| 6.53 | Bloqueio no estabelecimento para CT1, rede KL-15, factores de carga entre 1,4 e 2 e tipos de tabela TT131, TT132, TT133, TT134, TT135 e TT201. | 216 |
| 6.54 | Desconexão na falha para CT1, rede KL-15, factores de carga entre 1 e 2 e tipos de tabela TT133, TT135 e TT201. | 217 |
| 6.55 | Número de ramos do CA, para CT2, rede KL-15, factores de carga entre 1 e 2 e tipos de tabela TT131, TT132 e TT201. | 217 |
| 6.56 | Bloqueio no estabelecimento para CT2, rede KL-15, factores de carga entre 1 e 2 e tipos de tabela TT131, TT132 e TT201. | 218 |
| 6.57 | Bloqueio no estabelecimento para CT3, rede KL-15, factores de carga entre 1 e 2 e tipos de tabela TT131, TT132, TT134 e TT201. | 219 |
| 6.58 | Número de ramos do CA, para CT3, rede KL-15, factores de carga entre 1 e 2 e tipos de tabela TT131, TT132, TT134 e TT201. | 220 |
| 6.59 | Número de pedidos tentados versus estabelecidos, por esquema, para CT3, rede KL-15, factores de carga entre 1 e 2 e tipo de tabela TT201. | 221 |
| C.1 | Tabela “mono-esquema” (RL4-PL1). | 257 |
| C.2 | Tabela “mono-esquema” (RL4-PL2). | 257 |
| C.3 | Tabela “mono-esquema” (RL4-PG1). | 257 |
| C.4 | Tabela “mono-esquema” (RL4-RG2). | 257 |
| C.5 | Tabela “mono-esquema” (RL4-RL5). | 257 |
| C.6 | Tabela “mono-esquema” (RG1-PL1). | 258 |
| C.7 | Tabela “mono-esquema” (RG1-PL2). | 258 |
| C.8 | Tabela “mono-esquema” (RG1-PG1). | 258 |
| C.9 | Tabela “mono-esquema” (RG1-RG2). | 258 |
| C.10 | Tabela “mono-esquema” (RG1-RL5). | 258 |
| C.11 | Tabela “mono-esquema” (RL4-PL3). | 258 |
| C.12 | Tabela “mono-esquema” (RG1-PL3). | 258 |
| C.13 | Tabela “mono-esquema” (RL4-PG3). | 258 |
| C.14 | Tabela “mono-esquema” (RG1-PG3). | 258 |
| C.15 | Tabela “multi-esquema” (RL4-RG2-PL3-PL2). | 259 |
| C.16 | Tabela “multi-esquema” (RL4-RG2-PL3-PL3). | 259 |
| C.17 | Tabela “multi-esquema” (RL4-RG2-PG3-PL2). | 259 |
| C.18 | Tabela “multi-esquema” (RL4-RG2-PG3-PL3). | 259 |
| C.19 | Tabela “multi-esquema” (RL4-PL2-PL3-PL2). | 259 |
| C.20 | Tabela “multi-esquema” (RL4-PL2-PL3-PL3). | 260 |
| C.21 | Tabela “multi-esquema” (RL4-PL2-PG3-PL2). | 260 |
| C.22 | Tabela “multi-esquema” (RL4-PL2-PG3-PL3). | 260 |
| C.23 | Tabela “multi-esquema” (RL4-PG1-PL3-PL2). | 260 |
| C.24 | Tabela “multi-esquema” (RL4-PG1-PL3-PL3). | 260 |
| C.25 | Tabela “multi-esquema” (RL4-PG1-PG3-PL2). | 260 |
| C.26 | Tabela “multi-esquema” (RL4-PG1-PG3-PL3). | 260 |

| | | |
|------|---|-----|
| C.27 | Tabela por comprimento esperado de CA (RG1/RL4-PG1/PL1). | 260 |
| C.28 | Tabela por comprimento esperado de CA (RG1/RL4-PG1/PL3). | 261 |
| C.29 | Tabela por comprimento esperado de CA (RG1/RL4-PG3/PL1). | 261 |
| C.30 | Tabela por comprimento esperado de CA (RG1/RL4-PG3/PL3). | 261 |
| C.31 | Tabela por carga da rede (RL4-PL3/RL5). | 261 |
| C.32 | Tabela por carga da rede (RL4-PG3/RL5). | 262 |
| C.33 | Tabela por carga da rede (RL4-PL3/PL2). | 262 |
| C.34 | Tabela por carga da rede (RL4-PG3/PL2). | 262 |
| C.35 | Tabela por carga da rede (RL4-PL3/PG3). | 262 |
| | | |
| D.1 | Bloqueio no estabelecimento para CT1 e CT3, rede COST 239 e factores de carga entre 1 e 4 - Tabelas "mono-esquema" - Protecção. | 263 |
| D.2 | Bloqueio no estabelecimento para CT1 e CT3, rede COST 239 e factores de carga entre 2 e 6 - Tabelas "mono-esquema" - Reencaminhamento. | 264 |
| D.3 | Desconexão na falha para CT1 e CT3, rede COST 239 e factores de carga entre 2 e 5 - Tabelas "mono-esquema" - Reencaminhamento. | 264 |
| D.4 | Número de ramos dos CA para CT1 e CT3, rede COST 239 e factores de carga entre 1 e 4 - Tabelas "mono-esquema" - Protecção. | 264 |
| D.5 | Número de ramos dos CA para CT1 e CT3, rede COST 239 e factores de carga entre 1 e 5 - Tabelas "mono-esquema" - Reencaminhamento. | 265 |
| D.6 | LB média dos CA para CT1 e CT3, rede KL-15 e factores de carga entre 1 e 4 - Tabelas "mono-esquema" - Protecção. | 265 |
| D.7 | LB média dos CA para CT1 e CT3, rede KL-15 e factores de carga entre 1 e 4 - Tabelas "mono-esquema" - Reencaminhamento. | 265 |
| D.8 | LB média para o/s CR de um CA para CT1 e CT3, rede KL-15 e factores de carga entre 1 e 4 - Tabelas "mono-esquema" - Protecção. | 266 |
| D.9 | LB média para o/s CR de um CA para CT1 e CT3, rede KL-15 e factores de carga entre 1 e 4 - Tabelas "mono-esquema" - Reencaminhamento. | 266 |
| D.10 | Bloqueio no estabelecimento para CT1 e CT3, rede KL-15 e factores de carga entre 1 e 4 - Tipos de tabela TT2 (RL4/PL2) e TT7 (RG1/PL2). | 266 |
| D.11 | Bloqueio no estabelecimento para CT1 e CT3, rede KL-15 e factores de carga entre 2 e 5 - Tipos de tabela TT4 (RL4/RG2) e TT9 (RG1/RG2). | 267 |
| D.12 | Desconexão na falha para CT1 e CT3, rede KL-15 e factores de carga entre 2 e 5 - Tipos de tabela TT4 (RL4/RG2) e TT9 (RG1/RG2). | 267 |
| D.13 | Bloqueio no estabelecimento para CT1 e CT2, rede KL-15 e factores de carga entre 1 e 4 - Tipo de tabela TT2 (RL4/PL2). | 268 |
| D.14 | Bloqueio no estabelecimento para CT1 e CT2, rede KL-15 e factores de carga entre 2 e 5 - Tipo de tabela TT4 (RL4/RG2). | 268 |
| D.15 | Desconexão na falha para CT1 e CT2, rede KL-15 e factores de carga entre 2 e 5 - Tipo de tabela TT4 (RL4/RG2). | 268 |

| | |
|---|-----|
| D.16 Bloqueio no estabelecimento para CT1, rede COST 239 e factores de carga entre 1 e 4 – “mono-esquema” versus “multi-esquema”. . . | 269 |
| D.17 Número de ramos do CA para CT2, rede COST 239 e factores de carga entre 1 e 4 – “mono-esquema” versus “multi-esquema”. . . . | 270 |
| D.18 Bloqueio no estabelecimento para CT3, rede COST 239 e factores de carga entre 1 e 4 – “mono-esquema” versus “multi-esquema”. . . | 271 |
| D.19 Número de ramos do CA para CT3, rede COST 239 e factores de carga entre 1 e 4 – “mono-esquema” versus “multi-esquema”. . . . | 272 |

Agradecimentos

Muitas pessoas contribuíram para a elaboração desta obra, pelo que gostaria de assinalar a sua ajuda, ressalvando obviamente que quaisquer defeitos na obra apresentada são devidos unicamente a mim. Em particular, gostaria de agradecer:

- À minha orientadora, Professor Teresa Martinez Gomes, por todo o apoio, disponibilidade e rigor científico com que apoiou o desenvolvimento deste trabalho, e pela sua persistência indispensável para a conclusão trabalho;
- Ao Paulo, por todas as discussões, sugestões e comentários frequentes durante a elaboração do trabalho, e pelo desenvolvimento de ferramentas computacionais auxiliares ao programa de simulação;
- Ao Professor Albano Alves, Presidente do Conselho Directivo da ESTIG (Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Bragança) e colegas do DIC-ESTIG, pelas suas preocupações em proporcionar condições para a realização deste trabalho;
- Ao INESC-Coimbra (Instituto de Engenharia de Sistemas e Computadores de Coimbra), pelas facilidades concedidas para a elaboração deste trabalho;
- Ao Programa PRODEP/2003 (Concurso nº2/5.3/PRODEP/2003 (II Série) - Medida 5/Ação 5.3), que apoiou este trabalho com dispensa de serviço docente;
- E ainda a todos que de alguma forma contribuíram para a elaboração deste trabalho.

1 Introdução

1.1 Enquadramento

Hoje em dia as redes de comunicações são usadas simultaneamente por diversos tipos de tráfego, com diferentes necessidades em termos de largura de banda requerida, da sensibilidade ao atraso na propagação do sinal, da sensibilidade à variação de atraso (*jitter*), do suporte a perdas de pacotes, e potencialmente em muitas outras características. As diferentes características desejadas definem o padrão de *Quality of Service* (QoS) desse serviço e podem ser contratualizadas entre fornecedores do serviço de telecomunicações e os seus consumidores no âmbito de um *Service Level Agreement* (SLA). Uma solução usada actualmente para conseguir suportar simultaneamente diferentes tipos de serviços com diferentes necessidades de QoS passa pela utilização de *MultiProtocol Label Switching* (MPLS) (geralmente conjugado com *Differentiated Services* (DiffServ) e *Traffic Engineering* (TE)). O estudo apresentado aqui vai ter como pressuposto o uso destas tecnologias.

Qualquer recurso usado pelas redes de comunicações pode ser sujeito a falhas; rede fiáveis implicam suporte para recuperação dessas falhas. Esta recuperação pode ser feita pelas camadas superiores que utilizam a rede – por exemplo, ao nível do encaminhamento *Internet Protocol* (IP) – mas o seu tempo de resposta pode ser demasiado longo, da ordem das dezenas de segundos ou minutos (Sharma *et al.*, 2003). Estes tempos são considerados inaceitáveis para oferecer serviços de elevada fiabilidade, que pretendam tempos de recuperação muito curtos (da ordem dos cinquenta milissegundos), embora possam ser aceitáveis para serviços menos exigentes. Mecanismos de recuperação das camadas inferiores (*Synchronous Digital Hierarchy* (SDH)/*Synchronous Optical Network* (SONET) ou *Wavelength Division Multiplexing* (WDM)), integrados na infraestrutura de rede, conseguem fornecer recuperação com tempos muito curtos, mas não diferenciam o tráfego em termos de requisitos de QoS, e como tal podem ser demasiado desperdiçadores de recursos se aplicados a todo o tráfego da rede.

De uma forma simples podemos dizer que existem dois modelos diferentes de recuperação de falhas numa rede: uso de reencaminhamento e uso de protecção.

O modelo de reencaminhamento implica o cálculo de um novo caminho a seguir pelo tráfego perante uma falha na rede, e como tal implica um gasto de tempo adicional aquando da falha para cálculo do novo caminho a usar (para além do tempo de detecção da mesma e da decisão de alterar o caminho). No entanto não exige a ocupação prévia de recursos da rede (mas em contrapartida perante a falha não tem a garantia de poder ser usado um caminho com as características desejadas). O modelo de protecção implica o cálculo à partida de um ou de diversos caminhos alternativos a ser usados perante falha num recurso, e (geralmente) reserva de recursos para o caminho, de forma a poder ser usado quando (ou se) houver necessidade. Desta forma os esquemas de protecção podem geralmente conseguir tempos de recuperação mais curtos mas em contrapartida usam mais recursos. O compromisso entre tempo e recursos usados é um exemplo de vários tipos de compromissos que podemos encontrar quando comparamos diferentes tipos de esquemas de recuperação disponíveis. Em geral não se pode afirmar que exista um esquema que seja sempre o preferível para todas as situações e todos os tipos de serviço. Deve ser notado que embora este trabalho se foque apenas na recuperação, a escolha do encaminhamento é frequentemente também um aspecto importante a considerar na recuperação, trazendo consigo compromissos adicionais. Finalmente, a recuperação das falhas pode ser efectuada em MPLS usando um novo caminho entre a origem e o destino do tráfego de forma a evitar o recurso em falha (designada recuperação de âmbito global) ou simplesmente encontrando um novo troço de caminho que evite o recurso inoperacional (esta última forma de recuperação, que reutiliza parte do caminho anteriormente determinado e pode ser descentralizada, é designada de recuperação de âmbito local).

O uso de mais ou menos recursos para transmitir tráfego não tem sempre o mesmo efeito no desempenho da rede. Nomeadamente, se a rede possuir capacidade não usada em quantidade suficiente, o uso de esquemas de reencaminhamento (que não fornecem garantias de recuperação) pode ainda assim permitir obter um desempenho aceitável, simplesmente porque perante abundância de recursos não existe razão para recusar tráfego. Da mesma forma, na presença de recursos em excesso, o uso de mecanismos que consomem mais Largura de Banda (LB) pode ser justificado, se trouxerem características desejáveis, mesmo para serviços que não o exijam em termos de QoS. Isto significa que em situações de carga leve mecanismos de recuperação muito simples provavelmente terão bons desempenhos (e não existe real necessidade de usar TE), enquanto que em redes com tráfego mais próximo da sua capacidade compromissos terão de ser explicitamente considerados (favorecendo explicitamente certos tipos de

tráfego). Embora as redes sejam normalmente dimensionadas de forma a poder tratar o tráfego usualmente esperado, de preferência sem grande desperdício de recursos, os ciclos gerais de utilização das redes levam a que ao longo do tempo a mesma rede possa ser sujeita a situações de carga variando entre carga leve e sobrecarga.

Em conclusão, parece existir vantagem em integrar na determinação da recuperação a usar na rede conhecimento sobre os requisitos de QoS dos serviços que a utilizam (para fornecer as características desejadas a cada serviço), conhecimento sobre as características dos esquemas de recuperação a usar (para perceber os compromissos associados ao uso de cada esquema de recuperação) e conhecimento adicional sobre o estado da rede (para poder determinar os efeitos reais da escolha na situação actual da rede).

1.2 Objectivos e contribuições do trabalho

Esta dissertação apresenta uma proposta para a recuperação em redes MPLS tendo em consideração as características do tráfego a transportar bem como as características dos esquemas de recuperação e do estado da rede.

É apresentada uma classificação de esquemas de recuperação que apresenta para cada esquema analisado as características consideradas relevantes e a descrição do seu funcionamento.

Mecanismos de recuperação baseados em protecção, assumindo o pressuposto da não coincidência de falhas na rede, podem ser optimizados através da partilha da LB de protecção que usam. O cálculo do caminho usado em situações de funcionamento normal da rede – Caminho Activo (*Active Path*) (CA) – e do(s) caminho(s) a usar em situação de falha – Caminho de Recuperação (*Recovery Path*) (CR) – tem geralmente como objectivo a minimização da LB usada, particularmente quando se pretende fazer partilha de LB de protecção. Nesta dissertação é apresentada uma versão melhorada de um algoritmo de encaminhamento para protecção local com reserva exacta da Largura de Banda de Protecção (LBP). No contexto da protecção global, com partilha de LBP, é necessário resolver o problema da obtenção do par de caminhos disjuntos de custo total mínimo numa rede com custos duais nos arcos. São descritos dois algoritmos para a resolução exacta deste problema: o primeiro obtém apenas um par de caminhos disjuntos de custo mínimo e o outro obtém o conjunto de todos esses pares.

A eficácia da utilização de mecanismos de protecção com LBP depende da divulgação de informação acerca do estado de ocupação dos arcos, da informação que deve ser armazenada em cada nó acerca do estado dos arcos e dos protocolos

de sinalização necessários para os implementar. Tendo como objectivo a sua aplicabilidade numa rede real, em que os caminhos activos e de protecção têm acesso aos mesmos recursos, foi proposto um mecanismo de reserva mínima de LB na protecção local (adaptado de uma formulação para protecção global).

Finalmente, e como contribuição principal foi proposto um novo sistema de recuperação. Este é baseado numa metodologia de recuperação multi-esquema que consiste em aplicar esquemas de recuperação diferentes em função de vários critérios (incluindo características do tráfego a transportar e o estado actual da rede). Os esquemas de recuperação constituintes do novo sistema foram escolhidos de entre os esquemas propostos na literatura ou são suas adaptações. O sistema proposto é modular, pretendendo-se que seja fácil incorporar novos esquemas ou novas regras de selecção dos mesmos. O gestor da rede, ao definir os elementos e a sua articulação no novo sistema de recuperação, controla a forma como será efectuado o tratamento diferenciado dos vários serviços na rede usando para tal engenharia de tráfego tendo em consideração as várias classes de serviço, no âmbito de uma rede de comunicações resiliente. Este sistema foi implementado e sujeito a testes mediante a construção de um simulador (usando o ambiente simulacional OMNeT++). O simulador foi construído de raiz (com cerca de 24000 linhas de código), sendo uma ferramenta que permite diversas configurações e adaptações em termos de esquemas a usar e no métodos de selecção de esquemas por condições (tendo sido criado um mecanismo genérico para permitir a escolha do esquema a usar para o pedido em diversas situações – na geração do pedido, em falha ou no fim da falha). Por último, foram efectuados testes com o simulador, quer como prova de conceito quer como mecanismo de teste de um conjunto de hipóteses subjacentes ao modelo, e analisados os seus resultados.

1.3 Organização do documento

No Capítulo 2 são revistos com algum detalhe aspectos relativos à recuperação de falhas em redes MPLS. Na Secção 2.1 são referidos alguns termos e conceitos usados na recuperação em geral, na Secção 2.2 são apresentados alguns dos aspectos fundamentais do MPLS, na Secção 2.3 são descritos os princípios de recuperação que podem ser usados em redes MPLS e na Secção 2.4 são revistos os esquemas de recuperação de falhas usados em redes MPLS.

No Capítulo 3 são descritas algumas melhorias que podem ser efectuadas a esquemas de partilha de LBP. Assim, na Secção 3.1 é introduzida a nomenclatura utilizada e descrito um método para reserva mínima de LBP na protecção

local. Na Secção 3.2 é revisto um esquema de protecção local e proposta uma nova versão com várias alterações sendo relatados os resultados das modificações introduzidas. Finalmente na Secção 3.3 é descrito o problema de obtenção de um par de caminhos disjuntos de custo mínimo com custos duais nos arcos e da obtenção do conjunto de todos esses pares, sendo apresentados algoritmos para a sua resolução.

No Capítulo 4 é apresentada a proposta de um sistema de recuperação usando diversos esquemas que combina as características do tráfego e a situação da rede com as características dos próprios esquemas. Na Secção 4.1 é apresentado um exemplo de aplicação deste sistema.

No Capítulo 5 é feita a descrição do funcionamento e da implementação do simulador desenvolvido. A Secção 5.1 apresenta uma breve descrição do ambiente de simulação utilizado. Nas Secções 5.2 e 5.3 é feita a descrição do funcionamento do programa de simulação, sendo a sua implementação descrita na Secção 5.4.

No Capítulo 6 são apresentadas experiências que ilustram o funcionamento do esquema de recuperação proposto e validam a sua aplicabilidade. Nas Secções 6.1 e 6.2 são apresentadas as condições gerais das experiências realizadas. Na Secção 6.3 são revistas as medidas recolhidas pelo simulador. Na Secção 6.4 são apresentadas as hipóteses a testar e objectivos das experiências. Por fim na Secção 6.5 é apresentada a descrição das experiências efectuadas e análise dos seus resultados, que são agregados na Secção 6.6.

Finalmente, o Capítulo 7 apresenta uma síntese dos resultados obtidos e propostas de trabalho futuro.

2 Recuperação em MPLS

A recuperação baseada no MPLS refere-se à capacidade de efectuar um restabelecimento rápido e completo do tráfego afectado por uma falha numa rede com MPLS. Nessas redes pretende-se garantir serviços de alta fiabilidade e disponibilidade. Perante falhas, o fornecedor de serviços pretende oferecer, para certos serviços, uma recuperação rápida e que utilize o mínimo de recursos possíveis. Para tal, têm sido vários os esquemas de recuperação propostos. No entanto, deve ser lembrado que os objectivos propostos são, em geral, contraditórios. Em geral a eficiência no uso de recursos impede tempos de recuperação pequenos e vice-versa.

Este capítulo começa por apresentar, uma revisão de alguns conceitos básicos da recuperação, seguindo-se na Secção 2.2 uma descrição sucinta dos princípios de funcionamento do MPLS. Na Secção 2.3 é analisado em detalhe o funcionamento da recuperação em redes MPLS seguindo de perto o *Request For Comments* (RFC) 3469 (Sharma *et al.*, 2003) nas Subsecções 2.3.1, 2.3.2 e 2.3.3; na Subsecção 2.3.5 são abordadas questões sobre QoS e TE em redes MPLS. Na Secção 2.4 é apresentada uma resenha da literatura dedicada ao tema da recuperação em redes MPLS contendo um estudo comparativo dos esquemas em função das suas características mais relevantes.

2.1 Conceitos básicos da recuperação

Com a definição destes conceitos não se pretende estabelecer padrões mas antes arranjar uma terminologia para ser usada neste texto. Os conceitos aqui apresentados são, em parte, derivados do trabalho de Laprie (1992). Muitos destes termos e conceitos iniciais não estão ainda consolidados, existem vários autores que usam o mesmo termo para conceitos distintos e também termos diferentes para o mesmo conceito.

Falha versus avaria Uma avaria (*failure*) é definida como um desvio no serviço especificado, sentido pelo utilizador. E a falha (*fault*) é definida como a causa física ou algorítmica da avaria. As falhas podem ser classificadas

em falhas físicas, aquelas que ocorrem nos componentes, e falhas humanas (falhas de projecto e falhas de interacção). A terminologia, para os conceitos apresentados, foi adoptada de Madeira (1993). As falhas nos componentes físicos de uma rede de computadores não podem ser evitadas, por isso devem ser utilizados mecanismos que ofereçam uma certa confiabilidade para tentar evitar que o sistema apresente avarias. Algumas falhas podem levar horas ou mesmo dias a serem reparadas.

Disponibilidade versus Fiabilidade versus Confiabilidade A disponibilidade (*availability*) é definida como a probabilidade da rede poder ser usada quando necessário com o nível de desempenho combinado (entre o cliente e o fornecedor de serviço), ou por outras palavras como a fracção de tempo que o serviço está disponível. Como *benchmark*, o equipamento das redes de transporte requer disponibilidade da ordem dos cinco noventa e nove (Hussain, 2005), o que significa que o equipamento está disponível para serviço 99,999 por cento do tempo (o equipamento oferece um serviço contínuo excepto aproximadamente durante 5 minutos num ano). Esse valor de disponibilidade deve ser satisfeito pelas redes de dados para voz (Mendiratta e Witschorik, 2003). A fiabilidade (*reliability*) de uma rede é definida como a probabilidade da rede desempenhar as suas funções sem avaria durante um dado período de tempo. Uma medida de fiabilidade usada frequentemente é o tempo médio entre avarias (*Mean Time Between Failures* (MTBF)). Disponibilidade e fiabilidade são atributos de confiabilidade (*Dependability*). Confiabilidade indica a confiança no serviço fornecido por um dado sistema e que será tanto maior quanto maiores forem, simultaneamente, a fiabilidade e disponibilidade. A disponibilidade está relacionada com MTBF através da expressão seguinte:

$$\text{Disponibilidade} = \text{MTBF} / (\text{MTBF} + \text{MTTR})$$

onde *Mean Time to Repair* (MTTR) é a duração média de reparação. Diminuir o tempo de reparação aumenta a disponibilidade. E quanto maior for o tempo entre avarias maior será a fiabilidade. Um sistema pode ser muito fiável e pouco disponível, por outro lado um sistema pode ser altamente disponível e ter vários períodos de inoperacionalidade, desde que estes períodos sejam de pequena duração.

Resiliência versus Capacidade de sobrevivência A capacidade de sobrevivência (*survivability*) é a capacidade de uma rede manter e fornecer um nível de serviço aceitável na presença de falhas (internas ao sistema) e acontecimentos

adversos (externos ao sistema). A capacidade de sobrevivência inclui a resiliência (*resilience*) pois nesta é apenas incluída a capacidade de cumprir a sua missão perante acontecimentos internos. A capacidade de sobrevivência é uma propriedade emergente de um rede de comunicações e não dos seus componentes (Mendiratta e Witschorik, 2003).

Tolerância a falhas A tolerância a falhas (*fault tolerance*) é a capacidade de um sistema ou componente continuar a sua operação normal apesar da presença de falhas de hardware ou software, baseia-se na utilização de redundância. Num sistema com capacidade de sobrevivência a tolerância a falhas é necessária mas não suficiente.

2.2 Princípios do funcionamento do MPLS

Quando um pacote de dados entra na rede MPLS é classificado e de acordo com essa classificação é atribuído a uma *Forwarding Equivalence Class* (FEC). Uma FEC é identificada através de uma etiqueta. O MPLS usa a técnica conhecida por comutação de etiquetas para encaminhar os dados através da rede. Pacotes com a mesma etiqueta são encaminhados do mesmo modo.

A etiqueta é inserida no cabeçalho do pacote e viaja com ele. O formato da etiqueta e como é acrescentada ao pacote depende da tecnologia usada na rede MPLS na camada 2. A etiqueta pode fazer parte de um cabeçalho introduzido especificamente para esse fim ou pode ser introduzida no cabeçalho da camada de ligação de dados (camada 2) se nesse cabeçalho existir um campo para esse fim.

Um dos componentes importantes de uma rede MPLS são os *Label Switching Router* (LSR). Os LSR são “routers” que suportam MPLS e que são capazes de encaminhar pacotes nativos da camada 3 (portanto pacotes não etiquetados).

O caminho seguido por pacotes de uma FEC é designado por *Label Switched Path* (LSP). O LSP é definido por um conjunto de etiquetas associadas a uma FEC em cada *hop*. As decisões de encaminhamento tomadas em cada LSR são baseadas simplesmente na etiqueta com que o pacote chega, a qual é usada como um índice numa tabela que especifica o próximo *hop* e a etiqueta com que o pacote deve ser encaminhado, que é inserida no lugar da etiqueta que o pacote trazia.

De uma forma simplificada, a sequência de *routers* de um LSP, começa com o LSR de entrada no caminho que é o responsável por colocar a etiqueta no pacote e encaminhar o pacote pela interface adequada, seguem-se um ou vários LSR

intermédios que encaminham o pacote com base no valor da etiqueta com que chegam e que colocam o pacote numa interface de saída com uma etiqueta com um novo valor, e por último o LSR de saída do caminho que retira a etiqueta do pacote e o encaminha com base no cabeçalho da camada de rede.

Na Figura 2.1 aparecem representados vários pacotes (pertencentes a duas FEC) com as respectivas etiquetas. Na figura são também ilustrados os dois LSP correspondentes às duas FEC.

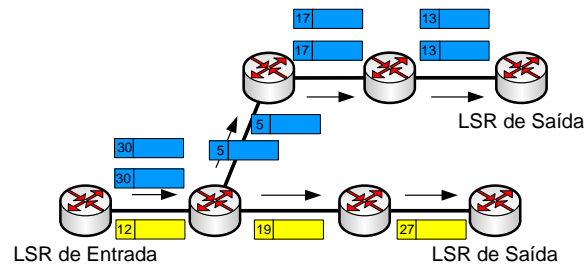


Figura 2.1: Etiquetas e LSP.

A arquitectura do MPLS suporta duas opções para a selecção do LSP para uma FEC particular, o encaminhamento *hop-by-hop* e o encaminhamento explícito. No encaminhamento *hop-by-hop* cada nó escolhe independentemente o próximo *hop* para cada FEC, podendo essa escolha ser baseada nas tabelas de encaminhamento construídas por protocolos de encaminhamento tradicionais. Num LSP de encaminhamento explícito, é um único LSR, geralmente o LSR de entrada ou de saída, que especifica todos (ou vários) os LSR no LSP. A capacidade de estabelecer LSP com encaminhamento explícito é uma das características mais úteis do MPLS. Existem várias situações em que o encaminhamento explícito é útil, por exemplo na implementação de engenharia de tráfego (referida posteriormente).

A arquitectura do MPLS permite que sejam usados diferentes protocolos de distribuição de etiquetas (conjunto de procedimentos através dos quais um LSR informa outro da associação - etiqueta/FEC - que fez) em diferentes cenários. Alguns protocolos têm sido explicitamente desenvolvidos com esse fim, são exemplos o *Label Distribution Protocol* (LDP) em Andersson *et al.* (2001) e o *Constraint-based Routing – Label Distribution Protocol* (CR-LDP) em Ash *et al.* (2002); Jamoussi *et al.* (2002), enquanto que outros protocolos, já existentes, têm sido alargados para permitir a distribuição de etiquetas, é exemplo disso o *Resource Reservation Protocol* (RSVP), com extensão em Awduche *et al.* (2001).

Um pacote etiquetado pode transportar mais do que uma etiqueta, organizadas como uma pilha LIFO (última a entrar, primeira a sair). O processamento de um pacote etiquetado é independente do nível hierárquico da etiqueta, e é

sempre baseado na etiqueta de topo e apenas nela. Quando pacotes etiquetados recebem uma etiqueta adicional diz-se que estão a entrar num túnel MPLS. Quando os pacotes saem do túnel MPLS a etiqueta adicional é retirada e o pacote é encaminhado com base na etiqueta que ficou no pacote, depois dessa operação. O processo de colocação de múltiplas etiquetas num pacote é conhecido como empilhamento de etiquetas. O mecanismo da pilha de etiquetas permite aninhar túneis LSP até qualquer profundidade. O critério usado para atribuir um pacote particular a um túnel LSP é uma questão local ao extremo transmissor do túnel.

2.3 Recuperação em redes MPLS

Têm surgido na Internet muitas aplicações críticas, para as quais as interrupções de serviço são intoleráveis, exemplos dessas aplicações são: *eBusiness*, *eLogistics* e *eHealth*. No entanto, as falhas dos recursos da rede são reais, e por isso os fornecedores de serviços devem fazer com que estas falhas sejam imperceptíveis para os clientes.

As linhas orientadoras relativas à recuperação em redes MPLS foram já objecto de normalização, processo do qual resultou o RFC 3469 (Sharma *et al.*, 2003). A descrição que apresentamos nesta secção segue em geral, esse RFC.

Todas as falhas que podem ocorrer numa rede MPLS são falhas em ramos (o corte de uma fibra ou a falha de um interface de um LSR) ou falhas em nós (a falha no *software*, a falha da alimentação ou uma falha planeada do LSR). Como consequência de uma destas falhas pode resultar a falha de um ou vários LSP.

Sempre que ocorrer uma falha num LSP o esquema de recuperação deve fazer com que o tráfego seja redireccionado para um caminho que contorne a avaria, de forma que a avaria seja o menos perceptível possível.

O LSP que o tráfego segue antes de ocorrer a falha de um nó ou de um ramo (ou mesmo falhas concorrentes) é designado por **caminho de trabalho**, sendo também frequentes as designações **caminho activo**, **caminho primário** ou mesmo **caminho protegido**. O caminho que o tráfego passa a seguir, após a ocorrência da falha pode ser designado por **caminho de recuperação**, **caminho de backup**, **caminho alternativo** ou ainda por **caminho de protecção**.

O caminho de recuperação pode ser um **caminho de recuperação equivalente** ou um **caminho de recuperação limitado** conforme os recursos reservados garantam ou não a mesma qualidade de serviço, respectivamente.

De acordo com o RFC 3469 (Sharma *et al.*, 2003) os dois modelos de recuperação básicos usados para redireccionar o tráfego, após a falha, são **recuperação por**

reencaminhamento¹ (*rerouting*) e **protecção por comutação** (*protection switching*).

Estes dois modelos diferem basicamente na altura em que é estabelecido o caminho de recuperação, depois ou antes da falha ser detectada.

Embora os esquemas de recuperação de tráfego sigam, em geral, um dos dois modelos de recuperação, ambos podem ser utilizados simultaneamente.

O mecanismo que faz com que o tráfego seja redireccionado do caminho de trabalho para o caminho de protecção (ou caminhos de recuperação), quando ocorre uma falha, é designado por *switchover*. O LSR em que é feito o redireccionamento é designado por *Path Switch LSR (PSL)*. O PSL pode simplesmente duplicar o tráfego, enviando-o simultaneamente através do caminho de trabalho, e do caminho de recuperação. Por seu lado o LSR que recebe o tráfego do caminho de recuperação e funde esse tráfego de volta para o caminho de trabalho é designado *Path Merge LSR (PML)*. Os LSR que se encontram entre o PSL e o PML, tanto no caminho de trabalho como no caminho de recuperação, são referidos por **LSR intermédios**.

Além do que foi referido anteriormente, o mecanismo de *switchover*, também designa o processo de transferência do tráfego do caminho de recuperação para um ou mais caminhos de trabalho óptimos. Então, de uma forma mais genérica, o mecanismo de *switchover* designa a operação de redireccionar o tráfego de um dado caminho para um ou mais caminhos alternativos. Por outro lado, o processo de repor o tráfego que seguia um ou mais caminhos de recuperação, de volta no caminho de trabalho original é designado por *switchback*.

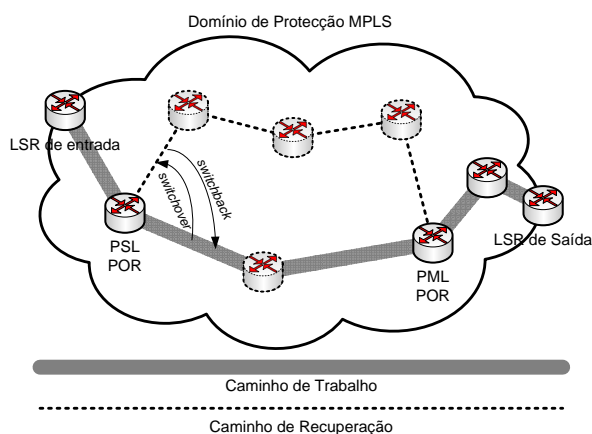


Figura 2.2: Rede que exemplifica alguns dos conceitos básicos.

A Figura 2.2 ilustra alguns dos conceitos definidos anteriormente. Note que, embora na figura o PSL não seja o LSR de entrada e o PML não seja o LSR de

¹Também designada por restauro.

saída, essas situações podem ocorrer. No caso do LSR de saída ser o PML então deve ter como função passar o tráfego aos protocolos das camadas superiores. Na figura aparece ainda a designação *Point of Repair (POR)* que é o LSR que faz a recuperação do LSP. O POR pode ser o PSL ou o PML dependendo do esquema de recuperação utilizado, no entanto, o mais comum é ser o PSL.

Se o LSR que detecta a falha num caminho de trabalho não tiver capacidade de fazer a recuperação deve notificar o POR. Para fazer essa notificação utiliza um sinal designado por *Fault Indication Signal (FIS)*. Um FIS é enviado por cada LSR intermédio para o seu vizinho a montante ou a jusante² até chegar ao POR. Um FIS é transmitido periodicamente pelo LSR/LSRs próximo da falha durante um intervalo de tempo configurável, ou até que o LSR que o transmite receba uma confirmação do seu vizinho.

O LSR que detectou a falha num caminho de trabalho também detectará que a falha foi reparada e quando isso acontecer deve enviar uma notificação para o POR com essa informação. Um sinal que indica que uma falha no caminho de trabalho foi reparada designa-se por *Fault Recovery Signal (FRS)*. Tal como acontece com o FIS também o FRS é enviado por cada LSR intermédio para o seu vizinho a montante ou a jusante² até chegar ao POR. Um FRS também é transmitido periodicamente pelo LSR/LSRs próximo da falha durante um intervalo de tempo configurável, ou até que o LSR que o transmite receba uma confirmação do seu vizinho.

Geralmente a velocidade de *switchover* está directamente relacionada com a distância entre o ponto em que a falha é detectada e o ponto em que é feita a reparação.

Dependendo da localização do POR, no LSP que necessita ser reparado, assim o âmbito da recuperação pode ser **global** ou **local**. Se a recuperação for global então o caminho de recuperação é estabelecido geralmente sempre pelo mesmo LSR do caminho de trabalho enquanto que se a recuperação for local é o LSR do caminho de trabalho que é mais próximo da falha que faz a recuperação.

2.3.1 Princípios de recuperação no MPLS

Nesta secção são apresentados alguns dos princípios de recuperação que podem ser combinados para especificar esquemas de recuperação.

²Se um nó adjacente e a jusante de uma falha envia um FIS/FRS para jusante, tal não implica que o POR (do LSP afectado) esteja necessariamente a jusante da falha.

Instante em que se estabelece o caminho de recuperação

O caminho de recuperação pode ser estabelecido ao mesmo tempo que o caminho de trabalho ou apenas na altura em que a falha ocorre. Para estabelecer o caminho de recuperação, assim como o caminho de trabalho, deve ser utilizado um protocolo de sinalização que pode ser o protocolo *Resource Reservation Protocol with Traffic Engineering* (RSVP-TE), RFC 3209 de Awduche *et al.* (2001), o protocolo CR-LDP (Jamoussi *et al.*, 2002) ou outro protocolo adequado.

Na protecção por comutação o caminho de recuperação é pré-estabelecido antes da falha ser detectada no caminho de trabalho.

Na recuperação por reencaminhamento o caminho de recuperação é estabelecido, através de sinalização, depois de ter sido detectada uma falha no caminho de trabalho. Aqui o caminho de recuperação pode ser **pré-calculado** ou apenas **calculado quando ocorrer a falha**. Por um lado, os caminhos pré-calculados podem utilizar informação de toda a rede, mas poderão não resolver o problema de falhas múltiplas. Por outro lado, calcular o caminho de recuperação apenas quando a falha ocorrer pode conseguir resolver situações de falhas múltiplas mas à custa do aumento do tempo de recuperação.

Existe ainda uma terceira possibilidade: um LSP é estabelecido e o respectivo caminho de recuperação será seleccionado oportunamente entre LSP que não foram inicialmente sinalizados com essa finalidade. Nesta situação intermédia, em que o caminho que vai ser utilizado como caminho de recuperação foi previamente estabelecido, mas que não foi estabelecido explicitamente para proteger o caminho de trabalho, diz-se que o caminho de recuperação é **pré-qualificado**.

Instante em que é feita a reserva de recursos

Os recursos que podem ser reservados são por exemplo a largura de banda, os *buffers* e a capacidade de processamento.

Pré-reservados Este tipo de reserva só é utilizado quando o caminho de recuperação é pré-estabelecido. Neste tipo de reserva os recursos necessários, ao longo de todo o caminho de recuperação, são reservados na altura em que o caminho é estabelecido.

Dependendo da forma como os recursos reservados são usados antes de serem necessários, podem ser considerados os três subtipos seguintes:

- Recursos dedicados - Nesta situação os recursos do caminho de recuperação só podem ser utilizados para transportar o tráfego do caminho de trabalho. A protecção por comutação 1 + 1 ("um mais um") é um

exemplo deste subtipo. Neste tipo de protecção os recursos do caminho de recuperação ficam completamente reservados, e transportam uma cópia do tráfego transportado no caminho de trabalho. Neste subtipo a inteligência está no PML e não no PSL.

- É permitido tráfego extra - É o que acontece quando o caminho de recuperação só transporta o tráfego do caminho de trabalho quando este falha. Nos outros instantes, os recursos reservados podem ser utilizados por tráfego extra de baixa prioridade. Na protecção por comutação designada por 1 : 1 ("um por um") mesmo se os recursos do caminho de recuperação forem pré-reservados, estes estão completamente disponíveis para serem utilizados temporariamente por tráfego de baixa prioridade. Só não podem ser utilizados por esse tráfego quando o caminho de recuperação precisa de ser utilizado pelo tráfego do caminho de trabalho devido à falha deste. Nesta situação o tráfego extra tem que ser deslocado para outros caminhos. Desta forma os recursos do caminho de recuperação são utilizados de uma forma mais eficiente. Note no entanto que na protecção por comutação 1 : 1 não é obrigatória a existência de reserva.
- Recursos partilhados - São recursos de recuperação que são dedicados para serem utilizados por múltiplos recursos primários assumindo que estes não falham simultaneamente. Um exemplo deste subtipo de reserva é o caso em que há reserva de recursos na protecção por comutação do tipo 1 : n ou m : n (" m por n "). Note que também na protecção por comutação do tipo 1 : n ou m : n (" m por n ") não é obrigatória a existência de reserva.

Reservados a pedido Aqui os recursos necessários ao estabelecimento do caminho de recuperação só são reservados depois da falha no caminho de trabalho ser detectada e notificada ao POR, mas antes do *switchover*.

Âmbito da recuperação

Topológico

1. Reparação Global

Na reparação global, o caminho usado na recuperação de qualquer falha (nó ou ramo) do caminho de trabalho (ou segmento do caminho de trabalho) é sinalizado pelo LSR de ingresso do LSP em causa. Por esta razão, é

geralmente considerável a distância entre o ponto em que a falha ocorre e o POR.

Segundo Sharma *et al.* (2003), neste tipo de recuperação, a notificação da falha precisa ser feita através de um FIS. Se o método utilizado se baseasse na falha do teste de continuidade do caminho (ver definição na Subsecção 2.3.1) haveria grandes custos em termos de tempos de recuperação, como Marzo *et al.* (2003) realçaram.

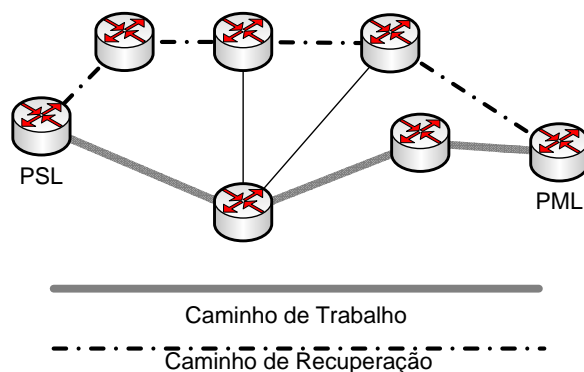


Figura 2.3: Exemplo do modelo de reparação global.

A Figura 2.3 apresenta uma rede simples formada por 10 LSR. É exemplificado nessa rede um caminho de trabalho e o caminho de recuperação respectivo. Em condições normais, o tráfego do LSR de entrada para o LSR de saída segue o caminho de trabalho. Quando ocorre uma falha o tráfego é redirecionado para o caminho de recuperação.

Nos esquemas que calculam o caminho de recuperação antes da falha ocorrer, se esse caminho não for completamente disjunto do caminho de trabalho pode acontecer que a avaria ocorra simultaneamente nos dois caminhos (mesmo tratando-se de uma falha isolada) ficando desse modo a recuperação comprometida. Note que o caminho de recuperação nunca consegue proteger contra falhas no LSR de entrada ou saída.

2. Reparação Local

Na reparação (recuperação) local é o LSR que é imediatamente **a montante** ou **a jusante** à falha, que inicia a recuperação. O caminho de recuperação, estabelecido por esse LSR, é usado na recuperação da falha do ramo adjacente e/ou do nó vizinho do caminho de trabalho.

Deste modo, neste método de recuperação, a falha é reparada pelo LSR mais próximo, conseguindo assim que o tempo de recuperação seja pequeno.

Este tipo de recuperação possui geralmente a vantagem de ter tempos de recuperação menores que o método de reparação global.

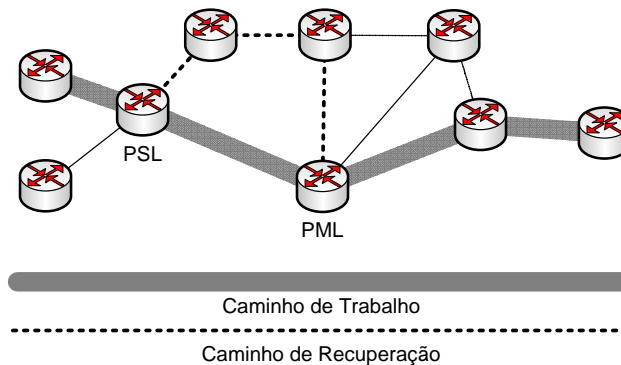


Figura 2.4: Exemplo do modelo de reparação local.

A Figura 2.4 mostra uma rede que também ilustra um caminho de trabalho e um caminho de recuperação, mas agora para o caso da recuperação local. Com o caminho de recuperação representado apenas se consegue recuperar falhas que ocorram no ramo entre o PSL e o PML.

Neste tipo de recuperação, o caminho de recuperação apenas precisa ser disjunto do caminho de trabalho num ramo particular, no caso de recuperação de ramos, ou num LSR particular, no caso de recuperação de nós.

Se apenas algumas partes de um LSP devem ser protegidas (por serem menos fiáveis que as restantes), a utilização de protecção local conduz a um menor consumo de recursos na rede do que a protecção global. Por outro lado se for utilizado este tipo de recuperação para proteger todo o LSP, ele possui a desvantagem, se for utilizado o modelo de protecção por comutação, da necessidade de configuração de vários caminhos de recuperação, além da provável ineficiência em termos de recursos. Este tipo de recuperação é geralmente ineficiente em termos de utilização de recursos pois o percurso que os pacotes seguem após uma falha é frequentemente mais comprido do que seria necessário (Menth *et al.*, 2004; Vasseur *et al.*, 2004).

É frequente encontrar esquemas de reparação local com o modelo de recuperação por reencaminhamento e de reparação global com protecção por comutação, como por exemplo nos esquemas comparados por Ahn *et al.* (2002), embora qualquer outra combinação seja possível como pode ser constatado pelo esquemas descritos na Secção 2.4.

3. Reparação Multi-Camada

Neste tipo de reparação permite-se que várias camadas sejam envolvidas na recuperação.

4. Outros âmbitos de recuperação ainda em termos topológicos

Embora menos referidos na literatura, métodos que consideram a concatenação de vários domínios de protecção e a reparação com LSR de saída alternativo, têm actualmente grande importância, pois hoje em dia é comum um serviço atravessar várias redes que possivelmente utilizam esquemas de protecção diferentes.

Com a concatenação de vários domínios de protecção pretende-se oferecer uma recuperação dos serviços extremo a extremo, embora os esquemas de recuperação dos vários domínios funcionem autonomamente. Devem ser utilizados vários pontos de ligação entre domínios para garantir que se o ponto de ligação falhar existem alternativas.

Na reparação com LSR de saída alternativo o LSR de saída do caminho de recuperação deve ser um *router* que seja aceitável para encaminhar a FEC transportada pelo caminho de trabalho.

Granularidade da recuperação Outro aspecto da recuperação tem a ver com a quantidade de tráfego que precisa ser protegido: parte do tráfego de um caminho, todo o tráfego de um caminho ou o tráfego de um grupo de caminhos.

- Recuperação selectiva do tráfego – Este item diz respeito à possibilidade de proteger apenas uma fracção do tráfego de um caminho individual. Este aspecto é importante pois um mesmo caminho pode transportar classes de tráfego diferentes, que podem ter diferentes necessidades de protecção.
- Agrupamento – Sendo a designação **Grupo de caminhos protegidos** (*Protected Path Group* (PPG)) utilizada para referir um agrupamento lógico de vários caminhos de trabalho que necessitam de protecção, e que são encaminhado da mesma forma entre o PSL e o PML. O princípio de agrupamento é a técnica que corresponde à criação de um PPG, de forma a recuperar simultaneamente os seus múltiplos caminhos.

Se ocorrer uma falha no PPG, este pode ser protegido como um todo redireccionando o tráfego para um *bypass tunnel*. Um *bypass tunnel* é um túnel, utilizado como caminho de recuperação para um PPG, que usa a aproximação do empilhamento de etiquetas (RFC 3031 de Rosen *et al.*, 2001). Quando

os pacotes entram num *bypass tunnel* e-lhes acrescentada uma etiqueta adicional, com base na qual são encaminhados. As etiquetas originais são mantidas inalteradas durante todo o encaminhamento através do *bypass tunnel* para os pacotes poderem ser distinguidos quando saírem dele. A Figura 2.5 apresenta um exemplo de um PPG constituído por três caminhos de trabalho e um *bypass tunnel* para recuperação desse PPG. Os três caminhos de trabalho estão estabelecidos respectivamente entre o LSR A e o LSR X, entre o LSR B e o LSR Y e entre o LSR C e o LSR Z, seguindo cada um os ramos assinalados.

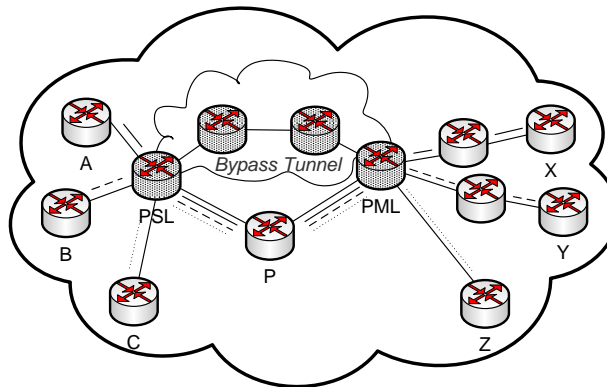


Figura 2.5: Exemplo de um PPG e de um *bypass tunnel*.

Vamos supor que ocorre a falha do LSR P (Figura 2.5). Nesse caso logo que o PSL detecte a falha, cada um dos caminhos protegidos do PPG será reencaminhado para o *bypass tunnel*. As operações de reencaminhamento efectuadas, em cada pacote, no PSL, consistem em trocar a etiqueta do pacote que chega por uma etiqueta apropriada, depois disso, em empilhar no pacote uma etiqueta adicional correspondente ao *bypass tunnel* e por último em redireccionar o tráfego para a interface de saída correspondente ao *bypass tunnel*. A etiqueta apropriada para utilizar na operação de troca é a etiqueta que é esperada pelo PML para o caminho de trabalho ao qual o pacote pertence. Então o PSL tem que conhecer as três etiquetas que o PML espera para os três caminhos de trabalho do PPG³.

Tal como os caminhos de recuperação, os *bypass tunnel* podem ou não ter reserva de recursos suficiente para oferecer a recuperação sem degradação dos serviços.

³Assumindo que o espaço de etiquetas é global.

Detecção da falha

Existe um conjunto de mecanismos que podem ser usados pelo MPLS para detectar as falhas. Assim, o MPLS detecta uma falha porque recebeu uma notificação de uma camada inferior ou da camada IP ou ainda devido à própria operação dos mecanismos baseados no MPLS.

Os dois mecanismos utilizados na detecção de falhas, baseados no MPLS são:

Liveness Message é uma mensagem trocada periodicamente entre dois LSR adjacentes (ver Figura 2.6) que serve como um mecanismo de teste ao ramo. Esta mensagem oferece um teste da integridade das duas direcções do ramo entre os dois LSR bem como um teste à “vida” de LSR vizinhos.

Teste de continuidade do caminho é um teste que verifica a integridade e continuidade de um caminho ou de um segmento de caminho.

Os dois mecanismos anteriores podem ser utilizados para detectar falhas num caminho, ou seja detectar situações em que o caminho deixou de ter conectividade. Embora um caminho não tenha perdido a conectividade a sua qualidade pode tornar-se inaceitável, nesta situação diz-se que o caminho sofreu uma degradação. A degradação de caminhos pode também ser detectada através de mecanismos baseados no MPLS tais como mecanismos de monitorização do desempenho do caminho ou de determinação de taxas de erro no caminho ou segmentos do caminho.

Por outro lado, as falhas nos ramos são normalmente detectadas por uma camada inferior e notificadas por ela ao MPLS. Embora a detecção de falhas de ramos pelas camadas inferiores dependa da tecnologia utilizada é em geral mais rápida que a detecção oferecida por mecanismos de detecção baseados em *hellos* oferecidos pelos protocolos de encaminhamento (por exemplo pelo *Open Shortest Path First* (OSPF)) e por mecanismos de detecção de falhas baseados apenas no MPLS. Apesar disto, como podem existir falhas que são consideradas falhas em ramos mas que não são falhas detectadas pela camada de ligação⁴, os mecanismos de detecção da camada 2 devem ser complementados com mecanismos baseados em *hellos*.

O protocolo RSVP-TE define um protocolo baseado em *hellos*, com funcionamento análogo a qualquer outro mecanismo *hello* oferecido pelos protocolos de encaminhamento, no entanto mais rápido na detecção de falhas.

Qualquer mecanismo baseado em mensagens *hello* possui o problema da escalabilidade. Por um lado, como o processamento das mensagens *hello* não é

⁴Por exemplo, falhas nos processadores das interfaces dos routers.

desprezável, não devem ser utilizadas frequências altas. Por outro lado, o número de vizinhos pode ter que ser limitado. Podem, por vezes, surgir situações em que um *router* vizinho não consiga responder às mensagens *hello* porque está demasiado ocupado e isso pode ser interpretado como uma falha, falha que na realidade não ocorreu.

Por vezes os LSR não conseguem distinguir entre a falha de um ramo e a falha de um LSR vizinho (em ambos os casos a falha detectada é a falha de um ramo). Em Vasseur *et al.* (2004) são indicadas algumas situações em que é importante distinguir entre a falha de um ramo e a falha de um nó, e é proposta uma solução para conseguir distinguir esses dois tipos de falha.

Notificação da falha

Quando um LSR detecta uma falha (falha de um ramo ou nó) que implica a necessidade de recuperação do caminho, e não possui a capacidade de recuperação deve enviar um sinal, designado por FIS, que faça a notificação da falha ao POR. Um FIS é uma mensagem de controlo que deve ser transmitida com alta prioridade. Um FIS, de acordo com Huang *et al.* (2002), pode ser enviado como um pacote da camada 2 ou da camada 3, dependendo da forma como a notificação foi configurada no LSR. Os LSR que recebem um FIS devem determinar para que ramos o propagar. Um exemplo de um FIS é a mensagem “RSVP *Path Error*” do protocolo RSVP (RFC 2205 de Braden *et al.*, 1997).

Um FIS é enviado por cada LSR intermédio, começando no LSR ou LSRs próximos ao ponto em que ocorreu a falha, para o seu LSR a montante ou a jusante até atingir o POR. Se o FIS tiver que ser transmitido no sentido a montante, é óbvio que isso obriga a que tenha que existir um caminho inverso (não necessariamente ao nível do plano do dados) ao caminho usado na transmissão dos dados. Para estas situações Huang *et al.* (2002) definem uma estrutura especial em árvore, designada por árvore de notificação inversa - *Reverse Notification Tree* (RNT), para distribuir eficientemente as mensagens de notificação e da reparação da falha desde o ponto em que a falha ocorreu até ao PSL. Pormenores sobre essa estrutura serão apresentados na Secção 2.4.1.

A Figura 2.6 apresenta um cenário no qual um dos ramos do caminho de trabalho falhou. A figura mostra o envio da notificação pelo LSR que detectou a falha em direcção ao PSL.

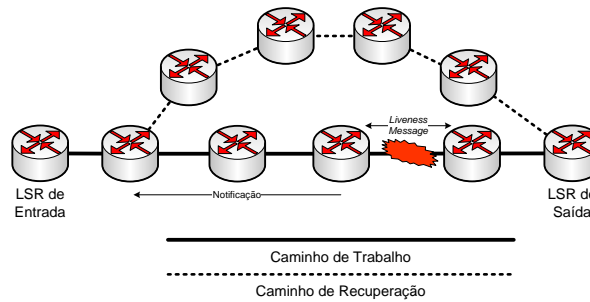


Figura 2.6: Exemplo de detecção e notificação de falhas.

Operação de *switchover*

É necessário que exista um mecanismo despoletador que redirecione o tráfego quando há a detecção de uma falha ou a recepção de uma notificação de falha. O redirecionamento pode ser automático ou ser a resposta a comandos externos. O redirecionamento é automático quando acontece como o resultado de uma falha detectada no PSL ou da recepção de uma notificação de falha no PSL. O redirecionamento é uma resposta a comandos externos quando é o resultado de comandos externos executados por um operador no POR.

Operações após a recuperação

Quando o tráfego é transportado no caminho de recuperação, deve ser feita uma escolha de entre as seguintes opções:

- manter o tráfego no caminho de recuperação mas passar a designá-lo por caminho de trabalho;
- efectuar uma operação de *switchback* para o caminho de trabalho (como é obvio esta operação só deve ser feita quando a avaria for reparada);
- efectuar uma operação de *switchover* para um caminho de trabalho novo.

Os caminhos de trabalho e de recuperação podem manter-se fixos ou podem ser alterados dinamicamente. Se o caminho de trabalho e de recuperação forem fixos podem surgir os problemas seguintes:

- O tráfego não está protegido entre o instante em que ocorre uma falha no caminho de trabalho (e o tráfego é *switched over* para o caminho de recuperação) e o instante em que ocorre a reparação da falha (e o tráfego é *switched back*).

- Os recursos associados com o caminho de trabalho, mesmo quando não estão a ser utilizados por ele, não são libertados para a rede, isto é mantêm-se reservados.

Segundo o RFC 3469 (Sharma *et al.*, 2003) as operações que podem ser configuradas para serem efectuadas após o *switchover* do caminho de trabalho para o caminho de recuperação são o modo reversivo, o modo não reversivo e a alteração dinâmica dos caminhos.

Modo reversivo Quando a avaria no caminho de trabalho for resolvida o tráfego é automaticamente *switched back* para ele. Assume-se que os recursos do caminho de trabalho não são libertados quando ocorre a falha.

Alteração dinâmica dos caminhos Quando o tráfego foi *switched over* do caminho de trabalho para o caminho de recuperação a associação entre eles pode deixar de existir, uma vez que após a falha o caminho de trabalho pode já não existir de existir. Quando a rede encontrar um estado estável, depois da convergência do encaminhamento, o tráfego pode ser *switched over* para um novo caminho preferido.

Modo não reversivo Neste modo a acção realizada depende do que for mais útil em cada caso. Assim sendo depois do *switchover* do caminho de trabalho para o caminho de recuperação pode acontecer uma das três situações seguintes:

- ser feito o *switchback* para o caminho de trabalho original (após reparação deste);
- continuar a utilizar o caminho de recuperação;
- calcular um novo caminho de trabalho óptimo e redireccionar o tráfego para esse caminho.

Restabelecimento

O restabelecimento MPLS significa o retorno do tráfego ao caminho de trabalho original ou o seu redireccionamento para um novo caminho de trabalho. O restabelecimento é feito pelo PSL após receber uma notificação, através de um FRS, indicando que o caminho já foi reparado. O restabelecimento também pode ser feito pelo PSL após receber uma notificação de que um novo caminho de trabalho está estabelecido.

No modo reversivo o LSR que detecta a falha no caminho de trabalho também detecta o seu restabelecimento. Quando detecta o restabelecimento notifica os LSR a montante, através de um FRS. Quando o PSL recebe o FRS, repõe o tráfego no caminho de trabalho original.

No modo dinâmico o mecanismo que despoleta o estabelecimento de um novo caminho de trabalho pode notificar o PSL para este efectuar o *switchover*.

Quando o caminho de trabalho é recuperado este pode trocar de funções com o caminho de recuperação. Isto é, o caminho de trabalho passa a ser o de recuperação e vice-versa. Esta solução requer que o caminho de recuperação seja equivalente ao caminho de trabalho e que seja possível transmitir a informação de uma possível falha ao longo do caminho de recuperação em direcção ao PSL. Aqui o restabelecimento não significa precisamente retorno, como definido inicialmente, mas sim troca de funções.

O retorno do tráfego ao caminho original ou o seu redireccionamento para um novo caminho pode ser efectuado usando a técnica *make-before-break*. Esta técnica consiste em sinalizar um novo caminho, capaz de partilhar recursos com o caminho que se pretende reencominhar. Quando o LSR de ingresso recebe a confirmação do estabelecimento, com sucesso, do novo caminho, o tráfego do caminho original deve ser *switched over* para esse caminho, podendo finalmente ser desligado o caminho original. Desta forma permite-se que o novo caminho possa ter arcos em comum com o caminho original, sem obrigar a dupla captura de largura de banda nesses arcos, o que poderia inviabilizar o redireccionamento.

Resumo de alguns dos princípios de recuperação

A Tabela 2.1 apresenta um resumo de alguns dos princípios de recuperação, nomeadamente o instante em que é estabelecido e calculado o caminho de recuperação e o instante em que é feita a reserva de recursos. Relativamente aos princípios de detecção e notificação da falha, operação de *switchover* e operações após a recuperação, na Secção 2.3.2 faz-se o seu enquadramento.

2.3.2 Ciclos do processo de recuperação do MPLS

Sempre que ocorre um determinado acontecimento (por exemplo uma falha) um esquema de recuperação executa uma sequência de fases designada por ciclo. No processo de recuperação do MPLS são definidos os três ciclos seguintes: o ciclo de recuperação do MPLS, o ciclo de reversão do MPLS e o ciclo de reencominhamento dinâmico. O **ciclo de recuperação** começa quando ocorre uma

| | Reencaminhamento | Pré-qualificado | Protecção por comutação |
|--|---|---------------------------------|-------------------------|
| <i>Estabelecimento do caminho</i> | quando a falha é detectada | — | pré-estabelecido |
| <i>Cálculo do caminho</i> | pré-calculado OU quando a falha é detectada | escolhido entre LSP admissíveis | em paralelo ao primário |
| <i>Reserva de recursos (se houver)</i> | quando a falha é detectada | — | pré-reservados |

Tabela 2.1: Resumo de alguns princípios de recuperação.

avaria e termina quando o tráfego é completamente restabelecido no caminho de recuperação. Depois deste ciclo a rede está de novo operacional.

O **ciclo de reversão** aplica-se ao tráfego encaminhado explicitamente que não depende de quaisquer outros protocolos de encaminhamento dinâmico para convergir, é neste ciclo que ocorre a operação de *switchback*, o que acontece quando a falha está completamente reparada. Aparentemente⁵ o RFC 3469 não considera a hipótese de utilizar como caminho de *switchover*, um LSP explícito (possivelmente diferente do caminho de trabalho original), que tenha sido re-calculado com base no estado da rede.

Por outro lado o **ciclo de reencaminhamento dinâmico**⁶ aplica-se para o tráfego que é encaminhado baseado no encaminhamento *hop-by-hop*, aqui o encaminhamento dinâmico deve determinar um (possivelmente) novo caminho de trabalho re-otimizado depois da ocorrência de uma falha. Se depois dos protocolos de encaminhamento convergirem for determinado que existe um caminho melhor para caminho de trabalho então desencadeia-se este ciclo.

Depois do ciclo de recuperação os caminhos utilizados (caminhos de recuperação) podem não ser tão bons como os caminhos de trabalho. A utilização dos recursos da rede pode ser otimizada se este ciclo for seguido pelo ciclo de reencaminhamento dinâmico. Outra possibilidade é esperar que ocorra a reparação da falha e redireccionar o tráfego do caminho de recuperação de volta para o caminho de trabalho, ou seja, o ciclo de recuperação ser seguido pelo ciclo de reversão. No entanto, existem situações em que o ciclo de recuperação pode não ser seguido por nenhum dos outros dois, é o caso, já referido na Secção 2.3.1, em que se decide manter o tráfego no caminho de recuperação que passará a ser

⁵Em Sharma *et al.* (2003), pág. 12 e Secção 2.2.3, é afirmado que o ciclo de reencaminhamento dinâmico poderá "ser sobreposto" ao ciclo de recuperação e/ou ao ciclo de reversão, mas esta possibilidade não parece contemplar a possibilidade do novo caminho de trabalho (LSP explícito), ser diferente do caminho de trabalho (original) recuperado.

⁶O conceito de encaminhamento dinâmico em redes MPLS não se limita ao encaminhamento *hop-by-hop*, contrariamente ao que aqui é sugerido, ao procurar apresentar a definição de *Dynamic Rerouting Cycle Model* tal como ela surge no RFC 3469.

designado por caminho de trabalho.

Cada um dos ciclos é constituído por vários intervalos de tempo, cada um dos quais será descrito a seguir no ciclo respectivo.

Modelo do ciclo de recuperação do MPLS

Após a ocorrência de uma falha a sucessão de fases seguida pelos diversos esquemas de recuperação é muito semelhante e é designada, como já referido, por ciclo de recuperação. A Figura 2.7 ilustra o modelo do ciclo de recuperação do MPLS.

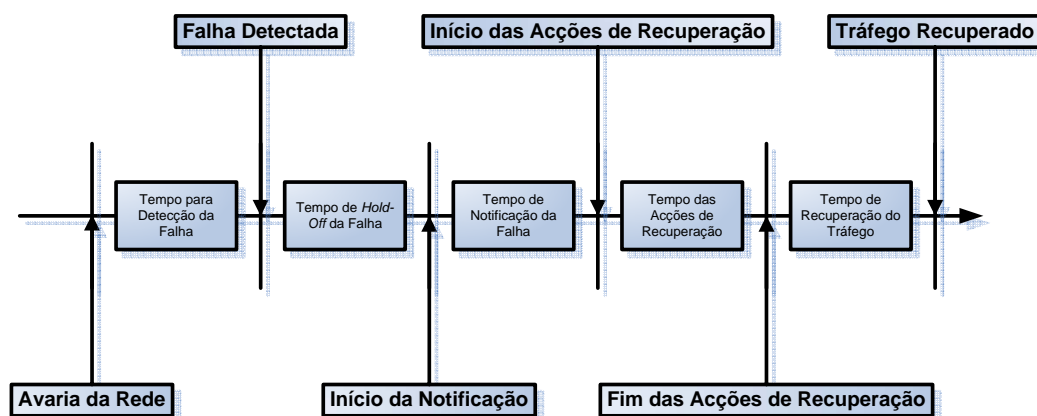


Figura 2.7: Modelo do ciclo de recuperação do MPLS.

As várias fases do modelo do ciclo de recuperação são definidas a seguir.

Tempo para Detecção da Falha Instante entre a ocorrência de uma avaria na rede e o momento em que a falha é detectada pelos mecanismos de recuperação do MPLS. O valor deste tempo é altamente dependente do mecanismo de detecção usado. Por este motivo a duração deste intervalo pode ser fortemente dependente dos protocolos das camadas inferiores, ou da frequência dos sinais enviados entre os LSR (*hellos*), etc.

Tempo de *Hold-Off* da Falha Tempo de espera entre a detecção de uma falha e o início da notificação. Este tempo terá grande importância se a estratégia de recuperação envolver várias camadas. Pode ser por exemplo o tempo necessário para permitir aos esquemas das camadas inferiores repararem a falha. Este tempo é configurável administrativamente e pode ser nulo.

Tempo de Notificação da Falha Se a falha ainda existir depois do Tempo de *Hold-Off* da Falha, são enviadas mensagens de notificação (FIS) para os LSR que

serão envolvidos nas acções de recuperação. O Tempo de Notificação da Falha é o tempo entre o instante em que o sinal que indica a falha (FIS) começa a ser emitido pelo LSR que a detectou e o instante em que o POR começa as acções de recuperação. Este tempo depende do esquema de recuperação usado, pode mesmo ser zero (pois a recuperação do tráfego pode ser feita pelo LSR imediatamente a montante à falha).

Tempo das Acções de Recuperação Tempo entre o instante em que o POR inicia as acções de recuperação e o instante em que termina essas acções.

Tempo de Recuperação do Tráfego Tempo entre o a última acção de recuperação e instante em que o tráfego (se existir) é completamente recuperado.

O tempo total da recuperação é obtido somando os valores de todos os tempos anteriores.

Modelo do ciclo de reversão do MPLS

Nos esquemas de recuperação revertivos a operação de *switchback* também segue uma sucessão de fases gerais a qual é designada por ciclo de reversão. A Figura 2.8 ilustra o modelo do ciclo de reversão do MPLS.

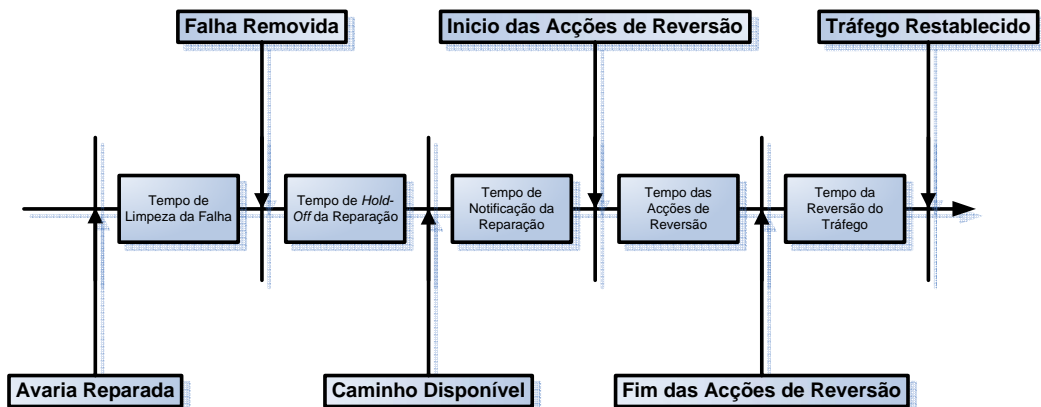


Figura 2.8: Modelo do ciclo de reversão do MPLS.

Os intervalos de tempo do modelo do ciclo de reversão são definidos a seguir.

Tempo de Limpeza da Falha Instante entre a altura em que a avaria foi reparada e o momento em que os mecanismos baseados no MPLS detectaram que a falha já não existe.

Tempo de *Hold-Off* da Reparação Tempo de espera, que é configurado, entre o instante em que os mecanismos baseados no MPLS detectaram que a falha já não existe e o instante em que estes mecanismos efectivamente iniciam a notificação da reparação da falha. Este tempo pode ser necessário para assegurar que o caminho é estável (que não se trata de uma falha intermitente).

Tempo de Notificação da Reparação Tempo entre o instante em que o FRS começa a ser emitido pelo LSR que inicia a notificação da reparação da falha e o instante em que o LSR responsável pelas acções de reversão as inicia. Este tempo pode ser zero.

Tempo das Acções de Reversão Tempo entre o instante em que o POR inicia as acções de reversão e o instante em que termina essas acções.

Tempo da Reversão do Tráfego Tempo entre o a última acção de reversão e instante em que o tráfego (se existir) é completamente restabelecido no caminho de trabalho.

Modelo do ciclo de reencaminhamento dinâmico

A Figura 2.9 ilustra o modelo do ciclo de reencaminhamento dinâmico.

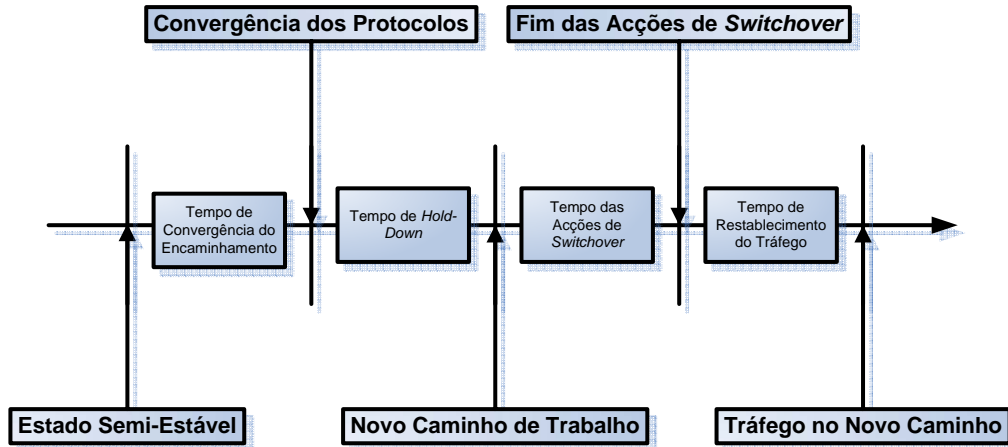


Figura 2.9: Modelo do ciclo de reencaminhamento dinâmico.

Os intervalos de tempo do modelo do ciclo de reencaminhamento dinâmico são definidos a seguir.

Tempo de Convergência do Encaminhamento É o tempo para os protocolos de encaminhamento convergirem e para a rede alcançar um estado estável.

Tempo de *Hold-Down* (opcional) É um período de tempo durante o qual deve ser utilizado o caminho de recuperação.

Tempo da Acções de *Switchover* Tempo entre a primeira e a última acção de *switchover*.

Tempo de Restabelecimento do Tráfego Tempo entre o a última acção de *switchover* e instante em que o tráfego (se existir) é completamente restabelecido no novo caminho de trabalho.

2.3.3 Critérios de comparação de esquemas de recuperação

Nesta secção vão ser enumerados alguns critérios que podem ser utilizados na comparação dos esquemas de recuperação baseados no MPLS ainda segundo Sharma *et al.* (2003).

Tempo da Recuperação É definido como sendo o tempo que demora a activação do *caminho de recuperação* após a falha. Na Figura 2.7 é o tempo total desde o instante em que ocorreu a avaria da rede até ao instante em que tráfego foi completamente recuperado.

Tempo Total da Reposição Tempo necessário até ao restabelecimento permanente do tráfego. Pode ser igual ao anterior se os caminhos trocarem de funções, ou pode ser diferente, se existir necessidade de reversão ou encaminhamento dinâmico.

Vulnerabilidades na inicialização Outro critério que pode ser utilizado na comparação dos esquemas de recuperação é o tempo que o caminho de trabalho ou conjunto de caminhos de trabalho funcionam sem protecção. Exemplos de situações em que esta condição ocorre são o tempo durante o cálculo do caminho de recuperação e o tempo necessário ao estabelecimento do caminho de recuperação.

Capacidade de *Backup* A quantidade da Capacidade de *Backup* necessária quando ocorre uma falha pode diferir de esquema de recuperação para esquema de recuperação.

Latência Cumulativa Os esquemas de recuperação podem aumentar a latência ao tráfego. Por exemplo, um caminho de recuperação pode ter muitos mais nós que o caminho de trabalho.

Qualidade da protecção O grau de garantia de sobrevivência do tráfego é outro factor dependente do esquema de recuperação. A sobrevivência de um pacote pode variar de *sobrevivência relativa* a *sobrevivência absoluta*. Sobrevivência relativa pode (por exemplo) significar que o pacote está num pé de igualdade com outro tráfego. Sobrevivência absoluta pode (por exemplo) significar que o tráfego protegido tem garantias explícitas de sobrevivência.

Reordenação de pacotes Também é dependente do esquema de recuperação a necessidade de reordenação de pacotes que pode ocorrer tanto após a operação de *switchover* como após a operação de *switchback*.

(Overhead) associado ao estado A quantidade de informação de *estado* necessária para registar os caminhos de recuperação aumenta à medida que aumenta o número de caminhos de recuperação. No entanto, a quantidade de informação de *estado* necessária por cada esquema de recuperação pode também depender de muitos outros parâmetros, como por exemplo o número, tipo dos itens protegidos e da definição de *Shared Risk Link Group* (SRLG).

Perda de pacotes Durante o *switchover* um determinado número de pacotes pode ser perdido. Uma medida do número de pacotes perdidos pode ser obtida através da proporção entre o tempo de recuperação e a velocidade do ramo.

Cobertura Diferentes esquemas de recuperação oferecem diferentes tipos de cobertura de falhas. Para avaliar um esquemas de recuperação em termos de cobertura total devem ser tidas em consideração várias métricas. Algumas dessas métricas são:

- **Tipo de falha** As falhas cobertas podem ser apenas falhas nos ramos, ou falhas nos nós e nos ramos ou também ser considerada a degradação de serviço.
- **Número de falhas simultâneas** É dependente do esquema de recuperação a possibilidade de recuperação de várias falhas que possam ocorrer simultaneamente.
- **Número de caminhos de recuperação** Número de caminhos de recuperação que podem existir por cada falha.
- **Percentagem da cobertura** Percentagem das falhas que podem ser cobertas. Esta percentagem pode ser subdividida em percentagens por tipo de falha.

- O número de caminhos protegidos pode afectar a rapidez com que o conjunto total de caminhos afectados por uma falha podem ser recuperados. A **razão de protecção** é n/N , onde n é o número de caminhos protegidos e N é o número total de caminhos.

Destes critérios de comparação os utilizados mais frequentemente são: o tempo da recuperação, a percentagem de pacotes reordenados, a percentagem de pacotes perdidos e a eficiência em termos de utilização da capacidade de *backup*.

2.3.4 Os vários significados de recuperação global e local - clarificando a notação

Na recuperação global o objectivo é oferecer protecção para a falha de qualquer nó ou ramo do caminho de trabalho ou seu segmento, como já foi referido. Neste tipo de recuperação é usual o LSR de entrada e o LSR de saída do caminho de trabalho coincidirem com o PSL e com o PML, respectivamente. Um exemplo de uma situação em que tal não se verifica é ilustrado na Figura 2.10, na qual o PSL (origem do tráfego recuperado) não é a origem do tráfego de trabalho. Isso acontece porque apenas o segmento entre o PSL e o PML necessita protecção (Huang *et al.*, 2002; Sharma *et al.*, 2003). Muitos autores omitem o caso em que apenas um segmento do caminho de trabalho necessita de protecção é o caso, por exemplo de Park *et al.* (2004), Mellah e Mohamed (2003) e Vasseur *et al.* (2004).

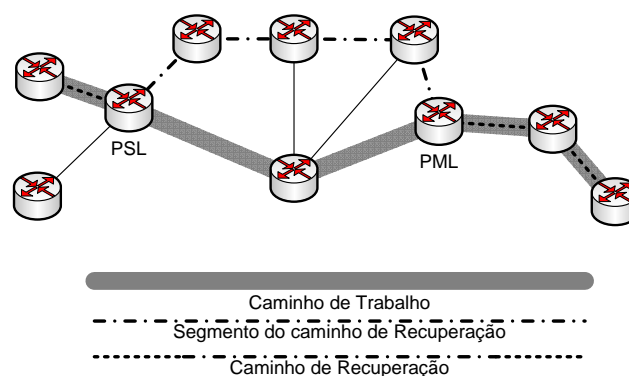


Figura 2.10: Exemplo do modelo de reparação global.

Consideramos, em sentido lato, que um mecanismo de recuperação é global sempre que a acção de recuperação é desempenhada pelo mesmo par de LSR (o PSL e o PML), qualquer que seja a localização da falha nesse caminho. Quando o PSL e o PML coincidem com o LSR de entrada e o LSR de saída, respectivamente, este mecanismo é designado por recuperação extremo a extremo (*end-to-end*).

As definições apresentadas para recuperação global e local (Secção 2.3.1) representam apenas dois extremos do conjunto de possibilidades, pois existem muitas situações intermédias, como realçado por Vasseur *et al.* (2004). Podem apontar-se como exemplos de situações intermédias os esquemas propostos por Hong *et al.* (2004), Yoon *et al.* (2001), Haskin e Krishnan (2001). Para abranger as situações intermédias, que consideramos se enquadram no âmbito da recuperação local, vamos alargar esta definição. Vamos considerar como sendo recuperação local toda a recuperação que é feita por um LSR próximo da falha, não obrigatoriamente um dos LSR adjacentes à falha (porque utilizando um destes LSR pode não existir um caminho que contorne a falha, ou porque estes LSR podem não ter capacidade de efectuar a recuperação, ou por qualquer outra razão que os impeça).

Normalmente a recuperação local será tentada pelo LSR a jusante mais próximo da falha no LSP de trabalho. Caso esse LSR não seja bem sucedido, poderá enviar para o nó seguinte a montante a informação necessária para que este tente por sua vez recuperar o LSP afectado. Consideramos que, quando a acção de recuperação é efectuada com sucesso pelo o LSR de entrada, porque nenhum dos outros LSR a montante da falha o conseguiu (ou seja pelo menos um tentou e falhou), estamos ainda na presença de um mecanismo de recuperação local.

2.3.5 A qualidade de serviço e a engenharia de tráfego no MPLS

As redes de telecomunicações têm actualmente que oferecer uma grande quantidade de serviços com características muito distintas. Na Tabela 2.2 são apresentadas as várias características de alguns serviços. A tabela inclui dados retirados de várias obras, incluindo Vasseur *et al.* (2004) e Tanenbaum (2003).

Na Tabela 2.2, para as aplicações de vídeo a pedido e de notícias a pedido, com grande sensibilidade ao *jitter* (sensíveis à chegada de pacotes com intervalos de tempo irregulares entre eles, considerando que estes foram enviados regularmente), o nível de *jitter* aparece assinalado com símbolo (a), o que significa que nessas aplicações o *jitter* pode ser eliminado fazendo *buffering* no receptor e utilizando os dados a partir do *buffer* em vez de os obter da rede em tempo real. Sempre que na coluna "Sensibilidade ao atraso/*Jitter*" aparece apenas um valor significa que a sensibilidade ao atraso e o *jitter* são semelhantes. As grandes exigências em termos de *jitter* das primeiras cinco aplicações da tabela resultam da necessidade de interacção em tempo real entre as pessoas. A última coluna da tabela dá uma indicação do nível de exigência dos mecanismos de recuperação para as várias aplicações consideradas. É visível que para muitas das aplicações existe

| Aplicações | Taxa de bits | Sensibilidade ao atraso/ Sensibilidade ao Jitter | Necessidade de recuperação |
|--|---------------------|---|----------------------------|
| Serviço de telefone tradicional (POTS) | 32-64 Kbps | Muito grande | Muito grande |
| Voz sobre IP | 8-32 Kbps | Muito grande | Muito grande |
| Vídeo Telefone | 256-1920 Kbps | Muito grande | Muito grande |
| Videoconferência | Pelo menos 256 Kbps | Muito grande | Muito grande |
| Teletrabalho | 64 Kbps até 2 Mbps | Muito grande | Grande |
| TV por Broadcast | 2-8 Mbps | /Grande | Grande |
| Ensino à distância | 64 Kbps até 2 Mbps | Muito grande | Muito grande |
| Vídeo a pedido | 750 Kbps até 4 Mbps | Reduzida /Grande (a) | Média |
| Notícias a pedido | 64 Kbps | Reduzida /Grande (a) | Reduzida |
| E-mail | 64 Kbps até 1 Mbps | Reduzida | Reduzida |
| Transferência de arquivos | 64 Kbps até 2 Mbps | Reduzida | Reduzida |
| Acesso à web | 64 Kbps até 1 Mbps | Média /Reduzida | Reduzida |
| Login remoto | 64 Kbps até 1 Mbps | Média | Reduzida |
| Televidas | 64 Kbps até 2 Mbps | Média | Reduzida |

Tabela 2.2: Necessidades em termos de QoS.

uma grande exigência em termos de recuperação. É também visível na tabela uma forte relação entre as exigências em termos dos mecanismos de recuperação e a sensibilidade ao atraso/*jitter* para cada uma das aplicações. As necessidades em termos de tempo de recuperação podem também ser estimadas da tabela, pois quanto mais sensíveis forem as aplicações ao atraso/*jitter* mais precisam de tempos de recuperação pequenos. As aplicações em tempo real, como por exemplo telefone e a videoconferência, são as mais exigentes em termos de tempos de recuperação.

Quando um cliente pretende receber serviços diferenciados do seu *Internet Service Provider* (ISP) eles devem estabelecer um SLA. O SLA especifica basicamente as classes de serviço suportadas e a quantidade de tráfego permitido em cada classe. A QoS de cada uma das classes de serviço é expressa em termos de latência, *jitter*, garantias de LB, resiliência face a falhas e o tempo em que a rede está em baixo (*downtime*).

Os ISP começaram por oferecer QoS usando apenas DiffServ, ou seja, dividiam o tráfego num pequeno número de classes e reservavam recursos para cada uma dessas classe. A indicação da classe a que o pacote pertence é feita diretamente no pacote, no campo *DiffServ Code Point* (DSCP) do cabeçalho IP. O campo DSCP determina o tratamento a ser dado a um pacote num nó particular da rede, este tratamento é designado por *Per-Hop-Behavior* (PHB). O PHB que um pacote recebe é expresso em termos de escalonamento e preferência de eliminação. Do ponto de vista da implementação o PHB traduz-se na fila usada para

o encaminhamento dos pacotes, na probabilidade de eliminação no caso da fila exceder um certo limite, nos recursos reservados (*buffers* e LB) para cada fila, e na frequência com que a fila é servida. Os PHB standards definidos pelo *Internet Engineering Task Force* (IETF) são:

Best Effort (BE) - tráfego que não recebe qualquer tratamento especial;

Assured Forwarding (AF) - tráfego com poucas perdas de pacotes mas com garantias mínimas em termos de latência;

Expedited Forwarding (EF) - tráfego com atraso, variação do atraso e perdas mínimas (tráfego com fila dedicada na qual a taxa de chegada dos pacotes é menor que a taxa de serviço).

Foram definidas apenas uma classe de encaminhamento expresso (classe EF), uma classe BE e 12 classes de encaminhamento garantido (classes AF), todas no formato AF_xy, no qual x indica o número da fila (um valor entre 1 e 4) e y indica o nível de preferência de eliminação (um valor entre 1 e 3).

Numa rede MPLS os LSR tomam as decisões de encaminhamento baseadas no cabeçalho de adaptação MPLS. Então para implementar DiffServ no MPLS pode-se usar o campo *Experimental* (EXP) desse cabeçalho para indicar o PHB. No entanto este campo possui apenas 3 bits o que pode ser considerado insuficiente em redes com mais de oito PHB. Faucheur *et al.* definiram em 2002 o modo do MPLS suportar DiffServ. A solução assenta no uso de dois tipos de LSP:

1. *EXP-Inferred-PHB Scheduling Class LSP* (E-LSP) – são LSP que podem transportar simultaneamente vários PHB, o campo EXP do cabeçalho de adaptação do MPLS indica ao LSR qual o PHB a ser aplicado ao pacote (indica, simultaneamente, o tratamento de escalonamento e a preferência de eliminação a ser aplicada ao pacote). Aqui um único LSP pode transportar pacotes com até oito PHB distintos.
2. *Label-inferred-PHB Scheduling Class LSP* (L-LSP) – são LSP que transportam tráfego de um único PHB ou de vários PHB que partilham o mesmo tratamento de escalonamento. O tratamento de escalonamento é inferido, pelo LSR, exclusivamente a partir do valor da etiqueta do pacote e a informação da preferência de eliminação é fornecida no campo EXP do cabeçalho de adaptação do MPLS.

O conjunto de nós com capacidades DiffServ nos quais foi definido um conjunto comum de PHB, com o mesmo mapeamento DSCP/PHB definem um domínio DiffServ.

DiffServ não pode garantir QoS quando o caminho que o tráfego segue não oferecer recursos adequados para atingir as necessidades de QoS.

Com a rede sobre-dimensionada é possível oferecer boa qualidade de serviço sem *Diff-Serv-aware Traffic Engineering* (DS-TE) (Osborne e Simha, 2003), a menos que a rede fique congestionada devido a falhas na rede (Minei, 2004). No entanto, esta não é a solução em geral pois isso seria dispendioso em termos dos custos da LB. Então para oferecer QoS a um serviço, numa rede não sobre-dimensionada, são necessárias duas coisas: encontrar um caminho na rede que possa oferecer o serviço e usar mecanismos de DiffServ (policiamento, marcação, filas de escalonamento e eliminação) para implementar o serviço. Ou por outras palavras, para oferecer os diferentes serviços os LSR na rede possuem filas de espera, escalonamento e comportamentos de eliminação diferentes baseados no tipo de aplicação e além disso devem existir garantias de LB por aplicação.

A TE é o processo que faz com que os fluxos de tráfego sejam distribuídos por caminhos particulares de forma a evitar a congestão por utilização inadequada da rede (partes da rede congestionadas enquanto outras possuem muita capacidade livre). A utilização inadequada da rede pode ser causada pela utilização dos protocolos de encaminhamento tais como o *Routing Information Protocol* (RIP), o OSPF ou o *Intermediate System – to – Intermediate System* (IS-IS). O encaminhamento baseado em restrições (*Constraint-based routing* (CBR)) torna o processo de TE automático. O CBR desenvolveu-se a partir do encaminhamento com QoS. Em Pinto (2005) é apresentado o estudo de vários algoritmos de encaminhamento com QoS. O CBR além do objectivo já referido de seleccionar caminhos que aumentam a utilização da rede possui também o objectivo de seleccionar caminhos que obedeçam a certas necessidades de QoS, o que não substitui a necessidade de DiffServ. Para atingir os objectivos o CBR entra em consideração com as necessidades dos fluxos, com os recursos disponíveis nos ramos, e possivelmente com outras métricas (tais como a fiabilidade, o atraso, o *jitter* e o número de *hops*). Devido à complexidade do problema a resolver uma possibilidade é escolher apenas a LB e uma das outras métricas.

Quando os pacotes são enviados nos LSP TE, é da responsabilidade de cada LSR ao longo do caminho fornecer um serviço apropriado ao pacote (através de DiffServ). São várias as razões que fazem com que seja importante utilizar engenharia de tráfego tendo em consideração os vários níveis de classes - DS-TE.

Para evitar que as aplicações que são sensíveis ao atraso/*jitter* sejam perturbadas é importante não transportar mais do que uma certa percentagem (proporção) de tráfego dessas aplicações em qualquer ramo (Faucheur e Lai, 2003). Esta é uma das razões para a necessidade de DS-TE, pois só assim é possível aplicar

diferentes restrições de LB a “classes de serviço”⁷ diferentes.

É necessário configurar, em cada ramo, o tamanho das filas de espera e as suas políticas de escalonamento para oferecer um tratamento adequado a cada “classe de serviço”. Torna-se então necessário garantir uma restrição de LB diferente para diferentes classes de serviço, e dessa forma fazer corresponder o tráfego de cada “classe de serviço” encaminhado num ramo à configuração de escalonamento desse ramo. Garantir uma restrição de LB diferentes para diferentes “classes de serviço” é conseguido através da utilização de DS-TE, sendo esta outra razão para a necessidade de DS-TE.

Outra razão ainda para a necessidade de DS-TE é pretender fornecer o nível de serviço adequado ao tráfego com garantias de QoS e ao mesmo tempo também usar TE no restante tráfego, o que requer que o CBR seja capaz de garantir diferentes restrições de LB para este tráfego.

No MPLS com TE um caminho (LSP TE) é determinado de forma a obedecer a um conjunto de restrições, através de CBR. Para tornar possível o CBR é necessário que os LSR possuam informação actualizada do estado dos ramos. Os protocolos de *Link-State Advertisement* (LSA) tais como o OSPF e o IS-IS já possuem extensões, definidas em Katz *et al.* (2003); Smit e Li (2004), para distribuição de informação útil na TE, e extensões adicionais para o suporte de DS-TE definidas em Faucheur (2005b). Para que a informação distribuída pelas extensões dos *Interior Gateway Protocol* (IGP) (OSPF ou IS-IS) traduza a verdadeira ocupação dos ramos, os vários *Class Type* (CT) devem reservar a LB que precisam (além da necessidade adicional, já referida, de mecanismos de policiamento na entrada da rede). O protocolo utilizado para reserva de LB utilizado pelo MPLS é o RSVP (também utilizado no estabelecimento dos LSP, como já referido previamente). Em Faucheur *et al.* (2002) são apresentadas as extensões ao RSVP para o suporte de DiffServ, e em Faucheur (2005b) as extensões necessária para o suporte de DS-TE no MPLS (ambas extensões adicionais às extensões ao RSVP para TE em Awduche *et al.* (2001)).

MPLS DS-TE combina as vantagens de TE com as vantagens de DiffServ, resultando na capacidade de oferecer garantias exigentes de QoS ao mesmo tempo que optimiza a utilização dos recursos.

O exemplo na Tabela 2.3 apresenta uma solução DS-TE para um caso particular. Juntamente com a apresentação do exemplo irão ser introduzidas algumas definições, úteis no contexto da engenharia de tráfego no MPLS tendo em consideração as diferentes classes de serviço.

⁷“classes de serviço” usado para referir uma {*Traffic Aggregate (TA)*}PHB *Scheduling Class* (PSC)

| DiffServ PHB | Classe de Serviço | Traffic Trunk | Tipo de Classe | Classe TE |
|--------------------|-------------------|---------------|----------------|--------------------------|
| <i>EF</i> | {TA}PSC1 | TT4 | CT3 | TE4(CT3,5) |
| | | TT3 | CT2 | TE3(CT2,4) TE2(CT2,2) |
| <i>AF1x</i> | {TA}PSC2 | TT2 | CT1 | TE1(CT1,3) |
| <i>AF2x</i> | {TA}PSC3 | TT1 | | |
| <i>Best Effort</i> | {TA}PSC4 | TT0 | CT0 | TE0(CT0,0) |

Tabela 2.3: Exemplo com quatro comportamentos DiffServ distintos.

Neste exemplo foi considerado que na rede existem quatro comportamentos DiffServ distintos no domínio, o PHB EF, o PSC AF1x, o PSC AF2x e o PHB BE. Se não for feita nenhuma divisão dos pacotes das quatro FEC correspondentes às 4 “classes de serviço” ({TA}PSC1 até {TA}PSC4) resultarão quatro *Traffic Trunks*. No entanto vamos considerar que a “classe de serviço” {TA}PSC1 é dividida em dois *Traffic Trunks* para fins de distribuição da carga (Faucheur e Lai, 2003). Considerando que neste domínio os objectivos de QoS de AF1x e de AF2x são semelhantes então as mesmas restrições de LB podem ser aplicadas a TT1 e TT2, definindo então apenas um tipo de classe para esses dois *traffic trunks*. Como consideramos que a “classe de serviço” {TA}PSC1 é dividida em dois conjuntos de *traffic trunks* então cada um desses conjuntos é sujeito a diferentes restrições de LB. Por último, considerou-se uma prioridade de preempção associada a CT0, uma associada a CT1 e uma associada a CT3. A CT2 foram associadas duas prioridades de preempção. Note que, embora no exemplo não sejam associados valores de prioridade iguais a tipos de classes diferentes tal é possível.

Com DS-TE é possível garantir restrições de LB – *Bandwidth Constraint* (BC) – diferentes para os diferentes conjuntos de *Traffic Trunk* (TT), através da utilização de um modelo de restrição de largura de banda – *Bandwidth Constraints Model* (BCM). Um BCM é definido por um número de restrições de LB, a que CT é que cada restrição de LB se aplica, e como esta se aplica. Existem vários desses modelos. Eles fornecem as “regras” que suportam a reserva de LB para os tipos de classes (CT) individuais e que podem ser usadas para suportar o controlo de admissão ao nível dos LSP e para o encaminhamento (cálculo dos LSP) baseado em restrições. Por simplicidade é aceitável escolher o mesmo modelo em todos os ramos da rede (Faucheur e Lai, 2003). Nos exemplos apresentados nas Figuras 2.11, 2.12 e 2.13, relativos a cada um dos BCM, é ilustrada uma situação em que são considerados três CT. O RFC 4125 (Faucheur, 2005a) e o RFC 4127 (Faucheur, 2005c) complementam as especificações funcionais para os mo-

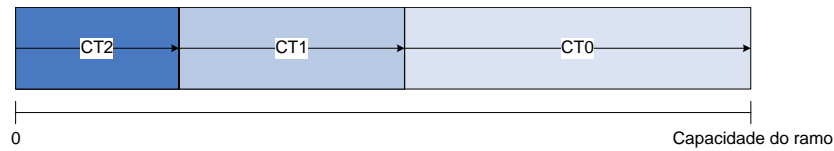


Figura 2.11: Diagrama do modelo MAM.

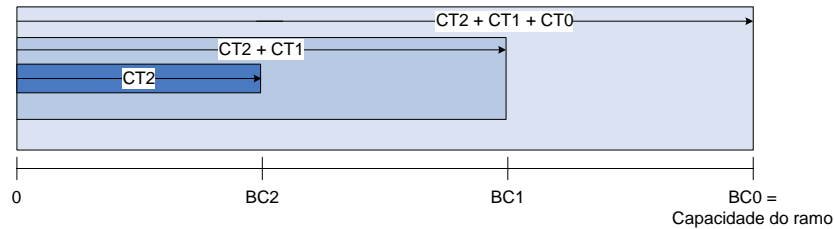


Figura 2.12: Diagrama do modelo RDM.

delos *Maximum Allocation Bandwidth Constraints Model* (MAM) e para o modelo *Russian Dolls Bandwidth Constraints Model* (RDM), respectivamente, previamente especificadas no RFC 3564 (Faucheur e Lai, 2003).

O modelo MAM é o modelo mais simples. Neste modelo a LB de um ramo pode ser dividida pelas diferentes CT conforme indicado na Figura 2.11. O valor da LB que cada CT pode reservar é quando muito o valor da parcela de LB que lhe foi atribuída, designada por restrição de LB, BC, para essa classe. Para permitir partilha de LB entre as várias CT, é necessário permitir que a soma das BC exceda a capacidade do ramo.

No modelo RDM o valor da LB que cada CT pode reservar é no máximo a soma da LB atribuída às CT mais exigentes que esta adicionado de uma constante, como visível na Figura 2.12. Porém esta LB pode na realidade ser partilhada por todas as CT menos exigentes. De igual forma esta CT pode usar LB não reservada pelas CT mais exigentes até ao limite indicado.

Uma característica geral destes métodos é o compromisso no uso da partilha da LB, entre as várias CT, para aumentar a eficiência (na utilização da LB) em condições normais de tráfego e a necessidade de obter protecção ou isolamento de CT de forma robusta perante sobrecarga (Lai, 2005).

O RFC 4126 (Ash, 2005) apresenta as especificações funcionais para outro modelo, o modelo *Max Allocation with Reservation Bandwidth Constraints Model* (MAR). Neste modelo cada CT possui um limite de LB que pode porém ultrapassar deste que não exista sobrecarga (não existe sobrecarga sempre que a LB total disponível no ramo ultrapassar o limiar de reserva de LB), conforme ilustrado na Figura 2.13.

Nestes modelos a LB reservada para um dado CT é a quantidade total de LB reservada por todos os LSP estabelecidos pertencentes a esse tipo de classe .

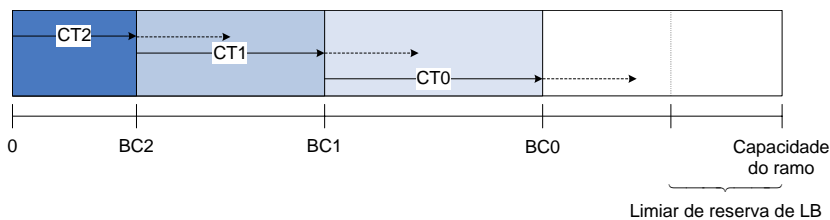


Figura 2.13: Diagrama do modelo MAR.

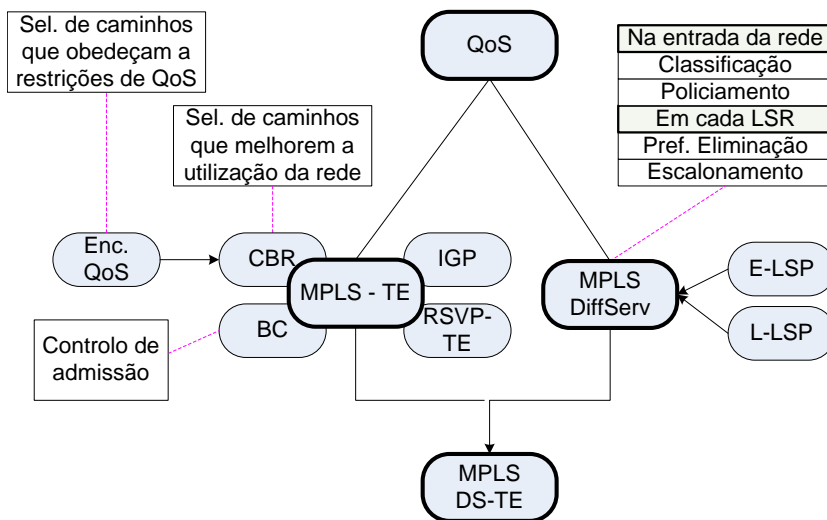


Figura 2.14: Diagrama MPLS DS-TE.

Para ter em simultâneo DS-TE e LB partilhada por vários LSP de protecção é necessário que os caminhos de protecção partilhados e os caminhos protegidos por estes pertençam todos ao mesmo tipo de classe DS-TE segundo Minei e Lucek (2005). Faucheur (2005a), Ash (2005) e Faucheur (2005c) referem esta necessidade no âmbito da conjugação de DS-TE com modelos de recuperação que partilhem LB entre caminhos de protecção, nomeadamente no *Shared mesh restoration* (Lang *et al.*, 2002).

O diagrama 2.14 resume os aspectos, descritos anteriormente, envolvidos quando se pretende fornecer QoS com o MPLS.

2.4 Esquemas de recuperação no MPLS

Têm sido vários os esquemas de recuperação propostos na literatura. Um fornecedor de serviços de rede deve poder aplicar esquemas de recuperação diferentes de acordo com as características de Qualidade de Serviço dos fluxos de tráfego (classes de serviço) que transporta.

Nas descrições seguintes relativas aos esquemas de recuperação apenas são detalhados, para cada um, o aspecto essencial à sua operação. Por exemplo para uns será o algoritmo de encaminhamento (para o caminho de trabalho e de recuperação), para outros será a forma de notificação das falhas, para outros ainda será a distribuição de carga por vários caminhos, etc.

O critério principal utilizado para organizar os esquemas de recuperação foi o modelo de recuperação (protecção por comutação ou reencaminhamento).

2.4.1 Protecção por comutação

Caminho de protecção inverso ao de trabalho

No documento Haskin e Krishnan (2001) é definido um esquema que estabelece caminhos de recuperação que oferecem protecção contra uma falha isolada (ramo ou nó). A característica essencial do esquema proposto é a forma como é construído o caminho de recuperação. O caminho de recuperação é formado por dois segmentos: o primeiro segmento começa no LSR, do caminho de trabalho, adjacente ao LSR de saída e vai através de todos os LSR protegidos, na direcção inversa ao caminho de trabalho, até ao LSR de entrada. Alternativamente este segmento pode começar no LSR de saída, sendo em tudo o resto igual ao anterior; o segundo segmento é estabelecido entre o LSR de entrada e o LSR de saída sem partilhar nenhum LSR com o caminho de trabalho, ou seja, é um caminho paralelo ao caminho protegido.

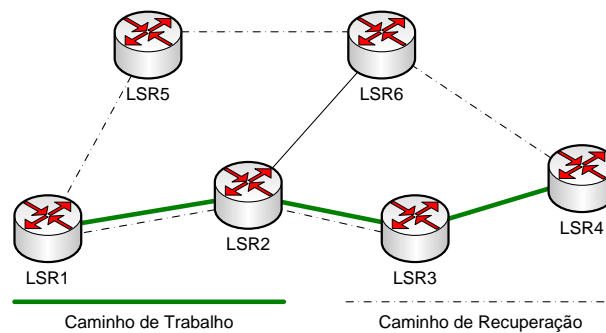


Figura 2.15: Rede com exemplo de caminhos, de acordo com o esquema de Haskin e Krishnan (2001).

Na Figura 2.15 apresenta-se uma rede simples constituída por 6 LSR. A figura mostra um caminho primário estabelecido entre o LSR 1 e o LSR 4 (1-2-3-4) e também o correspondente caminho de recuperação completo estabelecido entre o LSR 3 e o LSR 4 (3-2-1-5-6-4).

Quando uma falha é detectada por um LSR o tráfego é redireccionado para

o caminho de recuperação já pré-estabelecido. É o LSR que detecta a falha que inverte o tráfego de volta para o LSR de entrada (através do segmento do caminho de recuperação que tem direcção inversa ao caminho primário). Quando o tráfego chega ao LSR de entrada este faz com que ele siga o segundo segmento do caminho de recuperação (segmento do caminho de recuperação que é paralelo ao caminho protegido). Este esquema pode utilizar a protecção 1 : 1 ou 1 : n .

Recorrendo de novo à Figura 2.15 e supondo que falha o ramo entre o LSR3 e o LSR4, quando o LSR3 detectar essa falha inverte o tráfego para o caminho de recuperação (3-2-1-5-6-4), no entanto, se a falha for por exemplo no LSR2 o caminho de recuperação será apenas (1-5-6-4).

O esquema proposto tem as vantagens de uma recuperação rápida (o caminho de recuperação é pré-estabelecido e o LSR que detecta a falha reencaminha imediatamente os pacotes para o caminho de recuperação) e simultaneamente da minimização da complexidade do cálculo e da sinalização do caminho de recuperação (apenas é calculado um caminho de recuperação para proteger todo o caminho de trabalho). Consequentemente, como a recuperação é rápida é minimizada a perda de pacotes, quando um LSP falha.

Uma das desvantagens do esquema é o aumento da latência no caminho de recuperação face ao caminho de trabalho. A latência pode ser reduzida se o LSR de entrada ao detectar o fluxo de tráfego inverso deixar de enviar o tráfego pelo caminho de trabalho e o passar a enviar logo através do segundo segmento do caminho de recuperação, no entanto, esta opção aumenta possivelmente a necessidade de reordenação de pacotes. Outra desvantagem do esquema é não ser geralmente eficiente em termos de utilização de recursos.

Segundo Haskin e Krishnan (2001) o esquema por eles proposto requereria alguma extensão de sinalização.

Notificação na protecção global do caminho

O esquema proposto por Huang *et al.* (2002) segue o modelo de protecção por comutação. O esquema protege o caminho de trabalho (ou segmento do caminho de trabalho) de qualquer falha num seu nó ou ramo, como um todo. Quando um LSR detecta uma falha tem que enviar uma mensagem de notificação da falha que deve alcançar o PSL. O objectivo principal do esquema é minimizar o atraso na propagação dessas mensagens de notificação.

O mecanismo proposto define uma estrutura de notificação especial em árvore para distribuir eficientemente e rapidamente a informação da falha e/ou da recuperação. A essa estrutura chamaram árvore de notificação inversa – *Reverse*

Notification Tree (RNT). O mecanismo define também um protocolo *Hello* como uma forma de detecção de falhas, mais rápido do que o *Hello* da camada IP, para falhas que não são detectadas pelas camadas inferiores e que devem ser detectadas e corrigidas na camada MPLS. É definido ainda um protocolo de transporte leve e escalável para o transporte das mensagens de notificação.

O mecanismo proposto consiste portanto em três componentes:

RNT Antes de descrever a estrutura RNT é importante lembrar que o MPLS possui um mecanismo designado por fusão de etiquetas, no qual vários caminhos de trabalho podem convergir para formar uma árvore multiponto-ponto, em que os PSL são as folhas. Em Huang *et al.* (2002) chamam a esta fusão de etiquetas, fusão física dos LSP. E definem fusão virtual dos LSP quando vários LSP, que têm origem em diferentes LSR, partilham um segmento comum para além de algum nó e também um identificador, mas não são fundidos fisicamente.

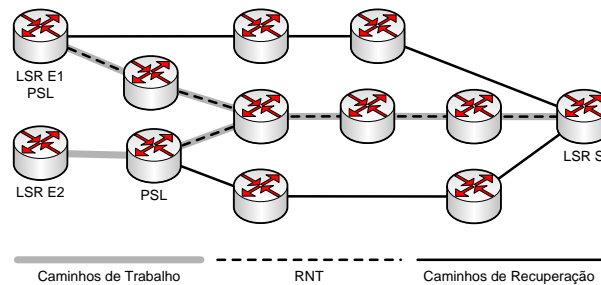


Figura 2.16: Exemplo de uma RNT.

Quando ocorre fusão física ou virtual dos LSP o mecanismo de propagação das mensagens de notificação não é simples. Nestas situações eles propõem uma simplificação na sinalização das notificações, através da criação de uma estrutura de notificação inversa (designada por RNT), que é uma árvore ponto-multiponto. A Figura 2.16 apresenta um exemplo de uma RNT numa rede que tem estabelecidos dois caminhos de trabalho e os dois caminhos de recuperação respectivos. A raiz da estrutura RNT é um LSR escolhido de entre os do segmento comum dos LSP em que há fusão dos caminhos e a estrutura termina nos PSL. A estrutura RNT define o modo como é feita a propagação das mensagens de notificação e da reparação da falha desde o ponto em que a falha ocorre até ao PSL. Esta estrutura pode ser criada como extensão ao processo de estabelecimento do LSP. A simplificação na sinalização das notificações consiste em enviar uma única mensagem de notificação nos segmentos partilhados pelos LSP. Portanto

a estrutura RNT possibilita a redução no *overhead* de sinalização associado com a recuperação, ao contrário do que se passa em esquemas que tratam cada LSP independentemente, que necessitam de sinalização por cada LSP. Esta estrutura pode ser implementada na camada 3, na camada 2 ou mesmo na camada 1 para reduzir mais o atraso.

Protocolo *Hello* para detecção de falhas Cada LSR deve ser capaz de detectar certos tipos de falhas. O protocolo *Hello*, semelhante ao protocolo *Hello* do OSPF, deve despoletar a geração do FIS, quando necessário.

Neste protocolo são trocadas mensagens *liveness* periodicamente entre os LSR vizinhos. Uma mensagem *liveness* transporta o identificador do LSR que a envia e todos os identificadores dos seus vizinhos, descobertos através das mensagens (*liveness*) enviadas pelos vizinhos. Um LSR pode determinar se um ramo bidireccional está a funcionar correctamente se vir a sua própria identificação na mensagem *liveness* enviada pelo LSR no outro extremo do ramo.

Protocolo de transporte leve Os pacotes de notificação são transmitidos periodicamente até que o nó responsável pelo *switchover* seja atingido. Neste protocolo as mensagens não são confirmadas por isso recorre-se a temporizadores. Um temporizador indica o tempo durante o qual são transmitidas as mensagens de notificação. Um outro temporizador serve para determinar o ritmo a que as mensagens de notificação são emitidas. O valor do período deste temporizador não deve ser demasiado grande nem demasiado pequeno de modo a que o atraso na notificação do FIS não aumente muito devido à perda de um FIS e para que não sejam consumidos demasiados recursos, respectivamente. A notificação é efectuada através da propagação do FIS na RNT.

Protecção/recuperação local baseada em túneis

Em Mellah e Mohamed (2003) é proposto um algoritmo de protecção local baseado na utilização de túneis de recuperação, designados por *bypass tunnels* (de acordo com a designação utilizada no RFC 3469). Os túneis são pré-estabelecidos localmente por cada LSR para contornar a avaria de um ramo. De acordo com o algoritmo proposto, para todos os pacotes que chegam a um LSR, este testa o estado do ramo de saída e efectua a operação normal na etiqueta do pacote. Se verificar que existe falha desse ramo então acrescenta uma nova etiqueta à pilha de etiquetas do pacote (a etiqueta do túnel de recuperação), e o pacote segue o

bypass tunnel correspondente. Logicamente, nesta situação o pacote sai do LSR por um *interface* diferente daquele que seria utilizado caso não houvesse a avaria do ramo. Se verificar que não existe falha do ramo de saída, o pacote é encaminhado nesse ramo (e sem empilhamento de etiquetas). A Figura 2.17 apresenta um cenário no qual um dos ramos do caminho de trabalho, que se encontra protegido por um *bypass tunnel*, falhou. A figura mostra o empilhamento da etiqueta do *bypass tunnel* nos pacotes dos dois caminhos de trabalho protegidos por ele.

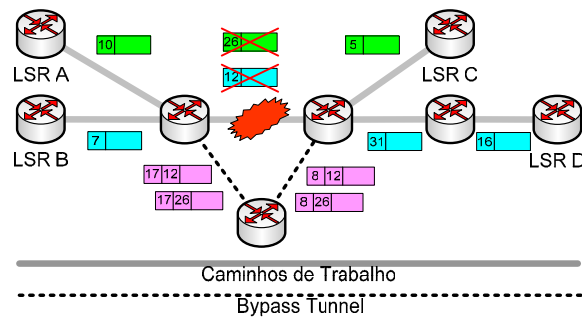


Figura 2.17: *Bypass tunnel* para protecção de um ramo.

A informação relativa ao estado dos ramos é actualizada periodicamente, nas entradas respectivas da tabela de encaminhamento de etiquetas pelos LSR adjacentes a esses ramos, e sempre que é detectada uma falha localmente.

A grande vantagem do esquema proposto são os tempos pequenos de recuperação, pois quando ocorre uma falha não existe a necessidade enviar uma notificação da falha para o LSR de entrada, dado que a recuperação é local, e porque simultaneamente o caminho de protecção é pré-estabelecido. No entanto, o esquema proposto só trata de falhas em ramos, o qual se pode considerar incluído no mecanismo designado por *one-to-one backup* (Pan *et al.*, 2005).

Recuperação utilizando “*p*-cycles”

Muitos esquemas de recuperação utilizam túneis de backup, como é o caso do esquema apresentado anteriormente. O encaminhamento utilizado nos túneis não é indicado em muitos desses esquemas. Em Kang e Reed (2003) é apresentado um trabalho dedicado ao problema do encaminhamento nos *bypass tunnels* com largura de banda garantida (o túnel de recuperação deve oferecer largura de banda garantida para todos os caminhos protegidos por ele, em casos de falha isolada). Nesse artigo os autores investigam a utilização de “*p*-cycles” (*preconfigured-cycles*) no encaminhamento desses túneis.

O conceito “*p*-cycle” foi inicialmente proposto para a recuperação em redes de

transporte como as redes SONET ou as redes WDM em Grover e Stamatelakis (1998). Os mesmos autores também estudaram em Stamatelakis e Grover (2000a) a utilização do conceito “ p -cycle” na recuperação de redes com características do MPLS, mas sem a restrição de largura de banda garantida.

A ideia base do conceito “ p -cycle” é a pré-configuração da capacidade extra da rede em ciclos, que devem ser criados antes de ocorrer qualquer falha. Com este conceito são recuperadas falhas não apenas no ciclo mas também falhas de ramos que ligam dois pontos não adjacentes do “ p -cycle” (ramos *straddling*) e neste caso existem dois caminhos de recuperação para cada falha. A Figura 2.18 apresenta um exemplo de um “ p -cycle” e de ramos *straddling*. A recuperação de ramos *straddling* contribui significativamente para a eficiência dos “ p -cycles”.

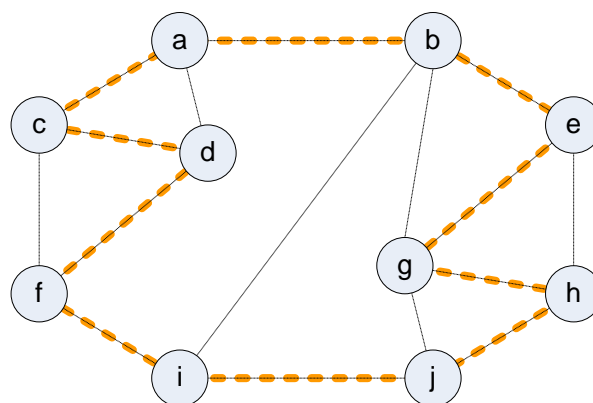


Figura 2.18: Exemplo de um “ p -cycle” e ramos *straddling*.

Em Grover e Stamatelakis (1998) os autores apresentam uma formulação para o projecto óptimo dos “ p -cycle”, para a recuperação em redes de transporte SONET ou WDM. Esta formulação divide-se em duas fases: a primeira fase consiste em gerar o conjunto de todos os ciclos distintos elementares, até um tamanho limite, para a topologia da rede existente; a segunda fase consiste na resolução de um problema de programação linear inteira que gera um plano óptimo dos “ p -cycle” através da escolha do número de cópias de cada ciclo elementar no grafo da rede, para ser configurado como um “ p -cycle”. Posteriormente os mesmos autores, em Stamatelakis e Grover (2000a), fizeram (segundo Kang e Reed (2003)) a adaptação da formulação original dos “ p -cycle” para redes IP/MPLS.

No trabalho apresentado em Kang e Reed (2003) foi adaptada a formulação, para o projecto óptimo dos “ p -cycle”, proposta em Stamatelakis e Grover (2000a) acrescentando-lhe a protecção com largura de banda garantida. Nessa adaptação foi tida em atenção apenas a protecção contra falhas isoladas de ramos. Como o problema de optimização dos “ p -cycle” é *NP-hard* (Stamatelakis e Grover, 2000a),

é proposto em Kang e Reed (2003) um método de relaxação que pré-selecciona ciclos candidatos. Este método foi baseado no resultado teórico apresentado em Stamatelakis e Grover (2000b), que indica que “*p*-cycles” grandes tendem a oferecer maior eficiência em termos de capacidade. Estes resultados foram confirmados, no caso geral, através dos resultados das experiências apresentadas em Kang e Reed (2003). Na primeira experiência apresentada, constataram que para os casos em que tal não se verifica isso é devido a duas razões: 1 – a utilização de ciclos grandes faz com que seja necessário reservar mais capacidade de *backup*, no caso da protecção de ramos sobrecarregados (ramos com muito tráfego de caminhos de trabalho), do que se fossem utilizados ciclos mais pequenos; 2 – a existência de poucos ciclos candidatos grandes, (nomeadamente com uma distribuição não uniforme da carga de trabalho), conduz a uma menor flexibilidade e consequentemente a uma diminuição da eficiência da solução.

Em Kang e Reed (2003) dizem que o problema do projecto dos “*p*-cycles” pode ser simplificado considerando apenas como ciclos candidatos os ciclos grandes, que são óptimos no caso da distribuição uniforme da carga na rede. No entanto como constaram que, no caso em que a distribuição de tráfego é não uniforme, existe uma degradação da eficiência causada pelo número limitado de ciclos candidatos, concluíram que alguns ciclos mais pequenos deveriam ser acrescentados. Assim, a heurística, proposta com o objectivo de reduzir a dimensão do problema, consiste em pré-seleccionar o conjunto dos ciclos maiores e o conjunto dos ciclos mais pequenos. No artigo apresentam resultados de experiências realizadas em várias redes que atestam que a sua heurística é superior a outras consideradas.

Através de várias experiências realizadas, em Kang e Reed (2003) os autores também mostram que uma boa distribuição da carga de trabalho pela rede pode melhorar o desempenho dos “*p*-cycles” em termos de utilização de capacidade.

Algumas considerações finais relativamente à utilização de “*p*-cycles” na recuperação de redes MPLS:

- Com a utilização de túneis de recuperação encaminhados utilizando o conceito “*p*-cycle” os tempos de recuperação são pequenos, pois os túneis são determinados e estabelecidos antes de ocorrer qualquer falha, e não há necessidade de sinalização para notificar a falha pois a recuperação é local. Portanto, o tempo de recuperação é apenas o tempo da detecção da falha e de comutar o tráfego para o caminho de recuperação.
- Uma desvantagem da aplicação do conceito “*p*-cycle” à recuperação em redes MPLS é a quantidade de túneis que é necessário estabelecer. Para

cada “*p-cycle*” é necessário no mínimo estabelecer tantos túneis quantos o número de ramos do “*p-cycle*” mais o dobro do número de ramos *straddling* desse “*p-cycle*”.

- Em redes reais de grandes dimensões não podem ser utilizados os ciclos máximos mas apenas os ciclos maiores permitidos, porque em ciclos grandes o sinal pode sofrer uma degradação inaceitável e porque ciclos grandes têm maior probabilidade de falhas múltiplas. Por outro lado, mesmo após a aplicação da heurística apresentada em Kang e Reed (2003), a dimensão do problema pode ainda ser excessiva.
- Para evitar a dependência de um controlo centralizado para o desenvolvimento e manutenção de um estado óptimo de pré-configuração dos “*p-cycle*”, pensamos que haveria vantagem em desenvolver um protocolo distribuído de auto-planeamento que continuamente aproximasse o estado óptimo, à semelhança do que foi proposto para a recuperação em redes de transporte SONET ou WDM em Grover e Stamatelakis (1998).

Recuperação com dois caminhos de protecção por cada nó protector

No esquema proposto em Bartoš e Raman (2001) os caminhos de protecção de todos os LSR de entrada para cada LSR de saída são calculados simultaneamente, após o que são estabelecidos. O algoritmo de cálculo dos caminhos de protecção (apresentado em Bartoš *et al.* (2001)) é executado concorrentemente em todos os LSR de saída, para oferecer protecção completa para todo o domínio de protecção. Como resultado de uma execução do algoritmo obtém-se, em cada LSR de saída, dois caminhos de protecção disjuntos para cada um dos pares LSR de entrada / LSR de saída, em que o LSR de saída é fixo. Esses caminhos de protecção protegem os caminhos de trabalho contra a falha isolada de um ramo qualquer. A localização dos caminhos de protecção, determinados pelo algoritmo, não é influenciada pela localização dos caminhos de trabalho (os quais podem não ser disjuntos com os caminhos que os protegem) o que oferece uma maior flexibilidade ao esquema de protecção. O algoritmo faz a fusão dos caminhos de protecção sempre que possível, diminuindo assim o consumo de recursos.

O algoritmo garante as duas condições seguintes: todo o LSR protector (LSR que pode efectuar recuperação) possui uma ligação, através de dois caminhos de protecção, ao LSR de saída; esses dois caminhos de protecção estão colocados de forma que nenhuma falha de um ramo causará perda de conectividade entre um LSR protector e o LSR de saída. Isto é, a falha de um ramo não causa a perda de

conectividade simultânea em ambos os caminhos de protecção.

A notificação da falha de um ramo, aos LSR afectados, é feita utilizando um FIS, transmitido pelos LSR adjacentes ao ramo que falhou e que se propaga através da RNT. Quando os LSR recebem um FIS, deixam de transmitir tráfego nos caminhos de trabalho em que ocorreu a avaria de um ramo, e redireccionam o tráfego para os caminhos de protecção globais. No entanto, todo o tráfego de caminhos de trabalho com falha num ramo que chegou aos LSR que são adjacentes ao ramo que falhou, imediatamente depois do ramo falhar e antes da propagação completa do FIS, é redireccionado para os caminhos de protecção que emergem desses LSR protectores. Desta forma, no esquema proposto, a recuperação é tão rápida como a recuperação local mas a quantidade de recursos necessários é normalmente reduzida relativamente a outros esquemas de recuperação local. O esquema proposto oferece protecção contra a falha de um ramo, mas falhas múltiplas em ramos e falhas em nós não foram consideradas pelo autores.

As medidas consideradas pelos autores para determinar a qualidade de um esquema de protecção foram o comprimento dos caminhos de protecção e o número de caminhos de protecção por ramo. Eles referem que o seu esquema é mais eficiente que outros em termos destes critérios mas não explicitam a forma de cálculo utilizada para obter esses parâmetros.

Embora não seja propriamente um esquema muito simples na construção (sinalização) dos caminhos, parece um esquema interessante, e fácil de implementar sem exigir grandes recursos computacionais. Uma vez que os caminhos de protecção ignoram o caminho de trabalho e a capacidade dos ramos no seu cálculo, a sua aplicabilidade em situações de tráfego elevado (*Best Effort*) não parece comprovada.

Em Bartoš e Raman (2001) a contagem do número de caminhos que é preciso estabelecer não é devidamente explicada. Para que essa contagem ficasse clara, seria necessário que tivesse sido explicado o método de sinalização dos LSP de protecção.

Os LSP de protecção, são calculados pelo LSR de saída mas têm de ser sinalizados pelo LSR de entrada de cada um deles. Essa dificuldade poderá ser contornável se houver troca de informação entre os LSR de saída e uma entidade central de gestão da rede que solicite (aos LSR de entrada) o estabelecimento dos LSP de protecção calculados em cada LSR de saída.

Os LSP de protecção poderão ser sinalizados sob a forma de túneis, mas para que as regras de fusão esboçadas em Bartoš e Raman (2001) se apliquem, será necessário que todos esses LSP pertençam à mesma sessão (considerando fixo o LSR de saída). Em Bartoš e Raman (2001) dizem que fazem a fusão dos caminhos

para diminuir o consumo de recursos. no entanto LSP explícitos só podem ser fundidos em situações específicas (ver RFC 4090 Pan *et al.*, 2005).

Determinação dos caminhos através da resolução de problemas de programação linear

Em Yetginer e Karasan (2002), os autores estudam um esquema de recuperação com as seguintes características: o caminho de trabalho e o caminho de recuperação são determinados *off-line*, e a recuperação é global, por protecção por comutação e com reserva de largura de banda. No esquema proposto existe a possibilidade de partilha de largura de banda entre caminhos de protecção que protegem caminhos de trabalho disjuntos nos ramos. Todas as falhas na rede de um único ramo são completamente recuperáveis.

No esquema de recuperação que os autores estudam, o aspecto central explorado é a forma como são determinados os caminhos. Os autores apresentam algoritmos para quatro abordagens ao processo de determinação dos caminhos de trabalho e dos caminhos de recuperação. Todos os algoritmos para a determinação dos caminhos são formulados como problemas de programação linear inteira.

Os algoritmos das quatro abordagens determinam um conjunto inicial de caminhos e seleccionam dele caminhos de trabalho e de recuperação para todos os pedidos. Considera-se que a procura na rede é conhecida, e que cada pedido de ligação entre dois nós requer uma dada largura de banda. A partir destes pedidos, o conjunto inicial de caminhos é obtido determinando para cada pedido, um conjunto de caminhos disjuntos nos ramos, em que o número de caminhos é maximizado e simultaneamente a largura de banda total utilizada por eles é minimizada.

O primeiro algoritmo começa por seleccionar (de uma vez só) os caminhos de trabalho, de entre os elementos do conjunto previamente determinado. Seguidamente selecciona os caminhos de recuperação. Em ambos os casos o objectivo é que a largura de banda total utilizada na rede seja mínima, isto é, que a capacidade residual total na rede seja máxima. A capacidade residual total é considerada como sendo a soma das capacidades residuais em todos os ramos.

No segundo algoritmo o projecto dos caminhos de trabalho e o projecto dos caminhos de recuperação também é feito em separado. Porém, além de pretendem que a capacidade residual total na rede seja maximizada também pretendem que a carga seja distribuída pela rede, de modo a evitar que uns ramos possam estar congestionados enquanto outros estão sub utilizados. Para o conseguir ma-

ximizam o mínimo das capacidades residuais individuais, considerando todos os ramos.

Os terceiro e o quarto algoritmos, ao contrário dos anteriores, integram o cálculo dos caminhos de trabalho e dos caminhos de recuperação. O terceiro algoritmo tem os mesmos objectivos que o segundo algoritmo diferindo deste apenas na forma integrada como são obtidos os caminhos de trabalho e protecção.

O quarto algoritmo (muito semelhante ao anterior) procura manter com elevada capacidade residual os ramos que estima (com base em dados históricos) que serão os mais utilizados. Para tal atribui a cada ramo um peso inversamente proporcional ao nível de utilização esperada.

Os autores comparam as quatro abordagens em função da sua capacidade de suportar variações nos padrões de tráfego. Para isso observam como é distribuída a capacidade residual pela rede e determinam o número de pedidos adicionais que podem ser transportados, mantendo os caminhos de trabalho existentes inalterados, para cada uma das abordagens.

Concluíram que ao distribuir cuidadosamente a carga de tráfego pelos recursos da rede, a abordagem que integra o projecto dos caminhos de trabalho e recuperação e que faz a distribuição da carga considerando os ramos com pesos adequados, tem um desempenho melhor que as outras abordagens no transporte de tráfego adicional resultante de impreviões no tráfego.

O esquema proposto, qualquer que seja a abordagem escolhida para seleccionar os caminhos, parece ser dificilmente implementável em redes de dimensão não trivial porque exige a resolução de problemas de programação linear inteira complexos. Como os cálculos são feitos *off-line* esta restrição é ligeiramente reduzida, mas ainda assim parece inviável para redes de dimensão média ou superior.

Determinação *on-line* dos caminhos para um novo pedido

No esquema de recuperação proposto em Kodialam e Lakshman (2002b), quando surge um pedido de um novo LSP os caminhos (activo e de recuperação) são determinados *on-line*. Se não for possível garantir largura de banda para recuperação (medida de QoS utilizada) o pedido é rejeitado. Caso contrário, o/os caminho/s de recuperação é/são estabelecidos ao mesmo tempo que o caminho activo. Em Kodialam e Lakshman (2002b) os autores descrevem algoritmos de encaminhamento dinâmico para a determinação do caminho activo CA e de recuperação CR. Consideram três situações distintas de informação disponível e designam os três modelos de informação resultantes por: modelo com Nenhuma Informação (*No Information* (NI)), modelo com Informação Completa (*Full Infor-*

mation (FI)) e modelo com Informação Parcial (*Partial Information* (PI)). No primeiro apenas é conhecida a LB reservada em cada ramo⁸ (e a LB residual), no segundo são conhecidos os percursos de todos os CA e CR e no último é conhecida a LB total utilizada pelos CA e a LB total utilizada pelos CR em cada ramo (apenas é conhecida informação agregada⁸ que não depende do número de LSP estabelecidos). O único modelo com interesse prático é o modelo com PI. Os modelos com NI e com FI⁹ podem ser utilizados para obter o limite superior e o limite inferior, respectivamente, de LB necessária ao modelo com PI. Em Kodialam e Lakshman (2002b) os autores apresentam a formulação de algoritmos para a recuperação global (para os três modelos de informação) e em Kodialam e Lakshman (2001) apresentam a formulação de um algoritmo para a recuperação local (para o modelo PI). Os algoritmos apenas consideram falhas isoladas de ramos mas podem facilmente ser estendidos para considerar também falhas isoladas de nós.

Designam por partilha inter pedido (*interdemand*) a partilha de caminhos de recuperação entre pedidos cujos LSP activos não partilham o mesmo ramo (mesmo que os CA não sejam disjuntos pode ainda ser possível existir partilha). Designam por partilha intra pedido (*intrademand*) a partilha entre os vários LSP de recuperação de um mesmo LSP activo (possível apenas na recuperação local). Em cada algoritmo exploram a possibilidade de partilha de largura de banda (LB) dos CR. Essa partilha será tanto maior quanto mais informação, relativa à utilização dos ramos, estiver disponível.

Recuperação global – Algoritmos para selecção do CA e do CR

1. Modelo com Nenhuma Informação

Neste caso não é possível qualquer partilha de LB. O objectivo é seleccionar os caminhos (activo e de recuperação) que minimizem a largura de banda total usada pelos dois. Para tal deve ser escolhido o par de caminhos disjuntos nos ramos com o menor número de ramos. Este problema é formalizado como um problema standard de fluxo de redes (considerando cada ramo com custo e capacidade unitária). Pode ser resolvido por qualquer algoritmo de fluxo de custo mínimo (por exemplo, pelo algoritmo de Suurballe (Suurballe e Tarjan, 1984) que um algoritmo bastante rápido).

⁸Esta informação pode ser distribuída como resultado de extensões aos protocolos de encaminhamento.

⁹A grande quantidade de informação que precisa de ser enviada faz com que o modelo FI não possa ser aplicado em situações práticas.

2. Modelo com Informação Completa

Neste caso é possível partilha inter pedido. O problema é formalizado como um problema de programação inteira com restrições quadráticas. A função objectivo minimiza a soma das duas parcelas seguintes: a soma dos custos dos ramos no CA (que é igual à soma das LB usadas pelo CA em cada um dos seus ramos) e a soma dos custos nos ramos no CR (determinados como sendo a LB adicional que é necessária reservar em cada ramos para *backup*).

3. Modelo com Informação Parcial

Neste caso é possível alguma partilha inter pedido. A formalização (função objectivo e restrições) para este caso pode ser igual ao caso anterior (modelo com FI) mas com os custos¹⁰ dos ramos do CR, utilizados na função objectivo, superiores ou iguais aos utilizados no caso anterior. Para a resolução do problema é apresentado um algoritmo que selecciona os caminhos através de um processo iterativo. O que passa pela construção (usando programação linear inteira) e resolução de formulações simplificadas do problema original. Em cada iteração são determinados os custos a associar a cada ramo (um para ser utilizado na determinação do CA e outro na determinação dos CR). Depois disso é necessário, em cada iteração, resolver o problema de encontrar dois caminhos disjuntos nos ramos, óptimos para os custos utilizados (determinados anteriormente). A resolução do problema deve minimizar a soma dos custos dos ramos no CA mais a soma dos custos nos ramos no CR. A solução do processo iterativo será a melhor solução obtida em todas as iterações.

O problema de encontrar dois caminhos disjuntos nos ramos, óptimos para os dois custos associados a cada ramo é um problema *NP-Hard*. Para obter a solução desse problema apresentam um algoritmo (solução heurística), baseado na resolução simultânea e iterativa de heurísticas para o primal e para o dual, obtendo sucessivos limites inferiores e superiores para o intervalo onde se encontra a solução. Pode-se assim ter alguma confiança que a solução admissível encontrada estará assim numa vizinhança de dimensão arbitrariamente definida do óptimo.

Embora nos modelos com informação parcial e com nenhuma informação, os caminhos de trabalho e de recuperação sejam calculados, no nó origem, sem informação completa (com informação parcial ou com nenhuma informação), é possível, reservar localmente, apenas a LB realmente necessária para recuperação.

¹⁰Definidos em Kodialam e Lakshman (2002b).

Para isso, ou seja para fazer reservas exactas de LB de protecção, é necessário que ao sinalizar o CR seja enviado o percurso do CA correspondente (informação que é gerada na sinalização do CA). Quando um CA protegido é redireccionado ou desligado o respectivo CR deve ser suprimido; a mensagem de terminação de um CR deve ser acompanhada da descrição do CA de forma a garantir a gestão correcta (da partilha) da LB de protecção. Essa informação deve ser mantida nos nós do CR, enquanto este estiver estabelecido.

Recuperação local – Algoritmo heurístico para selecção do CA e dos CR

- Modelo com Informação Parcial

O algoritmo principal é uma versão modificada do algoritmo de Dijkstra (1959). O algoritmo determina uma árvore de caminhos mais curtos, com raiz no destino do CA. A árvore pode não incluir todos os nós, pois o algoritmo termina quando o próximo nó a ficar com etiqueta permanente for o nó origem.

Em cada iteração do algoritmo é determinado o caminho de recuperação local de custo mínimo para todos os ramos que saem do nó que passou a estar etiquetado de forma permanente. Para determinar o caminho de recuperação de custo mínimo para contornar a avaria de um ramo (a, b) , é invocada uma sub-rotina que, iterativamente, determina todos os caminhos de recuperação de custo mínimo que contornam a avaria desse ramo, e que terminam num dos vários nós do caminho mais curto desde o nó a até ao nó destino. Para determinar cada um destes caminhos de custo mínimo é invocada uma nova sub-rotina, que também é uma versão modificada do algoritmo de *Dijkstra*.

O algoritmo entra em consideração com a partilha intra pedido e também com alguma partilha inter pedido no entanto, aparentemente, não entra em consideração com o custo do CA.

Para ser possível implementar o modelo com PI é necessário que o protocolo de encaminhamento anuncie a LB disponível e a LB reservada para protecção, em cada ramo. No OSPF e IS-IS já foram propostas extensões, para engenharia de tráfego, que abrangem parte desta informação.

Os algoritmos anteriores calculam e/ou reservam LB de protecção considerando que é possível fazer partilha inter pedido dessa LB. No entanto, actualmente os protocolos de sinalização existentes não permitem a possibilidade de

partilha de LB entre LSP, com caminhos explícitos, pertencentes a sessões diferentes. Os autores não falam acerca dos requisitos de sinalização necessários à implementação da partilha inter pedido de LB de protecção.

Em Qiao e Xu (2002) os autores propuseram um modelo PI, designado por DPIM (*Distributed Partial Information Management*), para protecção global. Este esquema é semelhante ao proposto em Kodialam e Lakshman (2000) (idêntico ao revisto em Kodialam e Lakshman (2002b)), mas utiliza informação adicional relacionada com a utilização de LB de protecção. Os resultados das simulações com redes com 15 e 70 nós apresentados em (Qiao e Xu, 2002) apontam para um desempenho significativamente superior de DPIM face ao algoritmo proposto para o modelo PI em (Kodialam e Lakshman, 2000).

Recuperação do LSP usando a abordagem baseada em CBR

No esquema de recuperação proposto em Dana *et al.* (2003) um LSP é recuperado redireccionando o seu tráfego para vários LSP pré-sinalizados, mas sem reserva de largura de banda.

Por cada caminho de trabalho é pré-estabelecido um conjunto de caminhos de recuperação. Quando ocorre a falha de um LSP protegido o seu fluxo de tráfego será dividido pelos LSP de recuperação correspondentes. Em Dana *et al.* (2003) é apresentada uma forma de obter as percentagens de tráfego que deve ser oferecido a cada LSP de recuperação utilizando uma abordagem baseada em *Constraint-based routing* (CBR) (Aamodt e Plaza, 1994).

A abordagem baseia-se na informação armazenada numa biblioteca de casos. Essa biblioteca contém um conjunto de casos, por LSP protegido. Cada um destes casos consiste numa situação possível para o valor da largura de banda requerida pelo caminho protegido, para os valores das larguras de banda disponíveis nos LSP de recuperação correspondentes e também na solução para essa situação (valores admissíveis para as percentagens de tráfego a atribuir a cada LSP de recuperação). Quando ocorre a falha de um caminho (os autores referem que são apenas consideradas falhas isoladas de ramos) é determinada a largura de banda necessária para o LSP que falhou e também a largura de banda disponível em cada LSP de recuperação. Para a situação actual vão ser procuradas situações semelhantes na biblioteca de casos. As soluções encontradas para situações semelhantes são combinadas através de uma técnica de adaptação por composição (Borner, 1994), para obter a solução que será utilizada para resolver o problema actual.

Duas considerações relativamente à abordagem baseada em CBR:

1. Os autores não demonstraram a qualidade das soluções obtidas, isto é, não mostram que os valores obtidos para as percentagens pela sua abordagem oferece uma maior eficiência em termos de utilização de recursos do que os valores obtidos por uma simples pesquisa de uma solução admissível na altura em que a falha ocorre. Parece-nos que isso dependerá fortemente da qualidade dos casos armazenados na biblioteca de casos e também da técnica de composição.
2. Se o número de casos armazenados for grande a obtenção de uma solução para o problema quando uma falha ocorre, consumirá demasiado tempo devido à dimensão do problema que tem de ser resolvido. E nesses casos uma pesquisa simples de uma solução que satisfaça as restrições poderia ter maior probabilidade de ser encontrada facilmente, principalmente se o espaço de soluções admissíveis for grande.

Distribuição da carga para protecção

O esquema proposto em Kim (2003), tal como o esquema de recuperação anterior, é baseado na possibilidade da carga poder ser distribuída por múltiplos caminhos existentes entre um mesmo par de LSR de entrada/saída. A distribuição de carga numa rede MPLS, segundo Kim, resulta numa maior utilização de recursos e melhor tolerância a falhas do que os sistemas com caminho único.

No artigo Kim (2003), a autora fez a análise numérica da distribuição de carga para a protecção do caminho em termos da relação entre a utilização de recursos e a fiabilidade do serviço, numa rede MPLS. A análise abrange dois mecanismos: um mecanismo de protecção com partilha total, designado por *Fully Shared Mechanism* (FSM), e um mecanismo de protecção com partilha parcial, designado por *Partially Shared Mechanism* (PSM).

Em ambos os mecanismos, quando é necessário fornecer um serviço, que solicita uma quantidade de largura de banda R , são procurados na rede os recursos disponíveis. Desta pesquisa resultam, para o serviço solicitado, n caminhos disjuntos cada um com largura de banda B (em que $nB \geq R$). Após os caminhos terem sido determinados, devem ser estabelecidos e a sua largura de banda reservada. Quando o tráfego é transmitido, existem mecanismos para detecção, notificação e recuperação de falhas.

Apesar dos caminhos de protecção serem pré-estabelecidos e com uma largura de banda garantida podem ainda ocorrer situações em que a recuperação não seja possível, pois os caminhos de recuperação podem falhar. No artigo a autora define fiabilidade como a “probabilidade de que as situações em que a recupera-

ção não é possível não existam” e a utilização de recursos y , como a “razão entre a quantidade de largura de banda pedida e a quantidade de largura de banda ocupada”. A quantidade ocupada é a soma da largura de banda dos caminhos de trabalho e de recuperação.

No mecanismo FSM, ilustrado na Figura 2.19, todos os n caminhos são simultaneamente caminhos de trabalho e caminhos de protecção. Inicialmente o tráfego é dividido pelos n caminhos sendo atribuída a fracção de tráfego R/n a cada um. Quando um caminho falha é atribuída, a cada um dos outros caminhos, a fracção de tráfego $R/(n - 1)$. Se ocorrer a falha de mais do que um caminho, aos caminhos válidos é sempre atribuída a fracção do tráfego pedido dividido igualmente por eles.

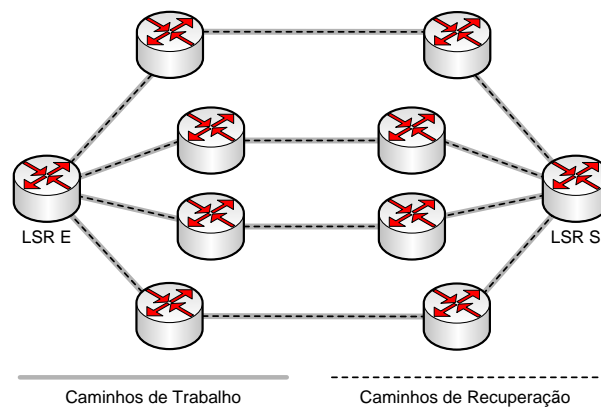


Figura 2.19: Exemplo do mecanismo FSM.

No mecanismo PSM, ilustrado na Figura 2.20, apenas alguns dos n caminhos são utilizados como caminhos de trabalho, sendo os restantes utilizados como caminhos de protecção. Quando ocorrer a falha de alguns caminhos de trabalho a fracção de tráfego desses caminhos é movida para os caminhos de protecção. Logo que ocorra a recuperação desses caminhos o tráfego que era transportado nos caminhos de recuperação “regressa” aos caminhos de trabalho originais.

Através de um estudo comparativo a autora concluiu que, nas condições consideradas (em que todos os caminhos têm a mesma probabilidade de falha), o mecanismo PSM é tão bom como o mecanismo FSM em termos de fiabilidade. No entanto, isto só é válido quando no mecanismo PSM o número de caminhos de protecção for igual a $\lfloor (1 - y)n \rfloor$, que é o número máximo de caminhos de protecção possível. Para ambos os mecanismos, apresentou soluções analíticas que permitem determinar a quantidade de largura de banda B que é necessário reservar de acordo com a largura de banda R e a fiabilidade pedida. No estudo foi considerado que a largura de banda solicitada pelo serviço é igualmente distri-

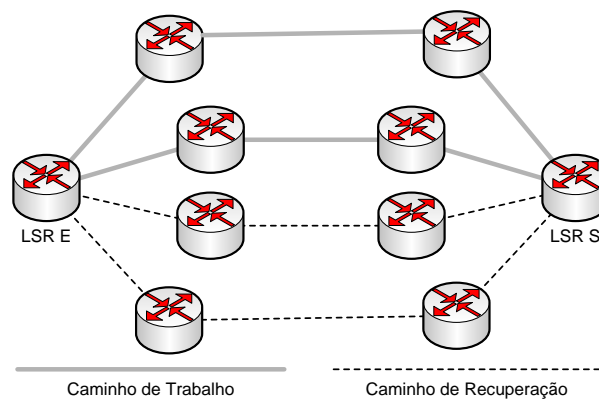


Figura 2.20: Exemplo do mecanismo PSM.

buída pelos vários caminhos de trabalho e que em todos os caminhos é reservada a mesma largura de banda. Os mecanismos com essas restrições não são muito realistas.

Protecção extremo-a-extremo com multi-caminho

Em Menth *et al.* (2004) os autores apresentam mecanismos de protecção por comutação e que utilizam os recursos de forma eficiente. Os mecanismos de protecção apresentados baseiam-se na utilização de caminhos de protecção extremo-a-extremo, e na utilização de múltiplos caminhos disjuntos pelos quais o tráfego é enviado simultaneamente (multi-caminho – *multi-path*). Pretendem minimizar a largura de banda a afectar aos ramos, dimensionando a rede para um determinado tráfego e otimizando a distribuição de carga através de algoritmos que correm em tempo polinomial (na prática, o encaminhamento e a largura de banda são otimizados simultaneamente).

Os autores propõem dois mecanismos, um dos quais designado por *Self-Protecting Multi-Paths* (SPM), que consiste na utilização de um multi-caminho, no qual é distribuído o tráfego. Se ocorre uma falha num dos caminhos o tráfego é distribuído pelos caminhos de trabalho restantes. O outro mecanismo, designado por protecção do caminho, tem como caminho de trabalho apenas um único caminho simples e um multi-caminho como protecção.

Foram comparadas três variantes do SPM que diferem na forma de distribuir a carga pelo multi-caminho: carga distribuída igualmente por todos os caminhos, factores de distribuição de carga inversamente proporcionais ao comprimento dos caminhos e optimização da distribuição de carga calculada através da resolução de um problema de programação linear para minimizar a capacidade extra requerida pela protecção.

O caminho primário, no mecanismo designado por protecção do caminho, pode ser determinado através da solução dos k caminhos disjuntos mais curtos *k-Disjoint Shortest Paths (k DSP)* ou através de um encaminhamento que minimize o fluxo de tráfego de trânsito em cada nó da rede. Cada uma destas duas possibilidades pode ser conjugada com diferentes formas de determinar os múltiplos caminhos de protecção: os caminhos de protecção podem ser calculados juntamente com um esquema apropriado de distribuição de carga através de resolução de problemas de programação linear; ou podem ser considerados os $k - 1$ caminhos disjuntos mais curtos. Neste último caso, as formas de distribuir a carga podem ser as mesmas três utilizadas no SPM.

Estas variantes foram comparadas pelos autores relativamente à capacidade necessária para protecção em cada uma delas. Para tal foram utilizadas diferentes tipos de matrizes de tráfego e várias topologias de rede. Perante estes testes, os autores afirmam que a capacidade extra necessária para oferecer fiabilidade contra todas as falhas isoladas de ramos ou LSR é apenas uma fracção da requerida pelo reencaminhamento OSPF. Verificaram também que a capacidade extra requerida depende da topologia da rede e em particular do número médio de caminhos disjuntos existentes, mas não da dimensão da rede.

O conceito geral dos mecanismos de protecção propostos (multi-caminhos) é semelhante ao dos mecanismos propostos em Kim (2003). Em Menth *et al.* (2004) são aprofundados estes conceitos nomeadamente na área da optimização do encaminhamento e de distribuição equilibrada da carga. A aplicação de algumas das abordagens a redes de grande dimensão parece difícil ou dependente de abordagens heurísticas ainda a desenvolver.

Recuperação rápida (dois esquemas)

Os esquemas de recuperação rápida (genericamente conhecidos por *Fast Reroute (FRR)*) utilizam o modelo protecção por comutação e em que a recuperação é efectuada pelo LSR mais próximo da falha.

Os dois esquemas do FRR são designados por *Facility backup* e *One-to-one backup* e foram normalizados recentemente, tendo sido o processo de normalização conduzido e documentado em Pan *et al.* (2005).

Os LSP afectados por uma falha (isolada de um ramo ou nó) são recuperados localmente e os caminhos de protecção são estabelecidos antes de ocorrer a falha. Estas características tornam possível redireccionar o tráfego para os caminhos de protecção em tempos da ordem das dezenas de milissegundos, o que vai de encontro às necessidades das aplicações em tempo real.

É possível aplicar o esquema de recuperação apenas a um subconjunto de LSP, os LSP sinalizados como a bandeira *local protection desired* activa. Esses LSP poderão ter protecção com largura de banda garantida, caso a mesma tenha sido solicitada.

O protocolo de sinalização utilizado por estes dois esquemas é o RSVP-TE (RFC 3209 de Awduche *et al.*, 2001) com um conjunto adicional de extensões especificadas em Pan *et al.* (2005).

One-to-one backup Neste esquema de recuperação é necessário um caminho de protecção (“Detour LSP”) em cada *Point of Local Repair* (PLR)¹¹ por cada LSP protegido, ou seja, cada LSR do LSP protegido terá que calcular um caminho de protecção com origem nesse nó e destino no LSR de saída desse LSP. A Figura 2.21 apresenta um exemplo que ilustra esta característica do esquema de recuperação. Na figura o caminho de trabalho com origem no LSR1 e destino no LSR4 (1-2-3-4) é protegido com três caminhos de protecção, pois é criado um caminho de protecção em cada PLR. Por exemplo o LSR 1 cria o caminho de protecção seguindo os LSR 1-5-6-7-4 para proteger o ramo 1-2 e o LSR 2.

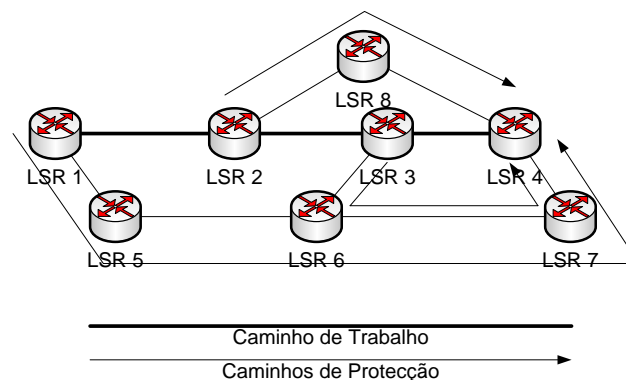


Figura 2.21: Exemplo com os vários caminhos de protecção de um LSP no *One-to-one backup*.

Na Figura 2.22 são ilustrados dois caminhos de trabalho seguindo os LSR 1-2-3-4-5-8 e 9-2-3-4-5-12, respectivamente e dois caminhos de protecção respectivos (caminhos seguindo os LSR 2-6-7-5-8 e 2-10-11-4-5-12). Repare-se que esses caminhos de protecção apenas protegem o ramo 2-3 e o LSR 3, para proteger outros recursos são necessários outros caminhos de protecção. É visível na figura que neste esquema os caminhos de protecção são criados por caminho de trabalho.

Como pode ser observado na Figura 2.21 e na Figura 2.22 este esquema tem a desvantagem de necessitar de um grande número de caminhos de protecção.

¹¹O RFC 4090 (Pan *et al.*, 2005) não utiliza a designação POR utilizando em seu lugar PLR.

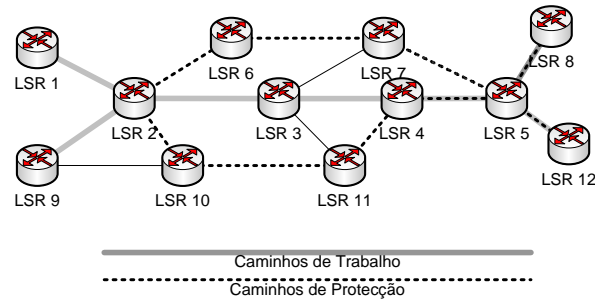


Figura 2.22: Exemplo de caminhos de protecção no *One-to-one backup*.

Esse número não é apenas função do número de caminhos protegidos mas é também função do comprimento desses caminhos (números de LSR). Para minimizar este aspecto é feita a fusão, quando possível, de vários caminhos de protecção de um LSP protegido, e possivelmente também com o LSP protegido. Em Pan *et al.* (2005) os autores definem várias regras para esse efeito. No exemplo da Figura 2.21 o LSR 6 detecta que existem dois LSP de protecção, para proteger o mesmo caminho de trabalho, e que portanto podem ser fundidos. Neste exemplo, os LSR nos dois caminhos de protecção entre o LSR 6 e o LSR 4 são os mesmos por isso não existe vantagem em escolher qualquer um dos dois para caminho resultante da fusão, mas no caso de serem diferentes o caminho resultante da fusão seria o caminho de protecção mais curto.

Cada "Detour LSP" pode ser recalculado e actualizado pelo PLR sempre que este achar necessário.

Facility backup Neste esquema é criado um *bypass tunnel* para proteger contra a falha de um ramo ou para proteger contra a falha de um nó. Na Figura 2.23 é configurado um *bypass tunnel* no LSR3 para proteger caminhos de trabalho que vão do LSR3 até ao LSR5 seguindo os ramos LSR3-LSR4 e LSR4-LSR5, contra a falha do ramo LSR3-LSR4 e do LSR4. Para qualquer uma destas falhas o PSL é o LSR3 e o PML é o LSR5.

No exemplo da Figura 2.23 aparece também representado um segundo *bypass tunnel* do LSR4 ao LSR5 para proteger caminhos de trabalho que utilizem o ramo LSR4-LSR5.

Na Figura 2.23 são apresentados exemplos dos dois tipos de *bypass tunnel* que é possível utilizar neste esquema de protecção, um *bypass tunnel* que protege contra a falha de um LSR (e necessariamente também de um ramo) e outro que protege apenas contra a falha de um ramo.

Este esquema de recuperação possui uma característica, muito importante em

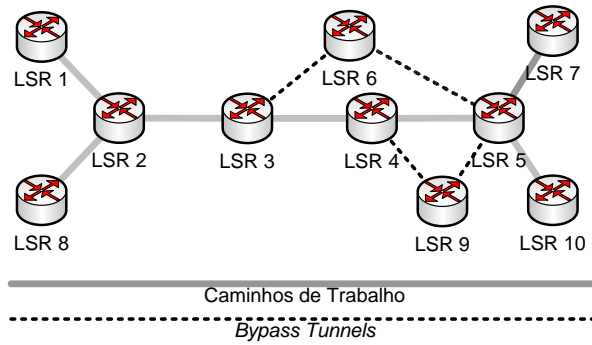


Figura 2.23: Exemplo de *bypass tunnels* na recuperação *Facility backup*.

termos de escalabilidade: o número de *bypass tunnels* necessários não é uma função do número de caminhos de trabalho. Um *bypass tunnel* pode proteger um conjunto de LSP que possuam restrições semelhantes em termos do caminho de protecção. Entre dois LSR, para garantir largura de banda de protecção aos caminhos de trabalho, poderá existir mais do que um *bypass tunnel*, dependendo da forma como são calculados e sinalizados. Por exemplo, o número de *bypass tunnel* entre dois LSR será elevado se for estabelecido um novo *bypass tunnel* cada vez que é sinalizado um novo LSP protegido.

2.4.2 Reencaminhamento

Caminho de recuperação pré-calculado

O esquema de recuperação proposto em Yoon *et al.* (2001) segue o modelo de recuperação por reencaminhamento local do tipo pré-calculado. Apresentam a sua proposta como sendo uma boa escolha quando um operador de rede pretende em simultâneo que o tempo de recuperação seja mínimo e que os recursos do caminho de recuperação sejam utilizados de forma o mais eficiente possível.

O caminho de recuperação é actualizado sempre que o LSR (que o calcula) recebe actualização da informação de estado da rede (utilização dos ramos). Nesse instante o caminho de recuperação calculado será óptimo de acordo com a informação do estado exacto dos recursos da rede. Assumem que as avarias possíveis são apenas falha ou degradação de um ramo e que se limitam a um único domínio de protecção MPLS.

Em cada caminho de trabalho, o LSR de entrada, e todos os LSR intermédios com capacidade de recuperação pré-calculam um caminho de recuperação para o LSR mais próximo a jusante. Perante uma falha, o LSR que faz a recuperação será o mais próximo a montante desta, se tiver capacidade de recuperação. Se

esse LSR não tiver capacidade de recuperação então será necessário enviar um FIS através da RNT¹². O LSR, que faz a recuperação, estabelecerá o caminho de recuperação, no instante em que detecta a falha ou no instante em que recebe a notificação da falha, utilizando o CR-LDP. Quando o problema no ramo for reparado, o LSR que recebe o FRS liberta o caminho de recuperação e repõe o tráfego no caminho de trabalho original. O diagrama de fluxo apresentado na Figura 2.24 resume o esquema proposto¹³.

Em cada LSR com capacidade de recuperação, existe um processo que actualiza os caminhos de recuperação, sempre que recebe uma actualização do estado da rede. Como isto implica a necessidade de troca de informação do desempenho da rede é necessário fazer uma extensão ao IGP utilizado (por exemplo ao protocolo OSPF ou ao protocolo IS-IS).

Os autores comparam o esquema que propõem com um esquema semelhante mas em que o caminho de recuperação pré-calculado mantém-se inalterado independentemente do estado de congestão actual da rede. Nos estudos simulacionais realizados concluíram, quando simularam a avaria de um ramo, que o número de pacotes perdidos assim como o número de pacotes que precisam ser ordenados é menor no esquema proposto.

O esquema proposto utiliza os recursos de forma eficiente e o tempo de recuperação é reduzido dado calcular o caminho de recuperação antes da falha ocorrer. No entanto a grande vantagem do esquema proposto relativamente aos esquemas que seguem o modelo de recuperação por reencaminhamento está na actualização constante do caminho de recuperação, tendo em consideração a utilização dos ramos.

Caminhos de recuperação de menor custo

No esquema de recuperação proposto em Ahn *et al.* (2002), o caminho de recuperação é determinado e estabelecido, depois da falha ocorrer, pelo LSR a montante que a detectou. Para caminho de recuperação é escolhido aquele que tiver menor custo, de entre todos os caminhos alternativos possíveis. O caminho de recuperação só será utilizado enquanto a falha não for reparada.

No artigo é definido PML-candidato como um LSR do caminho de trabalho que pode ser usado como PML. Através das mensagens de sinalização utilizadas

¹²Uma vez que os nós intermédios apenas calculam caminhos de recuperação (para um dado LSP) até ao nó adjacente a jusante, sempre que um nó é incapaz de resolver uma situação de falha o FIS será propagado até ao LSR de ingresso.

¹³O fluxograma da Figura 2.24 difere do fluxograma apresentado em Yoon *et al.* (2001) porque este não traduzia de forma completa o esquema aí proposto.

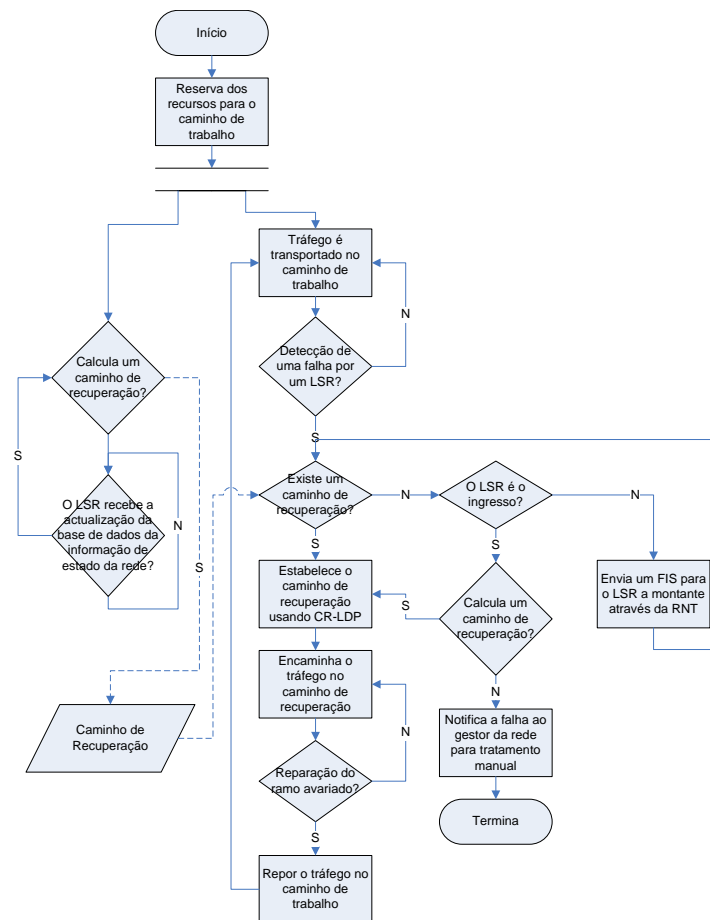


Figura 2.24: Diagrama de fluxo do esquema proposto em Yoon *et al.* (2001).

no estabelecimento dos caminhos de trabalho os LSR podem obter conhecimento dos PML-candidatos. Os caminhos alternativos possíveis para caminhos de recuperação são todos os caminhos entre o LSR a montante que detecta a falha e cada PML-candidato.

A grande vantagem do esquema proposto é a sua eficiência em termos de utilização de recursos pois utiliza o modelo de recuperação por reencaminhamento e escolhe para caminhos de recuperação os de menor custo.

Em Ahn *et al.* (2002) os autores explicam como o reencaminhamento local poderia ser feito usando o CR-LDP. Pensamos que este método também é viável usando o RSVP-TE, recorrendo à sinalização proposta em Gomes *et al.* (2005).

Recuperação hierárquica

Em Hong *et al.* (2004) os autores propõem um esquema de recuperação por reencaminhamento, cujos objectivos também são minimizar o tempo gasto na recu-

peração e maximizar a utilização de recursos da rede.

Para que o tempo gasto na recuperação seja minimizado o esquema de recuperação tenta inicialmente encontrar um caminho alternativo utilizando apenas um número reduzido de LSR, isto é, o âmbito da recuperação começa por ser limitado. O âmbito da recuperação é expandido quando não for possível calcular um caminho de recuperação razoável, que contorne a falha, dentro do âmbito anteriormente determinado.

O processo de criação de um caminho de recuperação é um processo iterativo que segue os seguintes passos:

1. Determinação de um grafo pesado, para o qual os autores apresentam um algoritmo. As entradas para esse algoritmo são: o grafo da rede, o local da falha e métricas de tráfego (largura de banda requerida, classe de tráfego e número de "saltos").
2. Determinação de um LSR que possa ser o PSL (pode ser um LSR arbitrário do caminho de trabalho entre o LSR de entrada e o LSR mais próximo a montante que detectou a falha), após o que o âmbito da recuperação fica delimitado.
3. Tentativa para encontrar um caminho alternativo óptimo do PSL até ao LSR de saída do caminho de trabalho, tendo em consideração os requisitos do tráfego do caminho de trabalho. Os autores apresentam também um algoritmo para a selecção dinâmica do caminho alternativo.

Os dois primeiros passos do processo determinam o âmbito da recuperação, dinamicamente, em termos topológicos. De forma a que o âmbito da recuperação comece por ser o mais apertado possível, no passo dois do processo, o PSL seleccionado deve ser o LSR do caminho de trabalho, mais próximo da falha no sentido a montante, que tenha essa possibilidade. Nas iterações seguintes o PSL passa a ser um LSR do caminho de trabalho que se vai aproximando, em cada iteração, do LSR de entrada do caminho de trabalho. Assim, o âmbito da recuperação começa por ser uma pequena sub-rede delimitada pelo LSR de saída e pelo LSR adjacente à falha (a montante), se este LSR puder ser o PSL. Nas próximas iterações o âmbito da recuperação vai sendo expandido dinamicamente em função do local da falha e do estado da rede. O esquema é designado por esquema de recuperação hierárquica devido a este aspecto de expansão gradual do âmbito de recuperação. O âmbito da recuperação pode ser expandido até englobar toda a rede. A Figura 2.25 mostra um exemplo de expansão do âmbito

da recuperação. Assume-se que o ramo assinalado falhou. Considerando que não foi possível encontrar um caminho de recuperação dentro do âmbito de recuperação **A** então o âmbito de recuperação foi expandido. Considerando que também não foi possível encontrar um caminho de recuperação dentro do âmbito de recuperação **B** então o âmbito de recuperação foi novamente expandido. O âmbito de recuperação **C** é a última possibilidade para encontrar um caminho de recuperação pois o LSR responsável pela recuperação coincide com o LSR de entrada.

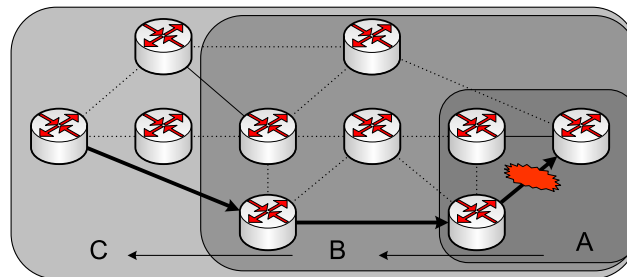


Figura 2.25: Exemplo de expansão do âmbito da recuperação adaptada de (adaptada de Hong *et al.*, 2004).

Antes de uma nova iteração o PSL anteriormente determinado é definido implicitamente como um nó com avaria (*implicit abnormal node*) o que conduz a que o âmbito da recuperação seja alargado. O processo iterativo termina quando o algoritmo conseguir encontrar um caminho alternativo entre o PSL e o LSR de saída. Se o PSL for o LSR de entrada do caminho de trabalho, significa que o âmbito foi expandido até ao limite.

Logicamente quantas mais iterações forem efectuadas mais lenta será a recuperação mas, por outro lado, maior o âmbito da recuperação, o que melhora a utilização dos recursos.

O caminho de recuperação encontrado é estabelecido utilizando um protocolo de sinalização e é libertado quando a avaria for reparada. O esquema proposto abrange a recuperação de ramos ou nós e não tem o problema da falha simultânea do caminho de trabalho e do caminho de recuperação, como acontece nos esquemas de protecção por comutação.

Em geral, a eficiência deste esquema será tanto maior quanto mais limitado for o âmbito da recuperação. De facto, se for preciso aumentar o âmbito diversas vezes, torna-se necessário repetir cálculos e trabalhar com redes de maior complexidade. No entanto, como os próprios autores referem, a escolha de um âmbito muito reduzido levanta problemas de utilização de recursos por ignorar capacidades disponíveis noutras zonas da rede (problemas aumentados pela de-

cisão de efectuar sempre a recuperação entre a falha e o fim do caminho, o que implica concentração de recursos na área da rede próxima do fim do caminho).

No método proposto, quando um PSL não consegue determinar um caminho de recuperação define-se esse PSL como um nó com avaria (*implicit abnormal node*), isto implica a alteração da sinalização para garantir que não é difundida informação errada do estado da rede. Pensamos no entanto que tal não é necessário, desde que um nó ao receber um FIS, com a indicação da localização da falha inferisse que todos os nós intermédios entre si próprio e o nó adjacente à falha não devem ser usados no caminho de recuperação a calcular – isto implica que cada LSR tem de conhecer o caminho total a recuperar.

Protecção dinâmica do caminho

Em Park *et al.* (2004) os autores propõem um mecanismo de protecção dinâmica do caminho em redes MPLS que permite recuperar rapidamente a falha num ramo e/ou nó. No esquema proposto, quando um LSR detecta uma falha, ele selecciona um caminho de recuperação apropriado de entre os caminhos de trabalho já estabelecidos. O caminho seleccionado é um caminho que tenha o mesmo destino que o caminho que falhou. Se o LSR, local à falha, encontrar vários caminhos para o esse destino então a escolha é feita seguindo os critérios indicados em Park *et al.* (2004).

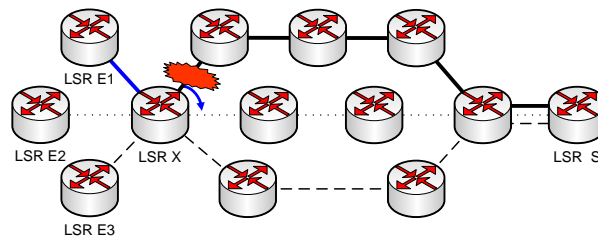


Figura 2.26: Rede exemplo com três caminhos de trabalho estabelecidos.

A Figura 2.26 apresenta um cenário de falha num dos ramos em um dos três caminhos de trabalho estabelecidos. Os LSR de entrada dos três caminhos de trabalho são respectivamente o LSR E1, o LSR E2 e o LSR E3 e o LSR S é o LSR de saída comum. Considerando que ocorreu a falha indicada na figura, o LSR X ao verificar que possui dois caminhos estabelecidos para o mesmo destino, selecciona um deles e redirecciona o tráfego do caminho que falhou para ele.

Este esquema é rápido porque não requer a sinalização de um novo caminho, mas apenas alterações nas tabelas no LSR que redirecciona o tráfego para um LSP existente. Trata-se de uma forma de implementação do método pré-qualificado.

No entanto, pode acontecer que o LSR não possua outro caminho estabelecido para o mesmo destino do caminho que falhou. Nesta situação o LSR estabelece localmente um novo caminho com origem em si e com o destino pretendido. Se o caminho de recuperação falhar, o esquema de protecção selecciona outro caminho de recuperação seguindo o mesmo procedimento utilizado na selecção do caminho de recuperação anterior.

Em Park *et al.* (2004) os autores afirmam que o mecanismo não requer extensões aos protocolos de sinalização.

É discutível se o mecanismo poderá ser usado muitas vezes pois o número de caminhos alternativos pré-estabelecidos para o mesmo destino poderá ser reduzido. Outro problema é saber o que é que se faz com o tráfego que utilizava o caminho estabelecido que agora também é caminho de recuperação. Provavelmente se for mantido no mesmo caminho sofrerá uma degradação.

Redireccionamentos sucessivos podem ser utilizados, segundo Park *et al.* (2004), para lidar com falhas múltiplas. No entanto, esses sucessivos redireccionamentos, poderão dar origem a caminhos com ciclos os quais contribuirão para a degradação do funcionamento da rede.

2.4.3 Conclusões gerais acerca dos esquemas analisados

A Tabela 2.4 resume, para os esquemas estudados, o modelo de recuperação utilizado e o âmbito da recuperação em termos topológicos.

No trabalho em Kodialam e Lakshman (2002b) são apresentados algoritmos (distintos) para recuperação global ou local, pelo que na Tabela 2.6 é dito que o algoritmo suporta essas duas opções.

Para além dos apresentados na Tabela 2.4, vários outros aspectos estão envolvidos na recuperação. Nesta secção vamos começar por lembrar alguns desses aspectos, e referir para cada um deles (os esquemas apresentados nas subsecções 2.4.1 e 2.4.2) os aspectos a que os respectivos autores deram ênfase especial.

A escolha dos **LSR entre os quais é estabelecido o caminho de recuperação** é uma característica do esquema. Existem esquemas em que o LSR de entrada e o LSR de saída são o PSL e o PML respectivamente (recuperação global e extremo a extremo), noutros esquemas o PSL é o LSR mais próximo da falha a montante e o PML é o LSR mais próximo da falha a jusante (recuperação local). Entre estes dois extremos existem também vários outros esquemas, são exemplo destes os propostos em Ahn *et al.* (2002) e Hong *et al.* (2004).

Quando o PSL não é o LSR que detecta a falha, é necessário fazer-lhe chegar uma **mensagem de notificação da falha**, o trabalho em Huang *et al.* (2002) é

| 1º autor do esquema | modelo de recuperação | reparação local/global |
|---------------------|-------------------------|---|
| Haskin | protecção por comutação | local |
| Huang | " | global |
| Mellah | " | local |
| Kang | " | local |
| Bartoš | " | global, transitoriamente faz a recuperação local do tráfego recebido no LSR que detecta a falha |
| Yetginer | " | global |
| Kodialam | " | global ou local |
| Dana | " | global |
| Kim | " | global |
| Menth | " | global |
| Pan (2 esquemas) | " | local |
| Yoon | reencaminhamento | local |
| Ahn | " | local |
| Hong | " | local |
| Park | pré-qualificado | local |

Tabela 2.4: Âmbito da recuperação em termos topológicos.

centrado neste aspecto.

Como é sabido, nos esquemas de recuperação por reencaminhamento o caminho de recuperação só é estabelecido quando a falha ocorre. No entanto, o **instante do cálculo/actualização do caminho de recuperação** pode ocorrer em diferentes alturas antes da falha, o que fornece a base para o esquema proposto em Yoon *et al.* (2001).

Uma **recuperação extremamente rápida** foi o objectivo em Pan *et al.* (2005), ao conjugar o modelo de protecção por comutação com a recuperação local.

Em muitos dos esquemas de recuperação é estabelecido um caminho de recuperação para recuperar uma ou várias falhas do caminho de trabalho. É contudo possível **utilizar vários caminhos de recuperação simultaneamente** com distribuição de carga entre eles. A distribuição de carga pode ocorrer não só em caminhos de recuperação mas também em caminhos de trabalho. Em Kim (2003), Menth *et al.* (2004) e Dana *et al.* (2003) os autores apresentaram propostas com base nesta possibilidade.

Outro aspecto é a utilização de um caminho/caminhos de recuperação por cada caminho de trabalho ou a utilização de **um caminho de recuperação para recuperar simultaneamente vários caminhos de trabalho**. Exemplo de um esquema que utiliza um só caminho para este último objectivo é o proposto em Mellah e Mohamed (2003) e mais recentemente o *Facility backup* em Pan *et al.*

(2005).

Outra preocupação é **determinar o algoritmo de encaminhamento** a utilizar no cálculo dos caminhos de recuperação. Trabalhos focados nesta questão são os trabalhos em Bartoš e Raman (2001), Yetginer e Karasan (2002) e Kang e Reed (2003), este último dedicado ao caso especial do encaminhamento em *bypass tunnels*. É comum serem apresentados algoritmos que calculam os caminhos de recuperação e de trabalho de forma integrada, como em Yetginer e Karasan (2002) e em Kodialam e Lakshman (2002b).

Apresentamos a seguir uma abordagem resumida da forma como cada um dos aspectos anteriores é **tratado** por um dado esquema de recuperação em particular, escolhendo para cada um desses aspectos o esquema que o explorou em detalhe. Os esquemas são apresentados neste resumo geralmente pela mesma ordem que foi utilizada nas subsecções 2.4.1 e 2.4.2.

A característica essencial do esquema proposto em Haskin e Krishnan (2001) é a forma como é construído o caminho de recuperação. Este caminho além de um segmento de recuperação global possui também um segmento de recuperação no sentido inverso ao do caminho de trabalho. Este último segmento é utilizado, pelo LSR que detecta a falha, para transmitir de volta para o LSR de entrada os pacotes (que não conseguiu fazer chegar ao destino), conseguindo deste modo reduzir o número de pacotes perdidos. Apesar de neste esquema a recuperação ser global, devido à forma de construção do caminho de recuperação não é necessário enviar uma mensagem de notificação da falha, pois é o LSR que a detecta que faz a sua recuperação.

O objectivo principal do esquema proposto em Huang *et al.* (2002) é minimizar o atraso na propagação das mensagens de notificação da falha. Para atingir esse objectivo definiram uma estrutura para o envio das mensagens de notificação rápida e eficiente, um mecanismo de transporte leve para essas mensagens de notificação e um mecanismo para detecção de falhas.

O esquema de recuperação em Mellah e Mohamed (2003) utiliza um algoritmo de protecção local baseado na utilização de *bypass tunnels* pré-estabelecidos localmente por cada LSR para contornar a avaria de um ramo. O trabalho apresentado em Kang e Reed (2003) também se baseia na utilização de *bypass tunnels*, mas estes autores focam o seu estudo no problema do encaminhamento nesses túneis e apresentam uma resolução recorrendo à utilização de “*p-cycles*”. Um dos esquemas de protecção, já normalizado em Pan *et al.* (2005), também é baseado na utilização de *bypass tunnels* mas ao contrário dos anteriores, em que os *bypass tunnels* só permitem a recuperação de ramos, aqui os *bypass tunnels* também permitem a recuperação de falhas em LSR. Além deste esquema de *Fast Reroute*,

é também apresentado em Pan *et al.* (2005) um outro em que são estabelecidos caminhos de recuperação por cada caminho de trabalho.

O esquema proposto em Bartoš e Raman (2001) é também um esquema de protecção por comutação que utiliza recuperação global, mas que possui a particularidade de conseguir recuperar localmente o tráfego temporário que é recebido no LSR que detecta a falha, enquanto esse tráfego não for *switched over* para o caminho de protecção global. O realce do esquema vai para o algoritmo que propuseram para calcular os caminhos de protecção, que determina dois caminhos de protecção por cada LSR que oferece recuperação. Os dois caminhos de protecção são posicionados de forma que a falha de um ramo não causa a perda de conectividade simultânea em ambos. O algoritmo possui a vantagem da determinação os caminhos de protecção sem considerar a localização dos caminhos de trabalho. Esta característica do algoritmo permite a possibilidade de partilha de ramos entre caminhos de protecção e caminhos de trabalho, mas devido à forma como os dois caminhos de protecção são construídos é garantido que não existem situações em que a recuperação não seja possível.

O aspecto central explorado em Yetginer e Karasan (2002) é a forma como são determinados os caminhos. Os autores apresentam algoritmos para quatro abordagens para o processo de determinação *off-line* dos caminhos de trabalho e dos caminhos de recuperação, que vão desde a determinação dos caminhos separadamente até à integração do cálculo dos caminhos de trabalho e de recuperação. Todos os algoritmos para a determinação dos caminhos são formulados como problemas de programação linear inteira. Outra proposta de algoritmos para o cálculo dos caminhos de trabalho e de recuperação foi apresentada em Kodialam e Lakshman (2002b). Mas, enquanto que em Yetginer e Karasan (2002) o cálculo é feito *off-line* para todos os pedidos, em Kodialam e Lakshman (2002b) o cálculo é feito *on-line* pedido a pedido.

Foram revistos três esquemas (Dana *et al.*, 2003; Kim, 2003; Menth *et al.*, 2004) baseados na possibilidade da distribuição dos pacotes de um fluxo de tráfego por vários caminhos existentes entre um par de LSR de entrada/saída. Em Kim (2003) e Menth *et al.* (2004) são apresentados dois mecanismos de distribuição de carga em cada, um dos quais é comum a ambos e consiste na utilização de múltiplos caminhos que são simultaneamente caminhos de trabalho e caminhos de protecção¹⁴. Nesse mecanismo em situações normais o tráfego é distribuído igualmente pelos vários caminhos estabelecidos. Quando um ou vários caminhos falham, o tráfego afectado é distribuído igualmente pelos caminhos válidos rema-

¹⁴A semelhança reduz-se apenas a esta forma de utilização dos caminhos!

nescentes. No outro mecanismo apresentado em Kim (2003) alguns caminhos do conjunto total são utilizados apenas como caminhos de trabalho e os restantes são utilizados apenas como caminhos de protecção. Neste mecanismo quando ocorrer a falha de alguns caminhos de trabalho a fracção de tráfego desses caminhos é movida para os caminhos de protecção. O caso particular em que é utilizado apenas um caminho de trabalho e vários caminhos de protecção corresponde ao segundo mecanismo apresentado em Menth *et al.* (2004). Para estas situações, ou seja, quando por cada caminho de trabalho é pré-estabelecido um conjunto de caminhos de protecção, em Dana *et al.* (2003) os autores apresentam um modo de obter as percentagens de tráfego que deve ser oferecido a cada LSP de protecção utilizando uma abordagem baseada em CBR (Aamodt e Plaza, 1994).

Em Yoon *et al.* (2001) os autores propuseram um esquema de recuperação por reencaminhamento que minimiza o tempo de recuperação e aparentemente com boa taxa de sucesso. O factor de decisão é o instante em que se deve determinar o caminho de recuperação. Para conseguir o primeiro objectivo o caminho de recuperação deve ser pré-calculado, mas para o segundo é necessário que o caminho de recuperação seja óptimo de acordo com o estado da rede na altura que a falha ocorre. Portanto, o esquema que propuseram, pré-calcula o caminho de recuperação e actualiza-o sempre que obtém actualização da informação do estado da rede.

Nos esquemas de recuperação propostos em Ahn *et al.* (2002) e Hong *et al.* (2004), o aspecto mais relevante é a escolha dos LSR entre os quais deve ser estabelecido o caminho de recuperação, quando ocorrer uma falha. Em Ahn *et al.* (2002) os autores escolheram sempre para PSL o LSR mais próximo a montante que detectou a falha mas, para PML permitem que seja usado qualquer LSR entre o LSR mais próximo da falha a jusante e o LSR de saída, e escolhem aquele LSR que permita obter o caminho de recuperação de menor custo. Por outro lado, em Hong *et al.* (2004) os autores escolheram para PML sempre o LSR de saída do caminho de trabalho. Para PSL começam por escolher o LSR a montante mais próximo da falha, e vão-se afastando da falha em direcção ao LSR de entrada até encontrar um caminho de recuperação que contorne a falha.

As Tabelas 2.5 e 2.6 apresentam um resumo das características dos esquemas de recuperação descritos.

Como sumário final podemos dizer que a grande maioria dos esquemas de recuperação propostos utilizam o modelo de protecção por comutação (considerando apenas os esquemas analisados) e portanto a grande maioria dos esquemas de recuperação apenas se propõem recuperar de falhas isoladas. Para os esquemas que utilizam o modelo de recuperação por reencaminhamento esta-

| 1º autor do esquema | LSR que faz a recuperação | Orientação | Falhas consideradas | Aspecto essencial |
|---------------------|---|------------------|--------------------------|---|
| Haskin | LSR que detecta a falha | Caminho | Isoladas de ramos ou nós | Forma de construção do caminho de recuperação (constituído por 2 segmentos) |
| Huang | LSR de entrada do segmento a recuperar | Caminho | Isolada de nós ou ramos | Notificação e detecção da falha |
| Mellah | LSR que detecta a falha | Recurso | Isoladas de ramos | Utilização de <i>bypass tunnels</i> |
| Kang | LSR que detecta a falha | Recurso | Isoladas de ramos | Encaminhamento dos <i>bypass tunnels</i> |
| Bartoš | É o LSR de entrada mas inicialmente é o LSR que detecta a falha | Recurso /Caminho | Isoladas de ramos | Algoritmo de cálculo do caminho de recuperação |
| Yetginer | LSR de entrada | Caminho | Isoladas de ramos | Algoritmos de cálculo <i>off-line</i> dos caminhos de trabalho e de recuperação |
| Kodialam | LSR de entrada ou LSR que detecta a falha | Caminho | Isoladas de ramos ou nós | Algoritmos de cálculo <i>on-line</i> dos caminhos de trabalho e de recuperação |
| Dana | LSR de entrada | Caminho | Isoladas de ramos | Distribuição de carga |
| Kim | LSR de entrada | Caminho | Caminhos | Distribuição de carga |
| Menth | LSR de entrada | Caminho | Isoladas de ramos ou nós | Distribuição de carga |
| Pan | LSR que detecta a falha | Recurso | Isoladas de ramos ou nós | Recuperação rápida – Utilização de <i>bypass tunnels</i> |
| Pan | LSR que detecta a falha | Caminho | Isoladas de ramos ou nós | Recuperação rápida – Utilização de <i>detours</i> |

Tabela 2.5: Protecção por Comutação.

mos perante situações diversas. O esquema proposto em Yoon *et al.* (2001), uma vez que faz uso de caminhos pré-calculados para a recuperação, pode ter problemas no caso de falhas múltiplas. Já Ahn *et al.* (2002) os autores consideram que o seu esquema recupera de falhas múltiplas uma vez que se trata de um esquema de recuperação por reencaminhamento em que o caminho de recuperação não está pré-calculado; e em Hong *et al.* (2004) os autores tratam explicitamente com a possibilidade de haver falhas múltiplas simultâneas. Em Park *et al.* (2004) os autores afirmam que o seu esquema pode ser aplicado a redes sujeitas a falhas múltiplas. Tal só será possível se, existindo vários caminhos alternativos para o destino a partir do LSR que detectou a falha, algum destes não tiver falhado. Nos esquemas analisados, o LSR que faz a recuperação é sempre o PSL e nunca o PML, por conseguinte quando nas tabelas se diz que a recuperação é feita pelo LSR que detecta a falha, deve subentender-se o LSR a montante da falha. A recuperação é classificada como **orientada ao recurso** (ramo ou LSR) se tentar a

| 1º autor do esquema | Modelo de recuperação | LSR que faz a recuperação | Orientação | Aspecto essencial |
|---------------------|-----------------------|--|------------|--|
| Yoon | Reencaminhamento | Pode ser o LSR que detecta a falha ou o LSR de entrada, depende da situação encontrada | Caminho | Manter o caminho de recuperação óptimo pré-calculado |
| Ahn | Reencaminhamento | LSR que detecta a falha | Caminho | Forma de determinar o caminho de recuperação |
| Hong | Reencaminhamento | Pode ser qualquer LSR entre o LSR que detecta a falha e o LSR de entrada, depende da situação encontrada | Caminho | Forma de determinar o caminho de recuperação |
| Park | Pré-qualificado | LSR que detecta a falha | Caminho | Como escolher o caminho a utilizar na recuperação |

Tabela 2.6: Recuperação por Reencaminhamento (e pré-qualificado).

recuperação de cada recurso da rede, independentemente dos caminhos de trabalho estabelecidos, ou **orientada ao caminho**, se tentar explicitamente recuperar o caminho de trabalho. No caso do esquema em Bartoš e Raman (2001) consideramos que a falha é orientada ao recurso/caminho¹⁵ uma vez que recupera localmente o tráfego temporário que é recebido no LSR que detecta a falha e posteriormente usa caminhos de protecção global. Dos esquemas apresentados três são orientados ao recurso (dos quais dois são apenas orientados ao ramo), 12 esquemas são orientados ao caminho e um é orientado simultaneamente ao recurso e ao caminho.

Ao longo desta revisão foi chamada a atenção para o facto de alguns destes esquemas requererem extensões de sinalização, as quais nem sempre são referidas pelos respectivos autores.

A grande maioria dos autores dos esquemas de recuperação apenas apresentam a preocupação de recuperar falhas isoladas de ramos ou de ramos e nós. Embora os autores apresentem justificação para tal, estas justificações não tomam em consideração o facto de muitas redes poderem ser organizadas através de SRLG, o que implica que falhas múltiplas podem ser bastante prováveis. No entanto, nos esquemas em que a recuperação é global mas as falhas múltiplas que ocorram se confinem aos recursos de um mesmo caminho de trabalho, também será possível, em princípio, a sua recuperação.

Embora os esquemas analisados até agora tratem o tráfego sem diferenciação, na prática pode ser conveniente distinguir o tráfego também na recuperação. Em Ricciato *et al.* (2003) é proposto um esquema que fornece algum tipo de trata-

¹⁵No entanto, gostaríamos de chamar a atenção para o facto deste esquema ao proteger um caminho está na realidade a proteger todos os caminhos entre um dado LSR de entrada e um dado LSR de saída.

mento diferenciado de protecção, mas apenas no que se refere a garantias de LB. Em Autenrieth e Kirstadter (2002) os autores mostram que a diferenciação de opções de recuperação para os fluxos de tráfego permite fornecer um uso de recursos mais reduzido. Neste trabalho, várias combinações de mecanismos de recuperação (por exemplo, protecção versus reencaminhamento, âmbito local versus global), foram analisados no respeitante ao uso de LB. Outras medidas foram propostas num estudo semelhante (Dong *et al.*, 2003) para redes ópticas e *Generalized Multiprotocol Label Switching* (GMPLS). Em Calle (2004) é proposto um esquema que prevê uma escolha explícita de mecanismos de protecção para diferentes classes de tráfego. Esta obra também apresenta a possibilidade de usar mecanismos alternativos de protecção para a mesma classe de tráfego, e descreve o desempenho esperado e o observado desses mecanismos. No entanto, a indicação operacional para escolher entre os mecanismos de protecção (calculada através de um algoritmo analítico, baseado na perda de pacotes, tempo de recuperação e consumo de recursos) é limitada à protecção usando três diferentes âmbitos de recuperação. De acordo com Calle (2004), incluir outros modelos de recuperação nesta abordagem pode levar a problemas de escalabilidade.

2.5 Conclusões

Foram revistas as técnicas relativas à recuperação que podem ser usadas nas redes MPLS bem como propostas de esquemas de recuperação do tipo protecção por comutação e do tipo recuperação por reencaminhamento. Além de apresentar as características mais relevantes do funcionamento destes esquemas foram ainda tecidas algumas considerações acerca da sua aplicabilidade em redes reais. Finalmente foi apresentada uma resenha comparativa das suas características mais relevantes.

Do trabalho realizado intui-se que o estudo da possibilidade de utilização de diversos esquemas de recuperação simultaneamente, utilizando critérios de escolha do esquema em cada instante (por exemplo os tipos de serviços pedidos e o estado da rede) tendo em vista a conjugação das suas melhores características, é uma área a explorar. Assim no Capítulo 4 surge uma proposta de recuperação que por um lado tira partido da articulação entre esquemas (conhecidos), incorporando alguma(s) variante(s) de esquemas descritos na literatura.

3 Partilha de Largura de Banda de Protecção

Neste capítulo são apresentados algumas questões relativas à determinação dos caminhos activos e de recuperação, para a recuperação em MPLS em situações em que se pretende partilhar a LBP. Deve ser mencionado que a partilha que é proposta pressupõe a inexistência de falhas simultâneas.

Neste capítulo são descritas algumas melhorias que podem ser efectuadas a esquemas de partilha de LBP. Assim, na Secção 3.1 é introduzida a nomenclatura utilizada e descrito um método para reserva mínima de LBP na protecção local. Na Secção 3.2 é revisto um esquema de protecção local e são propostas correcções e alterações que culminam numa nova versão; todas as alterações introduzidas são devidamente justificadas sendo descrito em detalhe o seu impacto no funcionamento do algoritmo. Finalmente na Secção 3.3 é descrito o problema de obtenção de um par de caminhos disjuntos de custo mínimo numa rede com custos duais nos arcos e ainda o problema da obtenção do conjunto de todos esses pares, sendo apresentados algoritmos para a sua resolução.

As Secções 3.1 e 3.2 baseiam-se em Jorge e Gomes (2006c, 2007). O texto da Secção 3.3 baseia-se nos trabalhos Gomes *et al.* (2006a,b,c).

3.1 Considerações sobre a partilha de LB de protecção

A partilha de LBP é um problema cuja resolução óptima depende da quantidade de informação disponível. Como já referido na Secção 2.4.1, a informação disponível para fazer a partilha pode ser classificada nos cenários denominados de *No Information* (NI), *Partial Information* (PI) ou *Full Information* (FI). Em resumo, podemos dizer que o modelo NI não assume mais informação sobre cada ramo do que a LB disponível nele, o modelo FI assume que todos os pontos que tem de tomar as decisões possuem toda a informação sobre todos os CA e CR na rede (e a relação entre estes) e o modelo PI assume apenas a distribuição pela rede de informação agregada dos ramos (informação para cada ramo sobre quanta LB é reservada neste para CA e quanta para CR).

Em redes reais com alguma dimensão, o modelo PI representa o melhor compromisso entre sobrecarga de sinalização e quantidade de partilha de LB de protecção, ou LBP. Para aumentar a partilha de LBP serão combinadas duas estratégias: uma melhor selecção do CA e do CR, baseada numa estimativa melhor da LBP adicional necessária nos CR, e reserva exacta da LB de protecção. Em Qiao e Xu (2002) foi apresentada uma proposta de uma estimativa para a LBP adicional necessária pelo CR de um CA, no contexto de protecção global. Ir-se-à apresentar uma extensão dessa estimativa para protecção local. Esta estimativa apresenta um valor mais preciso que o usado em Kodialam e Lakshman (2001), e resultará na redução da taxa de rejeição de pedidos. Por seu lado a utilização de reservas exacta conduz a uma redução da LBP usada (em cada pedido).

3.1.1 Notação

Utilizar-se-à, principalmente a notação usada em Kodialam e Lakshman (2001) e Qiao e Xu (2002). Considere-se $G' = (V, E')$, o grafo da rede, onde V é o conjunto de nós e E' o conjunto de ramos dirigidos. São indicados em seguida um conjunto de símbolos que serão usados nesta secção.

- $(i, j) \in E'$: representa um ramo dirigido de cabeça j e cauda i , $i, j \in V$.
- \mathcal{A} : conjunto dos ramos de um CA.
- C_{ij} : capacidade do ramo (i, j) .
- b_k : LB necessária ao pedido k , em cada arco do CA.
- \mathcal{A}_{ij} : conjunto dos pedidos que utilizam o ramo (i, j) no CA.
- F_{ij} : total da LB reservada para os pedidos que utilizam o ramo (i, j) no CA, $F_{ij} = \sum_{k \in \mathcal{A}_{ij}} b_k$.
- \mathcal{B}_{ij} : conjunto dos pedidos que utilizam o ramo (i, j) no CR.
- B_{ij} : total da LB reservada para proteger ramos que utilizam o ramo (i, j) nos respectivos CR.
- R_{ij} : LB residual no ramo (i, j) : $R_{ij} = C_{ij} - F_{ij} - B_{ij}$.
- $\mathcal{S}_{ij}^{uv} = \mathcal{A}_{ij} \cap \mathcal{B}_{uv}$: conjunto dos pedidos que utilizam o ramo (i, j) no CA e que utilizam o ramo (u, v) no CR, onde $(i, j), (u, v) \in E'$.
- $S_{ij}^{uv} = \sum_{k \in \mathcal{S}_{ij}^{uv}} b_k$: total da LB necessária aos pedidos em \mathcal{S}_{ij}^{uv} .

- δ_{ij}^{uv} : LBP adicional necessária no ramo (u, v) para o poder utilizar como parte de um CR para um pedido novo que utiliza o ramo (i, j) no CA. Este valor depende do modelo de informação considerado e, como salientado em Qiao e Xu (2002), no método de estimação da LBP.
- θ_{ij}^{uv} : custo de utilizar o ramo (u, v) no CR de um pedido novo que utilize o ramo (i, j) no CA.
- $P_B^{uv} = \{S_{ij}^{uv} | (i, j) \in E'\}$: um vector que contém a lista de valores S_{ij}^{uv} . Indica a quantidade de LBP no ramo (u, v) que é usada para proteger de uma falha em qualquer outro ramo da rede.
- $P_{B_{uv}} = \max_{ij} S_{ij}^{uv}$: o maior dos elementos do vector P_B^{uv} . É a LBP mínima necessária no ramo (u, v) para proteger todos os CA.
- $P_A^{ij} = \{S_{ij}^{uv} | (u, v) \in E'\}$: um vector que contém a lista de valores S_{ij}^{uv} . Indica a quantidade de LB do ramo (i, j) que é protegida em qualquer outro ramo da rede.
- $P_{A_{ij}} = \max_{uv} S_{ij}^{uv}$: o maior dos elementos do vector P_A^{ij} . É a LBP suficiente no ramo (u, v) para proteger todos os CA.
- λ_{uv} : quantidade de LB de protecção disponível no ramo (u, v) , para o pedido actual (considerando que já foram calculados outros CR para esse pedido). O nó origem do pedido conhecerá exactamente esta quantidade, para todos os ramos (u, v) , quando tiver calculado o CA e os CR.

No modelo PI, a minimização da LBP é baseada numa estimativa da quantidade necessária de LBP no ramo (u, v) para proteger todos os CA. Em Qiao e Xu (2002), no contexto de protecção global, os autores salientaram que $P_{A_{ij}} + b_k - B_{uv}$ é menor que $F_{ij} + b_k - B_{uv}$ usado em Kodialam e Lakshman (2000), onde k representa o pedido actual. Observamos que a instrução anterior ainda é válida no contexto de protecção local e então, a LBP adicional necessária no ramo (u, v) quando o ramo (i, j) é utilizado por um CA (correspondente ao pedido k) é quanto muito:

$$\delta_{ij}^{uv} = \max[\min(P_{A_{ij}} + b_k - B_{uv}, b_k) - \lambda_{uv}, 0] \quad (3.1)$$

A LB necessária em cada ramo de um CR, associada com o pedido k (considerando apenas partilha inter-pedido) é menor ou igual à LB, b_k , que seria necessária nos ramos do CA. Isto justifica a utilização de min na equação (3.1).

Também \max é necessário, porque $\min(P_{A_{ij}} + b - B_{uv}, b)$ pode ser menor do que λ_{uv} . Assim, θ_{ij}^{uv} é dado por:

$$\theta_{ij}^{uv} = \begin{cases} \delta_{ij}^{uv} & \text{se } R_{uv} \geq \delta_{ij}^{uv} \wedge (u, v) \neq (i, j) \\ \infty & \text{caso contrário} \end{cases} \quad (3.2)$$

Para ser aplicável numa rede real, o protocolo de sinalização terá que distribuir, R_{ij} , B_{ij} , e o valor de $P_{A_{ij}}$ calculados em cada nó i , em vez de F_{ij} . A sobrecarga de sinalização é, portanto, equivalente à abordagem em Kodialam e Lakshman (2001).

A proposta de *optimized Aggregate Information Scenario* (oAIS) de Raza *et al.* (2005), para o caso de protecção de um ramo, obtém H_{ij} (com $H_{ij} \leq F_{ij}$) como a quantidade de LBP usada simultaneamente em qualquer ramo, se o ramo (i, j) falha. De facto, $H_{ij} = P_{A_{ij}}$, embora H_{ij} tenha sido deduzido a partir de uma abordagem um pouco diferente e sem qualquer referência ao trabalho de Qiao e Xu (2002).

3.1.2 Um método para reserva mínima de LBP na protecção local

Em Kodialam e Lakshman (2002b); Liu *et al.* (2001); Qiao e Xu (2002) foram propostos métodos que, no contexto da protecção global, permitem fazer reserva mínima e libertação máxima de LB de protecção, numa rede com o modelo PI.

Um método para reserva mínima de LBP na protecção local, que seguidamente se descreve, requer tal como proposto em Qiao e Xu (2002), que para cada ramo (u, v) seja conhecido o conjunto completo de ramos (i, j) (dos caminhos activos) e P_B^{uv} , a quantidade de LB em (i, j) que é protegida pelo ramo (u, v) .

Dado um caminho activo e os seus caminhos de protecção, considere os pares formados por cada ramo de protecção (u, v) e o conjunto de ramos por ele protegido, P_{uv} . O método que permite fazer reserva mínima de LBP, para protecção global, proposto em Qiao e Xu (2002), pode ser aplicado a cada um desses pares, tomando o ramo (u, v) como um caminho de protecção global *virtual*, que protege o caminho activo *virtual* constituído pelos ramos em P_{uv} (que normalmente não coincidirá com nenhum caminho topológico). O mesmo procedimento pode ser utilizado para libertar caminhos de protecção (quando o caminho activo correspondente for libertado).

Um aspecto relevante desta aproximação é permitir que coexistam, numa mesma rede, esquemas de protecção local e global partilhando LB de protecção. Isto é possível porque ambos os métodos requerem o mesmo tipo de informação e podem partilhar as estruturas de dados que mantêm essa informação e por con-

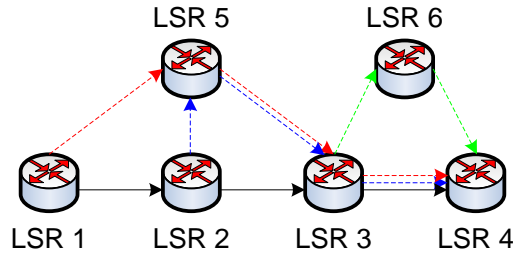


Figura 3.1: Rede para ilustrar a estrutura usada na reserva exacta.

seguinte as reservas. Para suportar esta partilha é também necessário que ambos os métodos usem o mesmo tipo de suporte de sinalização. Em geral reservas complexas como as pretendidas exigirão extensões ao RSVP para poderem ser implementadas, porem em Gomes *et al.* (2005) apresentou-se uma proposta para suportar estes tipos de sinalização sem necessidade de extensões ao RSVP.

A utilização do modelo (originalmente proposto para protecção global) num contexto de protecção local, implica em primeiro lugar obter o conjunto, L_{RP} , de todos os ramos pertencentes aos caminhos de protecção do caminho activo do pedido corrente. Em segundo lugar, dado L_{RP} , é preciso, para cada $(u, v) \in L_{RP}$, obter o conjunto, P_{uv} , dos ramos do caminho activo que são protegidos por (u, v) . Seguidamente o modelo de reserva exacta para protecção global, já conhecido, pode ser aplicado a cada caminho de protecção global *virtual*, o ramo (u, v) ($\forall (u, v) \in L_{RP}$), considerando que este protege o caminho activo *virtual* definido pelo conjunto P_{uv} . De notar que a LB adicional que o ramo (u, v) passa a proteger será igual à LB do pedido seja qual for o número de ramos que o caminho *activo virtual* possua.

Na Figura 3.1 é apresentada a rede usada para ilustrar o método proposto nesta secção. A Figura 3.2 descreve o caminho activo e os caminhos de protecção, através das respectivas sequências de ramos. Com base nesta figura é possível obter o conjunto de todos os ramos de protecção, L_{RP} . Para cada ramo, $(u, v) \in L_{RP}$, é obtido o conjunto dos ramos protegidos, P_{uv} , como se pode ver na Figura 3.3. Com base nesta informação pode então ser aplicado o método em Qiao e Xu (2002) a cada par de caminhos *virtuais* $(P_{uv}, (u, v))$, $\forall (u, v) \in L_{RP}$. Por exemplo, quando (u, v) é igual a $(LSR5, LSR3)$, $P_{uv} = \{(LSR1, LSR2), (LSR2, LSR3)\}$.

3.2 Um algoritmo para protecção local com reserva exacta

Em uma rede, onde é permitida a partilha LBP, o problema de encontrar um CA e o conjunto de CR que minimizam a LB total utilizada, é *NP-Hard* (Kodialam e

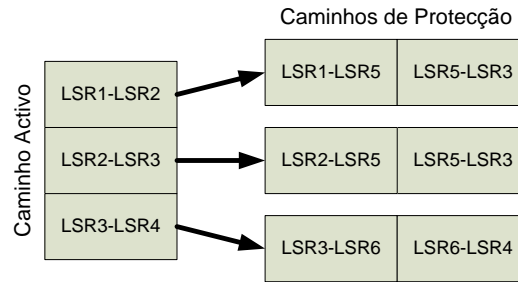


Figura 3.2: Estrutura activo-protecção.

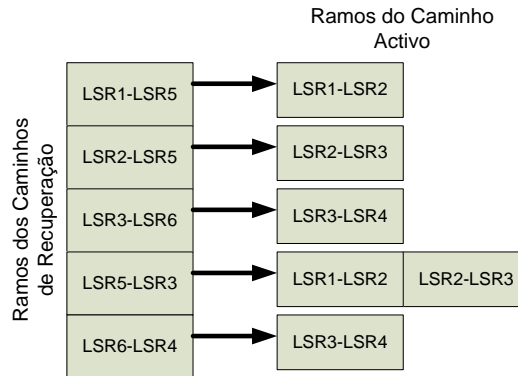


Figura 3.3: Estrutura protecção-activo.

Lakshman, 2003). No entanto, os algoritmos de encaminhamento *on-line* necessitam de gerar uma solução em poucos milissegundos. Portanto será apresentada a heurística descrita na Secção 3.2.2, que é uma melhoria de uma heurística extremamente eficiente (Kodialam e Lakshman, 2001), revista na Secção 3.2.1.

3.2.1 Visão geral do algoritmo em Kodialam e Lakshman (2001)

Seja $G = (V, E)$, o grafo da rede com todos os ramos invertidos. Daqui em diante, sempre que a notação apresentada anteriormente for usada na descrição do algoritmo é relativa a G (E em vez de E').

LOCAL_EDGE_DISJOINT(s, t) (Kodialam e Lakshman, 2001) é um algoritmo que calcula simultaneamente o caminho activo e os caminhos de protecção locais, de s para t ($s, t \in V$). O primeiro passo do algoritmo é a inversão de todos os ramos do grafo da rede – isto torna difícil a referência sem ambiguidades a um dado ramo. Nesta secção os ramos pertencerão sempre a G (o grafo da rede com todos os ramos invertidos), excepto quando indicado em contrário. Este algoritmo é uma versão modificada do algoritmo de *Dijkstra*, no qual o peso dos ramos da rede é calculado como o custo de os proteger, usando um caminho de protecção local (Kodialam e Lakshman, 2001, 2002a). Este algoritmo chama o

sub-algoritmo $\text{ALT_PATH_COST}(k, j)$ o qual calcula o custo mínimo de proteger localmente o ramo (k, j) . Isto é conseguido através de chamadas sucessivas ao sub-algoritmo $\text{SHORT_PRED_PATH}(k, u, j)$. Este sub-algoritmo é também uma versão modificada do algoritmo de *Dijkstra*, no qual o peso dos ramos é calculado como o custo de os usar no caminho de protecção local do ramo (k, j) . Em cada chamada do sub-algoritmo $\text{SHORT_PRED_PATH}(k, u, j)$, u é um nó no caminho activo candidato, de t para k , que começa por ser igual a k e percorre, nas chamadas sucessivas, todos os nós no caminho activo candidato até atingir t , inclusive. Desta forma $\text{ALT_PATH_COST}(k, j)$ determina o custo de todos os caminhos alternativos possíveis para protecção local do ramo (k, j) , com origem em u e destino em j e escolhe de entre eles o caminho de protecção local de custo mínimo.

$\text{LOCAL_EDGE_DISJOINT}(s, t)$ minimiza o custo de proteger o caminho activo, levando em conta a partilha de LB entre os LSP de protecção para o LSP activo calculado (partilha intra-pedido) e também entre os LSP de protecção de diferentes LSP activos (partilha inter-pedido).

A notação introduzida em Kodialam e Lakshman (2001) será utilizada com pequenas alterações. Um pedido é uma solicitação de um LSP protegido de s para t . O pedido r solicita b_r unidades de LB. Designa-se por k o último nó etiquetado permanentemente em $\text{LOCAL_EDGE_DISJOINT}(s, t)$ e por j um nó adjacente a k , o qual é o nó candidato corrente a integrar o caminho activo candidato que contém k . O custo mínimo de proteger o ramo (k, j) é calculado por $\text{ALT_PATH_COST}(k, j)$. Este valor é obtido chamando $\text{SHORT_PRED_PATH}(k, u, j)$ para cada nó u no caminho mais curto de t para k (incluindo k e t e com início em k). Isto porque o caminho de protecção do ramo (k, j) pode começar em qualquer nó no caminho mais curto de t até k (terminando sempre em j).

A lista de símbolos adicionais usados nos algoritmos é apresentada de seguida (assumindo que todos os nós pertencem a V e todos os ramos a E):

- λ_{mn}^u : quantidade de LB que será reservada para o conjunto dos caminhos que protegem os ramos no caminho activo candidato do nó t até u , para o pedido corrente, se o caminho activo candidato se torna o caminho activo efectivo, $(m, n) \in E'$.
- δ_{mn} : $\delta_{mn} = \min(P_{A_{ij}} + b - B_{mn}, b) - \lambda_{mn}^k$ na nova versão do algoritmo; $\delta_{mn} = F_{kj} + b - B_{mn} - \lambda_{mn}^u$ em Kodialam e Lakshman (2001).
- l_{mn} : custo do ramo (m, n) na determinação do caminho de protecção mais

curto.

- T (T'): conjunto dos nós etiquetados temporariamente no algoritmo de *Dijkstra* em **LOCAL_EDGE_DISJOINT**(s, t) (**SHORT_PRED_PATH**(k, u, j)).
- P (P'): conjunto dos nós etiquetados permanentemente no algoritmo de *Dijkstra* em **LOCAL_EDGE_DISJOINT**(s, t) (**SHORT_PRED_PATH**(k, u, j)).
- ϕ (γ): etiquetas dos nós no algoritmo de *Dijkstra* em **LOCAL_EDGE_DISJOINT**(s, t) (**SHORT_PRED_PATH**(k, u, j)). ϕ_i (γ_a) custo do caminho mais curto do nó t (u) para o nó i (a), se $i \in P$ ($a \in P'$). ϕ_i (γ_a) custo do caminho corrente do nó t (u) para o nó i (a), se $i \in T$ ($a \in T'$).
- Q (Q'): predecessores dos nós no algoritmo de *Dijkstra* em **LOCAL_EDGE_DISJOINT**(s, t) (**SHORT_PRED_PATH**(k, u, j)). $Q(i)$ ($Q(a)$) predecessor do nó i (a) no caminho mais curto com origem em t (u) no algoritmo de *Dijkstra*, se $i \in P$ ($a \in P'$). $Q(i)$ ($Q(a)$) predecessor do nó i (a) no caminho corrente se $i \in T$ ($a \in T'$).
- λ^u : vector formado por λ_{ij}^u . Vector de dimensão igual ao número de ramos na rede. Se u é um nó etiquetado permanentemente no algoritmo de *Dijkstra*, então λ^u atingiu o seu valor final.
- β (τ): vector auxiliar que armazena temporariamente λ^u em **SHORT_PRED_PATH**(k, u, j) (**ALT_PATH_COST**(k, j)).

Pretendendo tornar claras as referências às instruções dos vários algoritmos e sub-algoritmos, sempre que for utilizada a palavra “**linha**” (de um algoritmo) estamos a referir-nos a uma instrução da nova versão do algoritmo, e sempre que for utilizada a palavra “**passo**” (de um algoritmo) estamos a referir-nos a uma instrução da versão original.

3.2.2 Algoritmos original e melhorado

Na coluna do lado esquerdo é apresentado o algoritmo original (conforme apresentado em Kodialam e Lakshman (2001)), e na coluna do lado direito é apresentado o algoritmo com as correcções e alterações que propomos e também com as gralhas corrigidas.

A heurística proposta em Kodialam e Lakshman (2001) é difícil de seguir devido às interacções entre os três algoritmos, que não são explícitas na sua lista de parâmetros. Os algoritmos o acesso a variáveis que podem ser acedidas/alteradas

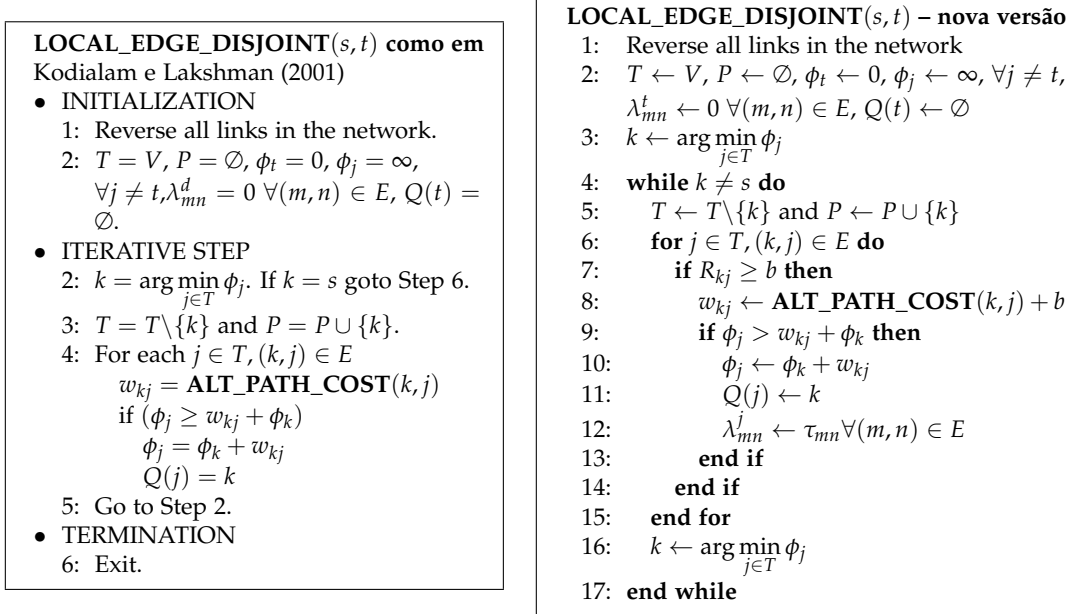


Figura 3.4: Algoritmo LOCAL_EDGE_DISJOINT(s, t) (original e modificado).

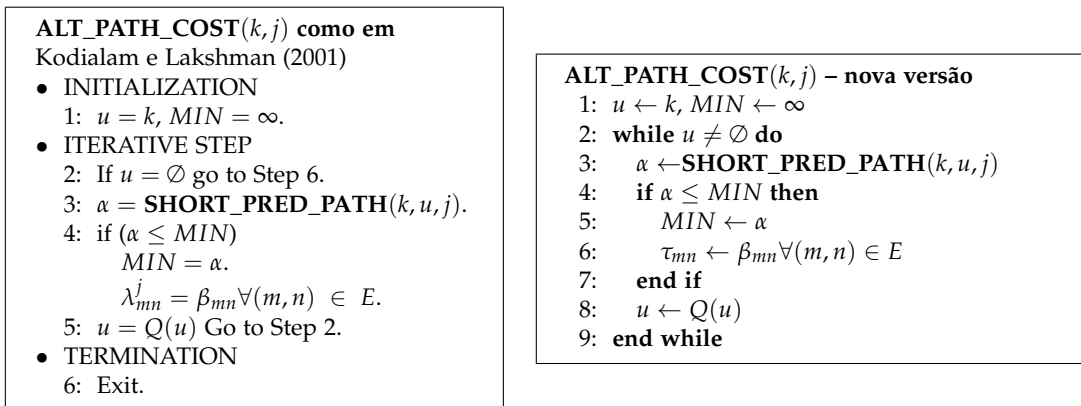


Figura 3.5: Algoritmo ALT_PATH_COST(k, j) (original e modificado).

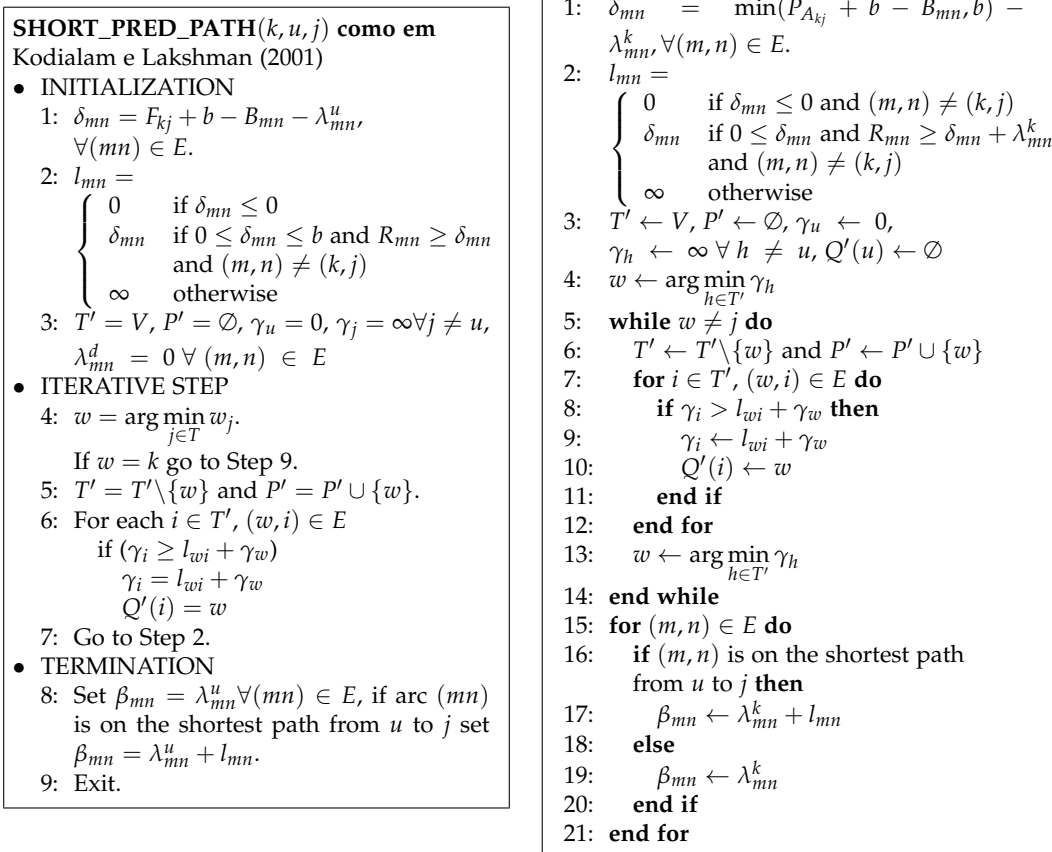


Figura 3.6: Algoritmo **SHORT_PRED_PATH(k, u, j)** (original e modificado).

em mais do que um algoritmo. Apresenta-se a seguir uma lista com as gralhas detectadas no algoritmo original em Kodialam e Lakshman (2001) (no algoritmo principal e num sub-algoritmo) e respectivas correcções.

No algoritmo **LOCAL_EDGE_DISJOINT**

- Substituir $\lambda_{m,n}^d$ (no passo 2) por $\lambda_{m,n}^t$ (como é visível na linha 2).
- A numeração dos passos deveria ser “1:, 2:, 3:, ..., 7:” e não “1:, 2:, 2:, 3:, ..., 6:”.

No sub-algoritmo **SHORT_PRED_PATH**

- Substituir $\gamma_j \leftarrow \infty, \forall j \neq u$ (no passo 3) por $\gamma_h \leftarrow \infty, \forall h \neq u$ (como é visível na linha 3), para evitar confusões com o parâmetro de entrada.
- Retirar $\lambda_{mn}^d = 0, \forall (m, n) \in E$ (no passo 3), pois está variável é inicializada em **LOCAL_EDGE_DISJOINT**.

- Acrescentar $Q'(u) = \emptyset$ (no passo 3), ver linha 3.
- Substituir $j \in T$ (no passo 4) por $h \in T'$ (como é visível na linha 4).
- Substituir w_j (no passo 4) por γ_h (como é visível na linha 4).
- Substituir “Step 9” (no passo 4) por “Step 8”.
- Substituir $w = k$ (no passo 4) por $w = j$ (ver linha 5).
- Substituir “if ($\gamma_i \geq l_{wi} + \gamma_w$)” (no passo 6) por “if ($\gamma_i > l_{wi} + \gamma_w$)” (como visível na linha 8), pois não há necessidade da alteração em caso de igualdade.
- Substituir “Step 2” (no passo 7) por “Step 4”.

Encontramos também algumas incorrecções no algoritmo original (no algoritmo principal e sub-algoritmos), as quais se não forem corrigidas conduzirão a resultados indesejados, como se mostrará na secção 3.2.3. As incorrecções detetadas são:

1. No sub-algoritmo **SHORT_PRED_PATH**(k, u, j), alterar a condição para atribuir um custo nulo a l_{mn} , de $\delta_{mn} \leq 0$ (passo 2) para $\delta_{mn} \leq 0$ and $(m, n) \neq (k, j)$ (linha 2). Só assim é garantido que ramos do caminho activo não sejam seleccionados para fazer parte dos caminhos de recuperação correspondentes.
2. No passo 4, do algoritmo **LOCAL_EDGE_DISJOINT**(s, t), a iteração j só deve ser efectuada se a LB residual do ramo (k, j) , R_{kj} , for superior ou igual a b (como visível na linha 7). Ou seja, um ramo só poderá fazer parte do caminho activo se tiver LB residual suficiente para o pedido em causa.
3. O valor de λ_{mn}^j não deve ser actualizado no passo 4 do sub-algoritmo **ALT_PATH_COST**(k, j) mas sim no passo 4 do algoritmo **LOCAL_EDGE_DISJOINT**(s, t). Para tal, no algoritmo **ALT_PATH_COST**(k, j) a instrução $\lambda_{mn}^j = \beta_{mn} \forall (m, n) \in E$ (no passo 4) foi substituída pela instrução $\tau_{mn} \leftarrow \beta_{mn} \forall (m, n) \in E$ (linha 6), e foi acrescentada a instrução $\lambda_{mn}^j \leftarrow \tau_{mn} \forall (m, n) \in E$ (linha 12) no algoritmo **LOCAL_EDGE_DISJOINT**(s, t). Assim evita-se a alteração de λ_{mn}^j quando a etiqueta, ϕ_j , e o predecessor, $Q(j)$, de j não sofrem alteração. Essa alteração poderia levar a que λ_{mn}^j tome valores incorrectos os quais poderão ser usados pelo algoritmo.

4. λ^k deve armazenar a LBP necessária para proteger todos os ramos no segmento final do candidato actual a CA (com cauda no nó k , o último nó etiquetado permanentemente em **LOCAL_EDGE_DISJOINT**(s, t)). Considere-se v pertencente ao caminho mais curto de t até k (excluindo k). Então λ^v armazena a quantidade de LBP necessária para proteger todos os ramos no segmento de t a v , do candidato actual a CA, que não inclui a largura de banda adicional necessária para proteger o segmento do CA de v até k . Portanto no sub-algoritmo **SHORT_PRED_PATH**(k, u, j), no cálculo dos β_{mn} (no passo 8), onde está λ^u deveria estar λ^k (ver linha 17 e 19). Caso contrário o valor final em **LOCAL_EDGE_DISJOINT**(s, t) pode levar a reservar um valor insuficiente de LB em algum dos ramos de protecção seleccionados.
5. No sub-algoritmo **SHORT_PRED_PATH**(k, u, j), alterar a condição para atribuir um custo a l_{mn} igual a δ_{mn} : de $0 \leq \delta_{mn} \leq b$ and $R_{mn} \geq \delta_{mn}$ and $(m, n) \neq (k, j)$, para $0 \leq \delta_{mn} \leq b$ and $R_{mn} \geq \delta_{mn} + \lambda_{mn}^k$ and $(m, n) \neq (k, j)$. Para o ramo (m, n) poder ser usado para proteger o ramo (k, j) a sua LB residual deve ser igual ou superior ao custo de o utilizar na protecção, δ_{mn} , mais a partilha intra-pedido, λ_{mn}^k , já conseguida na utilização desse ramo para protecção.

A ocorrência de problemas na utilização do algoritmo devido à falta da correcção 5 é menos frequente que devido à falta das outras correcções uma vez que as condições que conduzem a este problema são mais raras.

Uma versão mais eficiente de **LOCAL_EDGE_DISJOINT**(s, t) (considerando que todas as incorrecções e gralhas detectadas foram corrigidas) é obtida fazendo as seguintes alterações:

1. No sub-algoritmo **SHORT_PRED_PATH**(k, u, j) (no passo 1 – cálculo de δ_{mn}) onde está λ^u surge agora λ^k na nova versão (ver linha 1). Desta forma utiliza-se informação mais actualizada o que poderá conduzir a uma redução da LB de protecção, pois é utilizada toda a informação disponível para partilha intra-pedido.
2. No algoritmo **LOCAL_EDGE_DISJOINT**(s, t), alterar o valor a atribuir ao custo w_{kj} de **ALT_PATH_COST**(k, j) (passo 4) para **ALT_PATH_COST**(k, j) + b (linha 8). Desse modo esse custo toma não só em consideração o custo dos ramos do caminho de protecção mas também o custo do ramo do caminho activo, o que levará a CA com comprimentos menores, como será mostrado na Secção 3.2.4.

3. Esta alteração pode ser dividida em duas partes:

a) No sub-algoritmo **SHORT_PRED_PATH**(k, u, j) (no passo 1) onde está $\delta_{mn} = F_{kj} + b - B_{mn} - \lambda_{mn}^u, \forall (m, n) \in E$ passa a estar $\delta_{mn} = \min(F_{kj} + b - B_{mn}, b) - \lambda_{mn}^k, \forall (m, n) \in E$ na nova versão (ver linha 1). Dado um pedido de LB b , nenhum ramo dos caminhos de protecção do caminho activo correspondente necessitará de um aumento na reserva da LB de protecção superior a b . Assim a expressão presente na linha 1 da nova versão é uma estimativa mais precisa do valor mínimo da LB de protecção requerida no ramo (m, n) para proteger o ramo (k, j) .

Esta alteração implica que δ_{mn} será no máximo b e por conseguinte o teste $\delta_{mn} \leq b$, no passo 2, torna-se redundante pelo que foi removido na linha 2 da nova versão.

Na realidade, no algoritmo original, a atribuição do valor δ_{mn} ao custo l_{mn} , no sub-algoritmo **SHORT_PRED_PATH**(k, u, j) (no passo 2), apenas se $\delta_{mn} \leq b$ faz com que possam existir situações em que um pedido é rejeitado mesmo havendo LB disponível na rede.

b) Usar $P_{A_{kj}}$ em vez de F_{kj} resulta numa estimativa mais próxima da LB necessária nos CR, como será mostrado na Secção 3.2.4.

Em Kodialam e Lakshman (2001) o seguinte texto “the cost of using link (k, j) in the active path is the sum of bandwidth currently being routed and the cost of backing up link (k, j) ” parece indicar que a alteração 2 teria estado sob consideração por Kodialam e Lakshman. No entanto em **LOCAL_EDGE_DISJOINT**(s, t) (Kodialam e Lakshman, 2001, 2002a) foi escolhida uma abordagem diferente: a minimização da LB de protecção.

A alteração em 1 (combinada com a correcção 4) contribuirá para aumentar a partilha de LB de protecção, e a alteração em 2 poderá melhorar os tempos de propagação do tráfego nos caminhos activos e possivelmente também para reduzir a LB total necessária. A alteração 3a fará com que os custos estimados para os ramos de protecção sejam mais adequados.

O algoritmo proposto com reserva exacta de LBP necessita de recursos de memória adicionais nos nós da rede, quando comparado com a abordagem em Kodialam e Lakshman (2000). O cálculo de $P_{A_{ij}}$, introduzido na Secção 3.1 implica o armazenamento local (em cada nó $i \in V$) de P_{A}^{ij} , um vector de dimensão $|E|$, para cada ramo local (i, j) . Para implementar o mecanismo de reserva descrito na Subsecção 3.1.2, todos os nó $u \in V$, necessitarão de manter, para cada ramo local (u, v) , um vector P_B^{uv} , também de dimensão $|E|$. Sendo g o grau má-

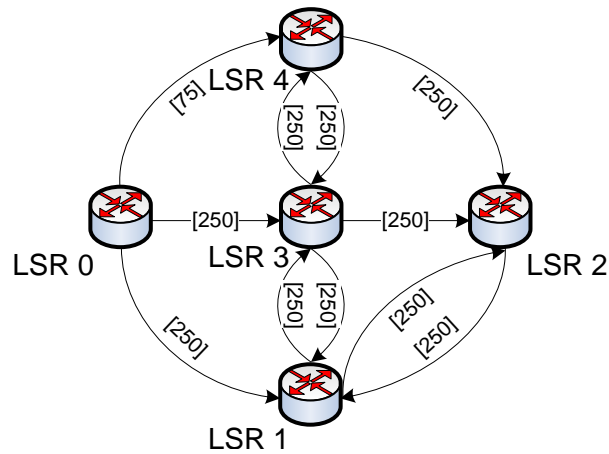


Figura 3.7: Rede exemplo usada para ilustrar as correcções e alterações propostas ao algoritmo.

| Ramo | 0-1 | 0-3 | 0-4 | 1-2 | 1-3 | 2-1 | 3-1 | 3-2 | 3-4 | 4-2 | 4-3 |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| F_{jk} | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| B_{nm} | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R_{nm} | 250 | 250 | 75 | 250 | 250 | 250 | 250 | 250 | 250 | 250 | 250 |

Tabela 3.1: Estado da rede antes do estabelecimento de qualquer pedido.

ximo de nós em G , a informação máxima necessária (localmente) em cada nó para implementar o algoritmo proposto é $\mathcal{O}(2g|E|)$.

3.2.3 Ilustração da pertinência das modificações propostas

Nesta subsecção recorrer-se-á a exemplos para ilustrar a pertinência das correcções e alterações propostas. Nesses exemplos utiliza-se a rede ilustrada na Figura 3.7 e a pedidos particulares nessa rede. A Tabela 3.1 corresponde ao estado inicial da rede.

Em cada um dos vários exemplos seguintes começa-se por estabelecer, em geral, um determinado número de pedidos. Os caminhos para estes pedidos serão determinados com a versão alterada do algoritmo. Para cada exemplo, os estados após esses pedidos iniciais serão os pontos de partida para ambos os algoritmos (original e alterado).

Lembra-se que o algoritmo começa por inverter os ramos da rede e a partir daí determina o CA e juntamente os CR do destino até à origem. No entanto, nesta subsecção, nas tabelas representam-se os ramos da rede original e quando são referidos ramos da rede são também apresentados de acordo com o sentido original (e não com o sentido inverso que o algoritmo usa). Perceberá os exemplos mais facilmente se tiver presente que o índice k corresponde ao último nó que

| Pedido | Origem-Destino | LB |
|--------|----------------|----|
| 1º | LSR0-LSR2 | 60 |
| 2º | LSR0-LSR2 | 50 |

Tabela 3.2: Pedidos considerados no exemplo para a Correção 1.

| Ramo | 0-1 | 0-3 | 0-4 | 1-2 | 1-3 | 2-1 | 3-1 | 3-2 | 3-4 | 4-2 | 4-3 |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| F_{jk} | 60 | 0 | 0 | 60 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| B_{nm} | 0 | 60 | 0 | 0 | 60 | 0 | 0 | 60 | 0 | 0 | 0 |
| R_{nm} | 190 | 190 | 75 | 190 | 190 | 250 | 250 | 190 | 250 | 250 | 250 |

Tabela 3.3: Estado da rede após o estabelecimento do primeiro pedido.

foi etiquetado definitivamente, o índice j corresponde ao nó que se está a tentar re-etiquetar e o índice u ao nó destino do caminho de protecção.

Correção 1

Deve ser notado que, no algoritmo **SHORT_PRED_PATH**(k, u, j) no teste do passo 2 (versão original), a condição apresentada para atribuir um custo nulo a l_{mn} arrisca-se a escolher para os CR troços pertencentes ao CA, se nestes o valor de LB para protecção ($B_{mn} + \lambda_{mn}^k$) ultrapassar a LB necessária para CA (F_{kj}) somada à LB para o pedido em causa (b).

Os pedidos para o exemplo desta correção foram os apresentados na Tabela 3.2.

A Tabela 3.3 corresponde ao estado da rede após o estabelecimento do primeiro pedido. Na determinação dos caminhos para o segundo pedido (Tabela 3.2), quando o algoritmo **SHORT_PRED_PATH**(k, u, j) procura um CR para o ramo LSR3-LSR2 ($j = 3$ e $k = u = 2$) começa por determinar os valores dos vectores δ e l . Os valores que obtém são apresentados na Tabela 3.4 (determinam-se facilmente a partir da Tabela 3.3), que mostra que foi atribuído um custo zero para a utilização do ramo LSR3-LSR2 como protecção do ramo LSR3-LSR2. Como é óbvio, com esses valores para os custos o algoritmo de *Dijkstra* (código da linha 3 à linha 14) obtém para CR o ramo LSR3-LSR2. O que finalmente resultará na escolha dos CR LSR0-LSR3-LSR2 e LSR3-LSR2 e do CA LSR0-LSR3-LSR2 (não disjuntos).

Este problema é evitado, na nova versão, através da correção da condição que atribuir um custo nulo a l_{mn} (linha 2). Deste modo, um ramo que pertença ao CA será sempre impedido de ser utilizado em CR (através da atribuição de um custo

| Ramo | 0-1 | 0-3 | 0-4 | 1-2 | 1-3 | 2-1 | 3-1 | 3-2 | 3-4 | 4-2 | 4-3 |
|---------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| δ_{nm} | 50 | -10 | 50 | 50 | -10 | 50 | 50 | -10 | 50 | 50 | 50 |
| l_{nm} | 50 | 0 | 50 | 50 | 0 | 50 | 50 | 0 | 50 | 50 | 50 |

Tabela 3.4: Valores dos vectores δ e l numa invocação particular de **SHORT_PRED_PATH**(k, u, j) na versão original do algoritmo.

infinito).

Correcção 2

A necessidade desta correcção parece-nos demasiado óbvia para necessitar de um exemplo.

Correcção 3

Como foi referido o vector λ^k guarda a quantidade de LB necessária para os CR de todos os ramos no caminho (na árvore de caminhos mais curtos) do nó k até ao nó destino, para o pedido actual.

No exemplo seguinte adicionou-se o ramo LSR3-LSR0 (com capacidade 250) à rede apresentada na Figura 3.7. Foi tentado estabelecer, com o algoritmo original (com as Correcções 1, 2 e 4), o pedido entre o LSR0 (origem) e o LSR2 (destino) com 5 unidades de LB. Para o pedido em causa o algoritmo **LOCAL_EDGE_DISJOINT**(s, t) começa por etiquetar definitivamente o LSR2 ($k = 2$) e temporariamente todos os restantes com etiquetas ∞ , após o que vai re-etiquetar temporariamente todos os nós adjacentes a LSR2 (LSR1, LSR3 e LSR4), quando $j = 1, 3$ e 4 (passo 4 desse algoritmo), e calcular os λ^j , indirectamente pela chamada do algoritmo **ALT_PATH_COST**(k, j) (no passo 4 deste), correspondentes a esses nós (ver Tabela 3.5). As etiquetas temporárias desses nós passam todos a ter o valor 10 ($\phi_j = 10$). O algoritmo escolhe etiquetar definitivamente o LSR1 ($k = 1$) após o que na análise das suas adjacências (LSR0 e LSR3) o valor de λ^3 é alterado para reflectir o custo da eventual inclusão do ramo LSR1-LSR3 na árvore de caminhos mais curtos. No entanto, como o custo mínimo dessa inclusão - protecção do ramo LSR3-LSR1 com o ramo LSR3-LSR2 - somado à etiqueta do LSR1 ($w_{13} + \phi_1 = 5 + 10$ - passo 4 do algoritmo **LOCAL_EDGE_DISJOINT**(s, t)) é superior à etiqueta actual do LSR3 ($\phi_3 = 10$) esta não é alterada bem como o não é o predecessor do LSR3, porém nos cálculos subsequentes λ^3 é utilizado após ter sido incorrectamente (indevidamente) alterado (ver Tabela 3.6).

Este problema foi resolvido, no algoritmo corrigido, utilizando uma variável

| Ramo | 0-1 | 0-3 | 0-4 | 1-2 | 1-3 | 2-1 | 3-0 | 3-1 | 3-2 | 3-4 | 4-2 | 4-3 |
|-------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| λ^1 | 0 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 5 | 0 | 0 | 0 |
| λ^3 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 0 |
| λ^4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 0 | 0 | 5 |

Tabela 3.5: Valores em vectores λ após a execução da primeira iteração do código identificado como "ITERATIVE STEP" da versão original do algoritmo **LOCAL_EDGE_DISJOINT**(s, t).

| Ramo | 0-1 | 0-3 | 0-4 | 1-2 | 1-3 | 2-1 | 3-0 | 3-1 | 3-2 | 3-4 | 4-2 | 4-3 |
|-------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| λ^3 | 0 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 10 | 0 | 0 | 0 |

Tabela 3.6: Valores no vector λ^3 após a execução da segunda iteração do código identificado como "ITERATIVE STEP" da versão original do algoritmo **LOCAL_EDGE_DISJOINT**(s, t).

| Ramo | 0-1 | 0-3 | 0-4 | 1-2 | 1-3 | 2-1 | 3-0 | 3-1 | 3-2 | 3-4 | 4-2 | 4-3 |
|-------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| λ^4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 0 | 0 | 5 |

Tabela 3.7: Valores no vector λ^4 quando LSR4 é etiquetado permanentemente.

auxiliar para guardar temporariamente o vector λ^j no algoritmo **ALT_PATH_COST**(k, j) (linha 6), e actualizando o vector λ^j apenas no algoritmo **LOCAL_EDGE_DISJOINT**(s, t) (linha 12) quando é alterada a etiqueta e o predecessor do nó j .

Correcção 4

No exemplo seguinte utiliza-se a mesma rede (usada na Correcção 3) e tenta-se estabelecer, com o algoritmo original (com todas as correcções anteriores), o mesmo pedido (entre o LSR0 e o LSR2 com 5 unidades de LB) usado no exemplo da correcção anterior.

O algoritmo etiqueta definitivamente o LSR4 a partir do LSR2, o que significa que o ramo LSR4-LSR2 pertence à árvore de caminhos mais curtos. Tendo sido escolhido LSR4-LSR3-LSR2 para caminho de protecção desse ramo. Por isso o vector λ^4 possui nessa altura os valores indicados da Tabela 3.7.

Posteriormente, LSR0 é etiquetado permanentemente pelo algoritmo a partir de LSR4 ($k = 4$). O algoritmo escolheu para caminho de protecção do ramo LSR0-LSR4 o caminho LSR0-LSR1-LSR2 ($u = 2$). O vector λ^0 possui, no final da execução do algoritmo, os valores indicados da Tabela 3.8, pois λ^0 foi calculado somando os custos dos dois ramos do caminho de protecção nas posições respectivas de λ^2 (este vector possui zeros em todas as posições uma vez que

| Ramo | 0-1 | 0-3 | 0-4 | 1-2 | 1-3 | 2-1 | 3-0 | 3-1 | 3-2 | 3-4 | 4-2 | 4-3 |
|-------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| λ^0 | 5 | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Tabela 3.8: Valores no vector λ^0 quando LSR0 é etiquetado permanentemente.

| Ramo | 0-1 | 0-3 | 0-4 | 1-2 | 1-3 | 2-1 | 3-0 | 3-1 | 3-2 | 3-4 | 4-2 | 4-3 |
|-------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| λ^0 | 5 | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 5 | 0 | 0 | 5 |

Tabela 3.9: Valores que o vector λ^0 deveria ter quando LSR0 é etiquetado permanentemente.

| Ramo | 0-1 | 0-3 | 0-4 | 1-2 | 1-3 | 2-1 | 3-1 | 3-2 | 3-4 | 4-2 | 4-3 |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| F_{jk} | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| B_{nm} | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R_{nm} | 0 | 0 | 0 | 80 | 200 | 200 | 100 | 75 | 0 | 0 | 0 |

Tabela 3.10: Estado da rede antes do estabelecimento de qualquer pedido - Correção 5.

corresponde ao nó destino).

No entanto, no final da execução do algoritmo, o vector λ^0 deveria possuir os valores indicados da Tabela 3.9, uma vez que o vector λ^0 deve guardar a quantidade de LB necessária para os CR de todos os ramos do CA, para o pedido actual.

Este problema foi resolvido, no algoritmo corrigido, utilizando λ^k em vez de λ^u no passo 8 do sub-algoritmo **SHORT_PRED_PATH**(k, u, j). Para o exemplo em causa, esta correção implicava que o cálculo de λ^0 tinha em consideração o vector λ^4 em vez do vector λ^2 .

Correção 5

Para ilustrar a necessidade da Correção 5 recorreremos mais uma vez à rede com a topologia apresentada na Figura 3.7. As capacidades dos ramos para este exemplo são as apresentadas na Tabela 3.10.

Após o estabelecimento dos dois primeiros pedidos com as características apre-

| Pedido | Origem-Destino | LB | CA | CR |
|----------------|----------------|----|-----------|----------------|
| 1 ^o | LSR3-LSR1 | 50 | LSR3-LSR1 | LSR3-LSR2-LSR1 |
| 2 ^o | LSR1-LSR2 | 30 | LSR1-LSR2 | LSR1-LSR3-LSR2 |
| 3 ^o | LSR3-LSR2 | 30 | - | - |

Tabela 3.11: Pedidos considerados no exemplo - Correção 5.

| Ramo | 0-1 | 0-3 | 0-4 | 1-2 | 1-3 | 2-1 | 3-1 | 3-2 | 3-4 | 4-2 | 4-3 |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| F_{jk} | 0 | 0 | 0 | 30 | 0 | 0 | 50 | 0 | 0 | 0 | 0 |
| B_{nm} | 0 | 0 | 0 | 0 | 30 | 50 | 0 | 50 | 0 | 0 | 0 |
| R_{nm} | 0 | 0 | 0 | 50 | 170 | 150 | 50 | 25 | 0 | 0 | 0 |

Tabela 3.12: Estado da rede após o estabelecimento dos dois primeiros pedidos - Correção 5.

| Ramo | 0-1 | 0-3 | 0-4 | 1-2 | 1-3 | 2-1 | 3-0 | 3-1 | 3-2 | 3-4 | 4-2 | 4-3 |
|-------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| λ^1 | 0 | 0 | 0 | 0 | 30 | 0 | 0 | 0 | 10 | 0 | 0 | 0 |

Tabela 3.13: Valores no vector λ^1 após a execução da primeira iteração do código identificado como “ITERATIVE STEP” da versão original do algoritmo **LOCAL_EDGE_DISJOINT**(s, t).

sentadas na Tabela 3.11, o estado da rede passou a ser o apresentado na Tabela 3.12.

Quando surge o terceiro pedido, de LSR3 para LSR2, uma vez que o ramo LSR3-LSR2 não possui LB residual suficiente para o CA o algoritmo original (com as correções 1, 2, 3 e 4 e também com a alteração 1) vai escolher para CA o caminho LSR3-LSR1-LSR2 que possui LB residual suficiente para o pedido. Para proteger o ramo LSR1-LSR2 escolhe o CR LSR1-LSR3-LSR2 e para proteger o ramo LSR3-LSR1 escolhe o CR LSR3-LSR2. Porém esta última escolha não pode logicamente ocorrer, por falta de LB. Os cálculos que estiveram na base dessas escolhas são os seguintes:

- O algoritmo começa por determinar adequadamente o CR LSR1-LSR3-LSR2, para proteger o ramo LSR1-LSR2. Quando este caminho de protecção é determinado o custo de usar o ramo LSR1-LSR3 na protecção é 30 e o custo de usar o ramo LSR3-LSR2 na protecção é 10. Os custos foram determinado no passo 2 do sub-algoritmo **SHORT_PRED_PATH**(1,2,2) (o primeiro custo é obtido da expressão $F_{12} + b - B_{13} - \lambda_{13}^2 = 30 + 30 - 30 - 0$ e o segundo é obtido da expressão $F_{12} + b - B_{32} - \lambda_{32}^2 = 30 + 30 - 50 - 0$). Estes custos serão a informação que fica armazenada no vector λ^1 (conforme ilustrado na Tabela 3.13) após a primeira iteração do código identificado como “ITERATIVE STEP” da versão original do algoritmo **LOCAL_EDGE_DISJOINT**(s, t). Este vector indica que os próximos CR a ser determinados para o CA que contém o ramo LSR1-LSR2 podem utilizar partilha intra-pedido nos ramos LSR1-LSR3 e LSR3-LSR2 de 30 e 10 unidades, respectivamente.
- Na segunda iteração do código identificado como “ITERATIVE STEP” da versão original (com as modificações referidas anteriormente) do algoritmo

LOCAL_EDGE_DISJOINT(s, t) quando determina a protecção para o ramo LSR3-LSR1 o sub-algoritmo **SHORT_PRED_PATH**(3,2,1) obtém 20 para o custo de usar o ramo LSR3-LSR2 na protecção (através da expressão $F_{31} + b - B_{32} - \lambda_{32}^1 = 50 + 30 - 50 - 10$ e porque a LB residual no ramo LSR3-LSR2, R_{32} , é igual a 25).

Ora o custo deveria ser infinito uma vez que a capacidade do ramo LSR3-LSR2 é 75 unidades das quais 50 já estão a ser usadas para protecção do ramo LSR3-LSR1 (para o primeiro pedido), restando apenas 25 para proteger outro pedido no ramo LSR3-LSR1 e o terceiro pedido necessita de 30.

É necessário existir a conjugação de várias condições para situações destas ocorrerem:

- Um ramo ser usado na protecção de dois ou mais ramos de um CA.

O ramo LSR3-LSR2 estava a ser usado para proteger o ramo LSR1-LSR2 e o LSR3-LSR1.

- Inicialmente esse ramo do CR poder ser usado para proteger um ramo do CA mas com custo não nulo (posteriormente indicado como partilha intra-pedido possível).

O custo do ramo LSR3-LSR2 proteger o ramo LSR1-LSR2 foi 10.

- O uso desse ramo para proteger um dos ramos finais do CA ser impossível por não haver LB residual suficiente. No entanto a LB necessária para essa protecção menos a partilha intra-pedido deverá ser inferior ou igual à LB residual.

A utilização do ramo LSR3-LSR2 para proteger o ramo LSR3-LSR1 é impossível uma vez que eram necessárias 30 unidades de LB e apenas estão disponíveis 25. No entanto a LB necessária para protecção (30) menos a partilha intra-pedido (10) é inferior à LB residual (25).

Para resolver este problema foi alterada a condição que atribui um custo não nulo e diferente de infinito ao ramo (m, n) (linha 2) de $\dots R_{mn} \geq \delta_{mn} \dots$ para $\dots R_{mn} \geq \delta_{mn} + \lambda_{mn}^k \dots$, assim o custo será infinito se a LB residual não for suficiente. Colocado de uma forma simples: Quando se determina o custo de usar um ramo na protecção o seu valor deverá ser infinito se a LB residual for inferior à soma de δ_{mn} mais a partilha intra-pedido, isto porque de δ_{mn} tinha sido subtraída a partilha intra-pedido.

| Pedido | Origem-Destino | LB | CA | CR | Algoritmo |
|----------------|----------------|----|----------------|-----------------------------------|-----------|
| 1 ^o | LSR3-LSR4 | 5 | LSR3-LSR4 | LSR3-LSR0-LSR4 | - |
| 2 ^o | LSR0-LSR2 | 60 | LSR0-LSR1-LSR2 | LSR0-LSR3-LSR2, LSR1-LSR3-LSR2 | Alterado |
| 2 ^o | LSR0-LSR2 | 60 | LSR0-LSR3-LSR2 | LSR0-LSR4-LSR2, LSR3-LSR1-LSR2 | Original |

Tabela 3.14: Pedidos considerados no exemplo para a Alteração 1.

| Ramo | 0-1 | 0-3 | 0-4 | 1-2 | 1-3 | 2-1 | 3-0 | 3-1 | 3-2 | 3-4 | 4-2 | 4-3 |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| F_{jk} | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 0 | 0 |
| B_{nm} | 0 | 0 | 5 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 0 |
| R_{nm} | 250 | 250 | 70 | 250 | 250 | 250 | 245 | 250 | 250 | 245 | 250 | 250 |

Tabela 3.15: Estado da rede após o estabelecimento do primeiro pedido.

Deve ser notado que o exemplo apresentado ilustra a necessidade da Correção no algoritmo original com a alteração 1, por uma questão de simplicidade. No entanto o mesmo problema pode ocorrer no algoritmo original (sem a alteração 1) pois este também entra para o cálculo do custo em consideração com a partilha intra-pedido, embora de uma forma menos eficiente.

Alteração 1

Para este exemplo acrescentámos também o ramo LSR3-LSR0 (com capacidade 250) à rede apresentada na Figura 3.7. Começámos por estabelecer um pedido entre o LSR3 (origem) e o LSR4 (destino) com 5 unidades de LB. De seguida foi estabelecido, com o algoritmo alterado e com o algoritmo original (este com todas as Correções), o pedido entre o LSR0 (origem) e o LSR2 (destino) com 60 unidades de LB, os resultados respectivos são apresentados na Tabela 3.14. Regista-se que no exemplo apresentado a alteração 3a não foi considerada no algoritmo modificado, no entanto se ela tivesse sido considerada o resultado final, em termos de CA e CR, seria exactamente o mesmo.

A Tabela 3.15 corresponde ao estado da rede após o estabelecimento do primeiro pedido.

As Figuras 3.8 e 3.9 ilustram as quatro iterações do código identificado como "ITERATIVE STEP" da versão original do algoritmo `LOCAL_EDGE_DISJOINT(s, t)` e as quatro iterações do ciclo "WHILE" (linha 4 a linha 17) da versão alterada do algoritmo `LOCAL_EDGE_DISJOINT(s, t)`, respectivamente (para o segundo pedido). Nessas figuras, em cada uma das iterações, os ramos candida-

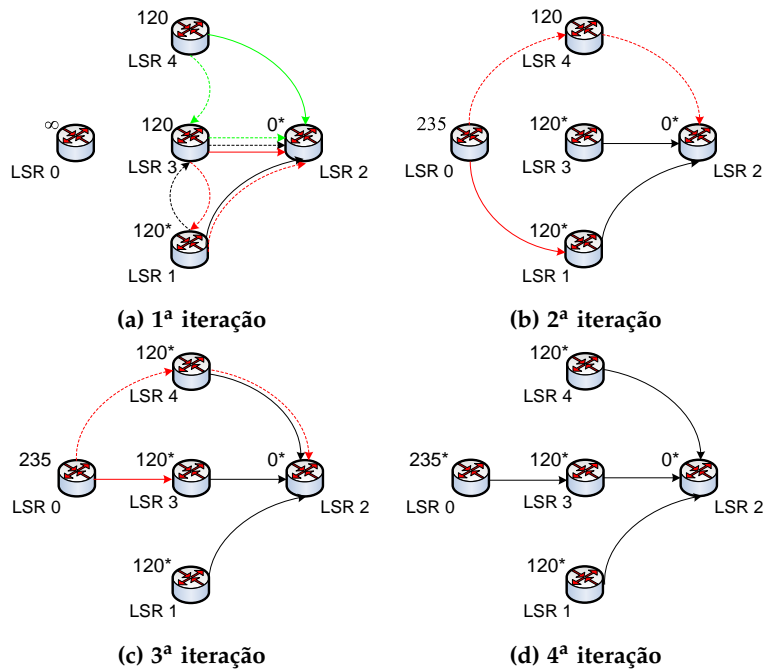


Figura 3.8: Iterações principais do algoritmo $\text{LOCAL_EDGE_DISJOINT}(s, t)$ - versão original.

tos a entrar na árvore de caminhos mais curtos são representados com uma linha contínua e os caminhos de protecção escolhidos para esses ramos são representados por linhas descontínuas. Sendo que, o caminho de protecção e o ramo protegido por este possuem a mesma cor, e a cor preta é usada para o próximo ramo a entrar na árvore de caminhos mais curtos. Para simplificar as figuras, em cada iteração apenas se mantém das iterações anteriores a representação dos ramos que fazem parte da árvore.

Ambos os algoritmos começam por etiquetar definitivamente o nó 2 com etiqueta zero (e todos os outros com etiquetas ∞) após o que re-etiquetam temporariamente os nós adjacentes a esse (nós 1, 3 e 4), como é visível nas Figuras 3.8(a) e 3.9(a). Com o algoritmo alterado as etiquetas temporárias obtidas são 180 para os três nós, enquanto que com o algoritmo original são 120 (cada um destes valores é igual ao correspondente anterior menos a LB do pedido em causa). Ambos os algoritmos etiquetam definitivamente o nó 1 após o que re-etiquetam temporariamente o nó 0 (o nó 3 embora adjacente mantém o valor), no algoritmo original passa a ter o valor 235 (etiqueta de LSR1 mais o custo da protecção do ramo LSR0-LSR1) enquanto que no alterado é re-etiquetado para 300, 180 (etiqueta de LSR1) + 60 (custo da protecção do ramo LSR0-LSR1) + 60 (LB do pedido). Tanto o algoritmo original como o alterado etiquetam definitivamente o LSR3, como é ilustrado nas Figuras 3.8(b) e 3.9(b), respectivamente.

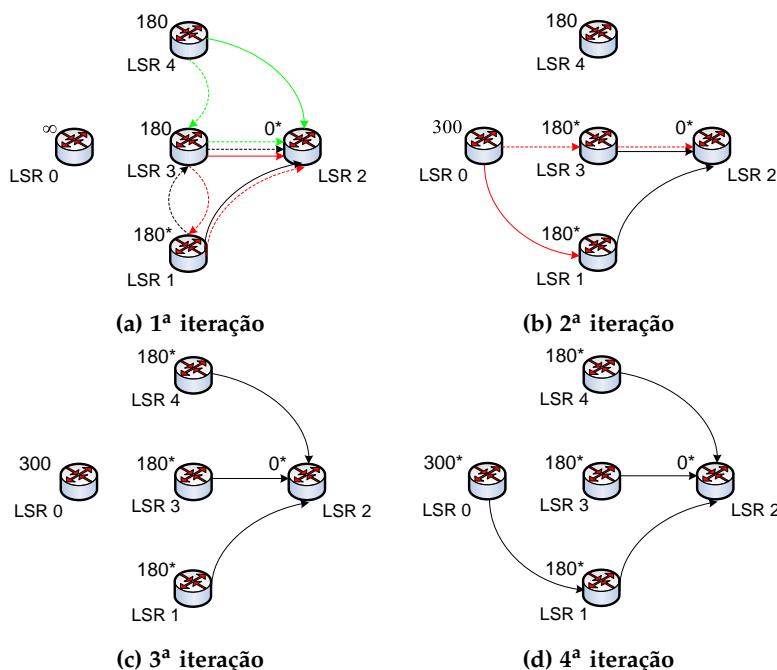


Figura 3.9: Iterações principais do algoritmo LOCAL_EDGE_DISJOINT(s, t) - versão alterada.

O algoritmo original na terceira iteração, ilustrada na Figura 3.8(c), re-etiqueta o LSR0 com 235, 120 (etiqueta de LSR3) + 115 (custo da protecção do ramo LSR0-LSR3). Note que o LSR0 foi re-etiquetado, desnecessariamente, com o mesmo valor que já possuía, sendo isso resultado da condição, $(\phi_j \geq w_{kj} + \phi_k)$, no passo 4 do algoritmo LOCAL_EDGE_DISJOINT(s, t). O LSR4 mantém a etiqueta tornando-se essa definitiva. Na mesma iteração o algoritmo alterado (ver Figura 3.9(c)) não efectua nenhuma alteração no valor das etiquetas mas torna definitiva a etiqueta do LSR4. Na última iteração tanto o algoritmo original (ver Figura 3.8(d)) como o algoritmo alterado (ver Figura 3.9(d)) apenas tornam definitiva a etiqueta de LSR0. Estes resultados confirmam as escolhas apresentadas na Tabela 3.14 para os CA e CR.

Os estados da rede, após o estabelecimento do segundo pedido, são representados nas Tabelas 3.17 e 3.18, respectivamente para a versão original e alterada do algoritmo. Neste exemplo, embora ambos os algoritmos tenham partido do mesmo estado da rede, é visível que, para satisfazer o segundo pedido, a LB de protecção é superior na versão original. Na solução do algoritmo alterado o ramo LSR3-LSR2 é utilizado tanto para protecção do ramo LSR1-LSR2 como para protecção do ramo LSR0-LSR1. Este tipo de partilha não ocorreu no algoritmo original porque na segunda iteração quando re-etiquetou temporariamente o LSR0, o custo da protecção do ramo LSR0-LSR1 115 (55 do ramo LSR0-LSR4 (recorrendo

| Ramo | 0-1 | 0-3 | 0-4 | 1-2 | 1-3 | 2-1 | 3-0 | 3-1 | 3-2 | 3-4 | 4-2 | 4-3 |
|-------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| λ^1 | 0 | 0 | 0 | 0 | 60 | 0 | 0 | 0 | 60 | 0 | 0 | 0 |
| λ^2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Tabela 3.16: Valores nos vectores λ^1 e λ^2 na 2ª iteração do algoritmo LOCAL_EDGE_DISJOINT(s, t).

| Ramo | 0-1 | 0-3 | 0-4 | 1-2 | 1-3 | 2-1 | 3-0 | 3-1 | 3-2 | 3-4 | 4-2 | 4-3 |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| F_{jk} | 0 | 60 | 0 | 0 | 0 | 0 | 0 | 0 | 60 | 5 | 0 | 0 |
| B_{nm} | 0 | 0 | 60 | 60 | 0 | 0 | 5 | 60 | 0 | 0 | 60 | 0 |
| R_{nm} | 250 | 190 | 15 | 190 | 250 | 250 | 245 | 190 | 190 | 245 | 190 | 250 |

Tabela 3.17: Estado da rede após o estabelecimento do segundo pedido - versão original.

| Ramo | 0-1 | 0-3 | 0-4 | 1-2 | 1-3 | 2-1 | 3-0 | 3-1 | 3-2 | 3-4 | 4-2 | 4-3 |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| F_{jk} | 60 | 0 | 0 | 60 | 0 | 0 | 0 | 0 | 0 | 5 | 0 | 0 |
| B_{nm} | 0 | 60 | 5 | 0 | 60 | 0 | 5 | 0 | 60 | 0 | 0 | 0 |
| R_{nm} | 190 | 190 | 70 | 190 | 190 | 250 | 245 | 250 | 190 | 245 | 250 | 250 |

Tabela 3.18: Estado da rede após o estabelecimento do segundo pedido - versão alterada.

a partilha inter-pedido) +60 do ramo LSR4-LSR2) não teve em consideração a informação mais recente relativa à partilha intra-pedido, utiliza o λ^2 em vez do λ^1 que é mais actualizado (ver Tabela 3.16), no cálculo de δ_{mn} a partir do qual se obtém os custos, l_{mn} , dos ramos no algoritmo SHORT_PRED_PATH(k, u, j). Como na versão alterada é utilizada a informação do λ^1 este consegue partilha intra-pedido no ramo LSR3-LSR2 (este ramo já estava a ser usado para protecção do ramo LSR1-LSR2).

O algoritmo mesmo com as alterações mantém-se uma heurística “greedy” uma vez que as escolhas iniciais podem impedir conseguir uma maior partilha no futuro. Como exemplo podemos utilizar o caso anterior. Vamos considerar que no segundo pedido, na primeira iteração, o algoritmo escolhia proteger o ramo LSR3-LSR2 com o caminho LSR3-LSR4-LSR2 (neste caso este caminho tem o mesmo custo que o que foi escolhido pelo algoritmo, embora isto não seja sempre condição necessária, ou seja pode mesmo haver situações em que a escolha inicial de um caminho de custo superior poderá conduzir a um consumo de LB final inferior). Se isso tivesse acontecido então na segunda iteração o LSR0 seria re-etiquetado a partir do LSR3 e não do LSR1 como aconteceu, pois neste caso estaria a usar simultaneamente partilha intra-pedido e inter-pedido. O estado da rede após o estabelecimento do segundo pedido seria o apresentado na Tabela

| Ramo | 0-1 | 0-3 | 0-4 | 1-2 | 1-3 | 2-1 | 3-0 | 3-1 | 3-2 | 3-4 | 4-2 | 4-3 |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| F_{jk} | 0 | 60 | 0 | 0 | 0 | 0 | 0 | 0 | 60 | 5 | 0 | 0 |
| B_{nm} | 0 | 0 | 60 | 0 | 0 | 0 | 5 | 0 | 0 | 60 | 60 | 0 |
| R_{nm} | 250 | 190 | 15 | 250 | 250 | 250 | 250 | 245 | 190 | 185 | 190 | 250 |

Tabela 3.19: Estado da rede após o estabelecimento do segundo pedido - com escolha de caminho de protecção alternativo.

| Pedido | Origem-Destino | LB | CA | CR |
|----------------|----------------|----|-----------|----------------|
| 1 ^o | LSR3-LSR2 | 5 | LSR3-LSR2 | LSR3-LSR1-LSR2 |
| 2 ^o | LSR0-LSR4 | 60 | LSR0-LSR4 | LSR0-LSR3-LSR4 |
| 3 ^o | LSR0-LSR3 | 80 | LSR0-LSR3 | LSR0-LSR1-LSR3 |
| 4 ^o | LSR1-LSR2 | 80 | LSR1-LSR2 | LSR1-LSR3-LSR2 |

Tabela 3.20: Pedidos considerados no exemplo para a Alteração 2.

| Ramo | 0-1 | 0-3 | 0-4 | 1-2 | 1-3 | 2-1 | 3-1 | 3-2 | 3-4 | 4-2 | 4-3 |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| F_{jk} | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 0 | 0 | 0 |
| B_{nm} | 0 | 0 | 0 | 5 | 0 | 0 | 5 | 0 | 0 | 0 | 0 |
| R_{nm} | 250 | 250 | 75 | 245 | 250 | 250 | 245 | 245 | 250 | 250 | 250 |

Tabela 3.21: Estado da rede após o estabelecimento do primeiro pedido.

3.19, o que mostra que a LB total necessária para o segundo pedido foi de 295 unidades (ver também Tabela 3.15) enquanto que para o mesmo pedido tinha anteriormente sido 300 unidades (ver Tabela 3.15 e Tabela 3.18). Isto aconteceu porque a escolha da protecção do ramo LSR3-LSR2 foi efectuada sem ter em consideração as futuras possibilidades de partilha inter-pedido.

Alteração 2

A sequência de pedidos iniciais bem como os caminhos obtidos, para esses pedidos, são apresentados na Tabela 3.20. De novo regista-se que no exemplo apresentado a alteração 3a não foi considerada no algoritmo modificado, no entanto se ela tivesse sido considerada o resultado final, em termos de CA e CR, seria exactamente o mesmo.

As Tabelas 3.21, 3.22, 3.23 e 3.24 correspondem ao estado após o estabelecimento do primeiro pedido, do segundo pedido, do terceiro pedido e do quarto pedido, respectivamente.

Vamos considerar que o quinto pedido é entre o LSR0 e o LSR2 com 80 de LB e vamos mostrar, para esse pedido, quais são as escolhas efectuadas para o CA

| Ramo | 0-1 | 0-3 | 0-4 | 1-2 | 1-3 | 2-1 | 3-1 | 3-2 | 3-4 | 4-2 | 4-3 |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| F_{jk} | 0 | 0 | 60 | 0 | 0 | 0 | 0 | 5 | 0 | 0 | 0 |
| B_{nm} | 0 | 60 | 0 | 5 | 0 | 0 | 5 | 0 | 60 | 0 | 0 |
| R_{nm} | 250 | 190 | 15 | 245 | 250 | 250 | 245 | 245 | 190 | 250 | 250 |

Tabela 3.22: Estado da rede após o estabelecimento do segundo pedido.

| Ramo | 0-1 | 0-3 | 0-4 | 1-2 | 1-3 | 2-1 | 3-1 | 3-2 | 3-4 | 4-2 | 4-3 |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| F_{jk} | 0 | 80 | 60 | 0 | 0 | 0 | 0 | 5 | 0 | 0 | 0 |
| B_{nm} | 80 | 60 | 0 | 5 | 80 | 0 | 5 | 0 | 60 | 0 | 0 |
| R_{nm} | 170 | 110 | 15 | 245 | 170 | 250 | 245 | 245 | 190 | 250 | 250 |

Tabela 3.23: Estado da rede após o estabelecimento do terceiro pedido.

| Ramo | 0-1 | 0-3 | 0-4 | 1-2 | 1-3 | 2-1 | 3-1 | 3-2 | 3-4 | 4-2 | 4-3 |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| F_{jk} | 0 | 80 | 60 | 80 | 0 | 0 | 0 | 5 | 0 | 0 | 0 |
| B_{nm} | 80 | 60 | 0 | 5 | 80 | 0 | 5 | 80 | 60 | 0 | 0 |
| R_{nm} | 170 | 110 | 15 | 165 | 170 | 250 | 245 | 165 | 190 | 250 | 250 |

Tabela 3.24: Estado da rede após o estabelecimento do quarto pedido.

e para os CR, pelos algoritmos alterado e original (com a alteração anterior e todas as correcções). As Figuras 3.10 e 3.11 ilustram as quatro iterações do código identificado como “ITERATIVE STEP” da versão original do algoritmo **LOCAL_EDGE_DISJOINT**(s, t) e as quatro iterações do ciclo “WHILE” (linha 4 a linha 17) da versão alterada do algoritmo **LOCAL_EDGE_DISJOINT**(s, t), respectivamente. Nessas figuras, em cada uma das iterações, os ramos candidatos a entrar na árvore de caminhos mais curtos são representados com uma linha contínua e os caminhos de protecção escolhidos para esses ramos são representados por linhas descontínuas. Sendo que, o caminho de protecção e o ramo protegido por este possuem a mesma cor, e a cor preta é usada para o próximo ramo a entrar na árvore de caminhos mais curtos. Para simplificar as figuras, em cada iteração apenas se mantêm das iterações anteriores a representação dos ramos que fazem parte da árvore.

Ambos os algoritmos começam por etiquetar definitivamente o nó 2 com etiqueta zero (e todos os outros com etiquetas ∞) após o que re-etiquetam temporariamente os nós adjacentes a esse (nós 1, 3 e 4), como é visível nas Figuras 3.10(a) e 3.11(a). Com o algoritmo alterado as etiquetas temporárias obtidas são 240, 240 e 160, respectivamente para os nós 1, 3 e 4, enquanto que com o algoritmo original são 180, 180 e 80 (cada um destes valores é igual ao correspondente an-

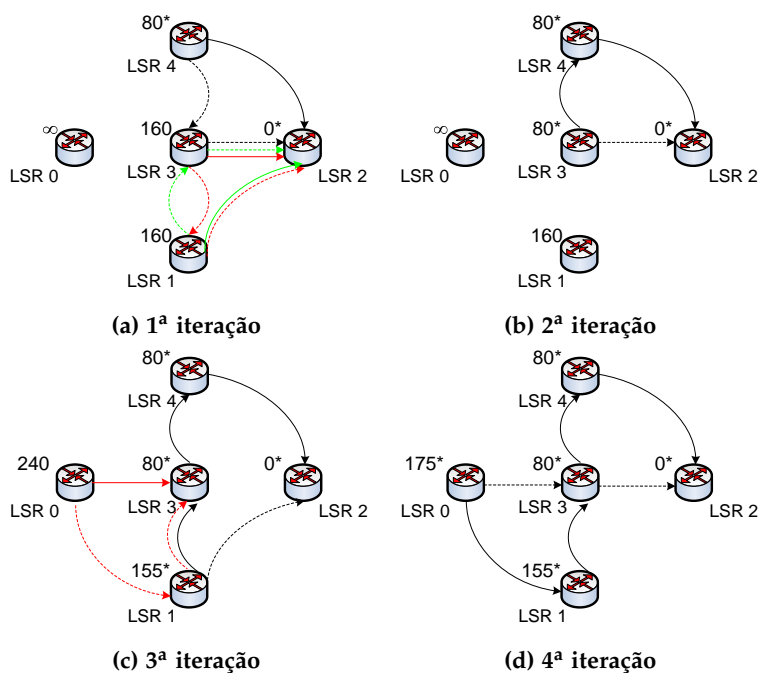


Figura 3.10: Iterações principais do algoritmo LOCAL_EDGE_DISJOINT(s, t) - versão original.

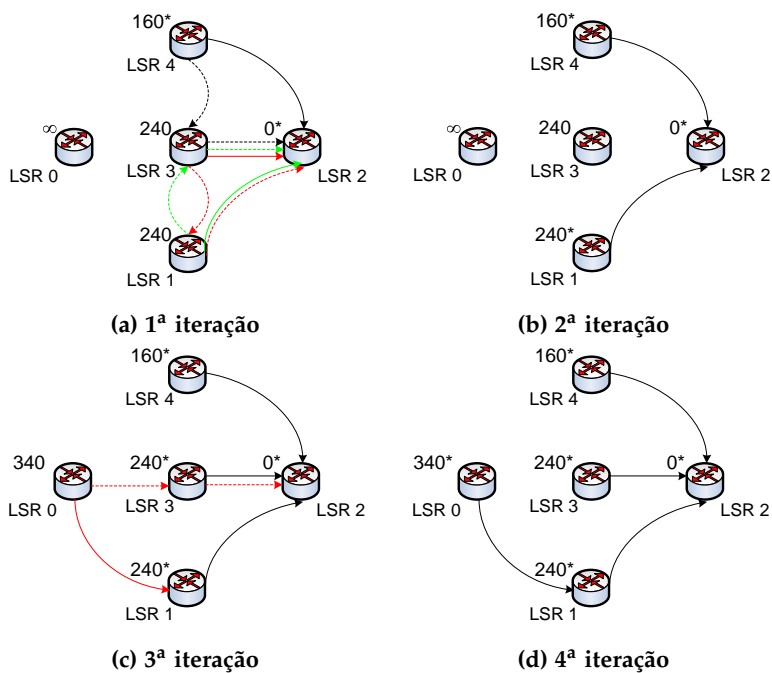


Figura 3.11: Iterações principais do algoritmo LOCAL_EDGE_DISJOINT(s, t) - versão alterada.

terior menos a LB do pedido). Ambos os algoritmos etiquetam definitivamente o nó 4 após o que re-etiquetam temporariamente o nó 3 (o nó 0 embora também seja adjacente não é re-etiquetado porque não possui LB livre suficiente para o pedido em causa), no algoritmo original passa a ter o valor 80 (etiqueta de LSR4 mais o custo da protecção do ramo LSR3-LSR4) enquanto que no alterado mantém o valor 240; pois o valor que obtém é também 240^1 (160 (etiqueta de LSR4) + 0 (custo da protecção do ramo LSR3-LSR4) + 80 (LB do pedido)). O algoritmo original etiqueta definitivamente o LSR3 e o algoritmo alterado o LSR1 (embora também pudesse ter sido etiquetado definitivamente o LSR3²) como é ilustrado nas Figuras 3.10(b) e 3.11(b), respectivamente. O algoritmo original na terceira iteração, ilustrada na Figura 3.10(c), re-etiqueta o LSR1 com 155 , 80 (etiqueta de LSR3) + 75 (custo da protecção do ramo LSR1-LSR3), e o LSR0 com 240 ($80 + 160$) e torna a etiqueta de LSR1 definitiva. Na mesma iteração o algoritmo alterado (ver Figura 3.11(c)) re-etiqueta o LSR0 com 340 , 240 (etiqueta de LSR1) + 20 (custo da protecção do ramo LSR0-LSR1) + 80 (LB do pedido) e torna definitiva a etiqueta de LSR3. Na última iteração o algoritmo original (ver Figura 3.10(d)) re-etiqueta o LSR0 com 175 , 155 (etiqueta de LSR1) + 20 (custo da protecção do ramo LSR0-LSR1), e torna a sua etiqueta definitiva. O algoritmo alterado apenas torna definitiva a etiqueta de LSR0 (ver Figura 3.11(d)).

Quando se utiliza o algoritmo alterado, este escolhe para CA o caminho LSR0-LSR1-LSR2, para CR do ramo LSR1-LSR2 o caminho LSR1-LSR3-LSR2 e para CR do ramo LSR0-LSR1 o caminho LSR0-LSR3-LSR2. Quando se utiliza o algoritmo original, este escolhe para CA o caminho LSR0-LSR1-LSR3-LSR4-LSR2, para CR do ramo LSR4-LSR2 o caminho LSR4-LSR3-LSR2, para CR do ramo LSR3-LSR4 o caminho LSR3-LSR2, para CR do ramo LSR1-LSR3 o caminho LSR1-LSR2 e para CR do ramo LSR0-LSR1 o caminho LSR0-LSR3-LSR2. Neste exemplo, embora ambos os algoritmos tenham partido do mesmo estado da rede, com o algoritmo alterado o CA obtido é formado por dois ramos enquanto que com o algoritmo original o CA obtido é formado por quatro ramos. As razões destas escolhas são o algoritmo original minimizar o custo dos CR e o algoritmo alterado minimizar o custo total do CA e dos CR.

Os estados da rede, após o estabelecimento deste pedido, são representados nas Tabelas 3.25 e 3.26, respectivamente para a versão original e alterada do

¹Como o novo valor é igual ao anterior – valores alternativos – o algoritmo poderia passar a utilizar o novo, o que significava que o LSR3 passava e ser etiquetado a partir de LSR4, como na versão original, no entanto o resultado final em termos de CA e CR seria o mesmo se em vez de manter o valor anterior se passasse a utilizar o novo.

²No entanto, também aqui o resultado final em termos de CA e CR seria o mesmo, ou seja é independente qual dos dois é etiquetado definitivamente em primeiro lugar.

| Ramo | 0-1 | 0-3 | 0-4 | 1-2 | 1-3 | 2-1 | 3-1 | 3-2 | 3-4 | 4-2 | 4-3 |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| F_{jk} | 80 | 80 | 60 | 80 | 80 | 0 | 0 | 5 | 80 | 80 | 0 |
| B_{nm} | 80 | 80 | 0 | 80 | 80 | 0 | 5 | 80 | 60 | 0 | 80 |
| R_{nm} | 170 | 110 | 15 | 165 | 170 | 250 | 245 | 165 | 190 | 250 | 250 |

Tabela 3.25: Estado da rede após o estabelecimento do quinto pedido - versão original.

| Ramo | 0-1 | 0-3 | 0-4 | 1-2 | 1-3 | 2-1 | 3-1 | 3-2 | 3-4 | 4-2 | 4-3 |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| F_{jk} | 80 | 80 | 60 | 160 | 0 | 0 | 0 | 5 | 0 | 0 | 0 |
| B_{nm} | 80 | 80 | 0 | 5 | 160 | 0 | 5 | 160 | 60 | 0 | 0 |
| R_{nm} | 90 | 90 | 15 | 85 | 90 | 250 | 245 | 85 | 190 | 250 | 250 |

Tabela 3.26: Estado da rede após o estabelecimento do quinto pedido - versão alterada.

| Pedido | Origem-Destino | LB |
|--------|----------------|----|
| 1º | LSR0-LSR3 | 10 |
| 2º | LSR0-LSR3 | 60 |

Tabela 3.27: Pedidos considerados no exemplo para a Alteração 3a.

algoritmo. É visível que, para satisfazer o 5º pedido, a LB de protecção aumentou 180 na versão alterada e apenas 175 na versão original, no entanto a quantidade LB necessária para os CA foi de 160, 80 (LB do pedido) * 2 (número de ramos do CA), na versão alterada e de 320, 80 (LB do pedido) * 4 (número de ramos do CA), na versão original, o que mostra que neste exemplo a LB total para satisfazer o pedido foi superior na versão original.

Alteração 3a

Para simplificar o exemplo vamos considerar que na rede da Figura 3.7 a capacidade dos ramos LSR0-LSR1 e LSR4-LSR2 é 50 unidades em vez das 250 lá indicadas. Nesta rede, tentamos estabelecer os pedidos apresentados na Tabela 3.27.

Na Tabela 3.28 é apresentado o estado da rede após o estabelecimento do primeiro pedido. Na determinação dos caminhos para o segundo pedido (Tabela 3.27), quando o algoritmo **SHORT_PRED_PATH**(k, u, j) procura um CR para o ramo LSR0-LSR3 ($j = 0$ e $k = u = 3$) começa por determinar os valores dos vectores δ e l .

Na versão original (este com todas as Correções) os valores obtidos são apresentados na Tabela 3.29 (determinam-se facilmente a partir da Tabela 3.28), que

| Ramo | 0-1 | 0-3 | 0-4 | 1-2 | 1-3 | 2-1 | 3-1 | 3-2 | 3-4 | 4-2 | 4-3 |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| F_{jk} | 0 | 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| B_{nm} | 10 | 0 | 0 | 0 | 10 | 0 | 0 | 0 | 0 | 0 | 0 |
| R_{nm} | 40 | 240 | 75 | 250 | 240 | 250 | 250 | 250 | 250 | 50 | 250 |

Tabela 3.28: Estado da rede após o estabelecimento do primeiro pedido (ver Tabela 3.27).

| Ramo | 0-1 | 0-3 | 0-4 | 1-2 | 1-3 | 2-1 | 3-1 | 3-2 | 3-4 | 4-2 | 4-3 |
|---------------|------|------|------|------|-----|------|------|------|------|------|------|
| δ_{nm} | 60 | 70 | 70 | 70 | 60 | 70 | 70 | 70 | 70 | 70 | 70 |
| l_{nm} | Inf. | Inf. | Inf. | Inf. | 60 | Inf. | Inf. | Inf. | Inf. | Inf. | Inf. |

Tabela 3.29: Valores dos vectores δ e l numa invocação de **SHORT_PRED_PATH**(k, u, j) na versão original do algoritmo, no estabelecimento do segundo pedido (ver Tabela 3.27).

| Ramo | 0-1 | 0-3 | 0-4 | 1-2 | 1-3 | 2-1 | 3-1 | 3-2 | 3-4 | 4-2 | 4-3 |
|---------------|------|------|-----|-----|-----|-----|-----|-----|-----|------|-----|
| δ_{nm} | 60 | 60 | 60 | 60 | 60 | 60 | 60 | 60 | 60 | 60 | 60 |
| l_{nm} | Inf. | Inf. | 60 | 60 | 60 | 60 | 60 | 60 | 60 | Inf. | 60 |

Tabela 3.30: Valores dos vectores δ e l numa invocação de **SHORT_PRED_PATH**(k, u, j) na versão modificada do algoritmo, no estabelecimento do segundo pedido (ver Tabela 3.27).

mostra que foi atribuído um custo “Infinito” a todos os ramos com excepção do ramo LSR1-LSR3. Com esses valores para os custos de utilização desses ramos o algoritmo de *Dijkstra* (código da linha 3 à linha 14) não encontra nenhum CR para protecção do ramo LSR0-LSR3, como é óbvio. Repare que, os valores de δ_{mn} são todos 70 ($F_{kj} + b = 60 + 10$) e portanto maiores que a LB do pedido (60 unidades), excepto para os ramos LSR0-LSR1 e LSR1-LSR3 uma vez que nestes o valor de B_{mn} é não nulo (pois estes são os ramos usados na protecção do primeiro pedido). Tendo sido essa a razão para quase a totalidade dos valores “Infinito” dos custos. O que levou a que o pedido tenha sido rejeitado apesar de haver LB disponível para o estabelecer.

Este problema é evitado, na versão modificada, através da alteração da atribuição a δ_{mn} (linha 1). Deste modo, o custo de um ramo com LB residual não inferior à LB do pedido (b) será quando muito igual a b (com excepção do ramo que se está a tentar proteger).

No exemplo, mas agora com a versão modificada, na determinação dos caminhos para o segundo pedido, quando o algoritmo **SHORT_PRED_PATH**(k, u, j) procura um CR para o ramo LSR0-LSR3 ($j = 0$ e $k = u = 3$) os valores que obtém para os vectores δ e l são apresentados na Tabela 3.30. O que resulta na escolha do caminho LSR0-LSR3 para CA e do caminho LSR0-LSR4-LSR3 para CR.

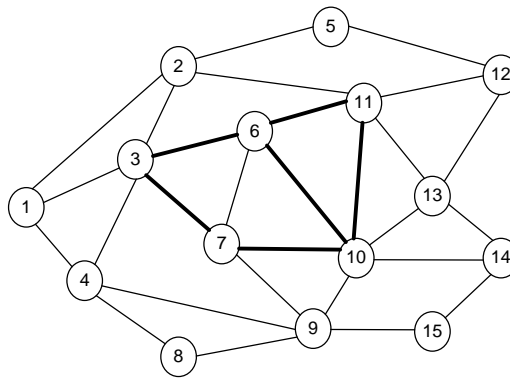


Figura 3.12: Topologia da rede (adaptada de Kodialam e Lakshman, 2000).

3.2.4 Resultados Simulacionais

Não se vai tentar replicar os resultados em Raza *et al.* (2005) por várias razões. Primeiro porque em Raza *et al.* (2005) é dito que o seu critério de encaminhamento dos pedidos é minimizar a soma da LB reservada para o CA e para os CR, mas não explicam como o fazem. Segundo porque em Raza *et al.* (2005) é dito que foi usada a heurística proposta em Kodialam e Lakshman (2001), contudo como mostrado na Secção 3.2.2, essa heurística, como publicado em Kodialam e Lakshman (2001, 2002a) apresenta algumas gralhas e incorrecções que precisam de ser corrigidas. Por último, porque não foi mencionado o mecanismo para reserva de LB.

Em **SHORT_PRED_PATH**(k, u, j) (passo 2) o teste $\delta_{mn} \leq b$ poderá originar a rejeição de pedidos apesar de existir LB suficiente para os satisfazer. Este comportamento será ilustrado numa rede com capacidade infinita, em que esta condição leva a uma taxa de rejeição extremamente elevada! No contexto de reserva exacta esta restrição pode ser removida pois não influenciará a LB reservada para os CR.

O algoritmo **LOCAL_EDGE_DISJOINT**(s, t) na sua forma original, com as correcções da Secção 3.2.2 será designado por LED; o mesmo algoritmo na sua forma original, também com as correcções da Secção 3.2.2 e com o teste $\delta_{mn} \leq b$ removido no passo 2 de **SHORT_PRED_PATH**(k, u, j) será designado por LEDb; a versão melhorada do algoritmo **LOCAL_EDGE_DISJOINT**(s, t), com todas as correcções e alterações propostas na Secção 3.2.2 será designado por ILED.

A topologia de rede considerada para a análise é apresentada na Figura 3.12. Trata-se de uma rede com 15 nós e 56 ramos dirigidos (correspondem a 28 ramos não dirigidos). São apresentados resultados para duas experiências: uma em que a capacidade dos ramos é infinita e outra em que todos os ramos têm capacidades finitas. Para a segunda experiência, a capacidade de cada ramo fino é de 60

unidades e de cada ramo grosso é de 240 unidades, em ambos os casos em cada direcção. Em cada experiência chegam à rede pedidos (um de cada vez) de LB uniformemente distribuída no intervalo [6, 9], como em Kodialam e Lakshman (2002b). A origem e o destino dos LSP são escolhidos aleatoriamente. Foram executadas dez corridas para cada uma das experiências seguintes em que a reserva de LB de protecção foi efectuada de forma exacta de acordo com o método proposto na secção 3.1.2.

Primeiro considera-se o caso de tráfego incremental (os LSP são estabelecidos com duração infinita) com capacidade infinita nos ramos da rede. Cada corrida de simulação, para este caso, termina quando forem solicitados 10001 pedidos. A Figura 3.13 mostra, para LED e LEDb, o número total de pedidos aceites versus o número total de pedidos solicitados. É claramente visível, na Figura 3.13, o efeito da remoção ou não da restrição $\delta_{mn} \leq b$: LEDb aceita todos os pedidos como esperado e LED apresenta uma taxa de rejeição de pedidos muito alta (maior que 70% depois dos primeiro 1000 pedidos). O comportamento de LED face aos pedidos solicitados pode ser visto em mais detalhe na Figura 3.14, foram aceites menos de 650 pedidos dos 10001 solicitados.

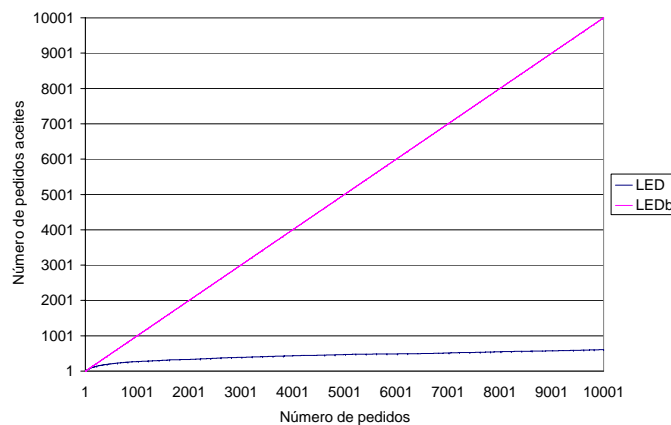


Figura 3.13: Pedidos aceites, usando tráfego incremental numa rede de capacidade infinita.

Numa rede de capacidade infinita e usando tráfego incremental não é necessário comparar, no que concerne à utilização de LB, ILED (ou LEDb) com LED, pois apenas mostraria que este último utiliza menos LB devido à rejeição de pedidos, como é visível nas Figuras 3.14 e 3.13. Assim sempre que for usada uma rede de capacidade infinita, ILED será comparado apenas com LEDb. Na Figura 3.15 apresenta-se a LB usada por ILED e LEDb na rede com capacidade infinita usando tráfego incremental. Pode observar-se que ambos os algoritmos utilizam quantidade semelhantes de LB de protecção mas ILED apresenta significativa-

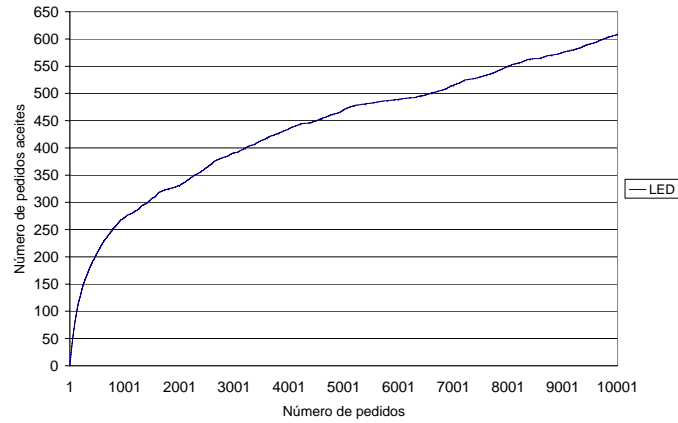


Figura 3.14: Pedidos aceites, usando tráfego incremental numa rede de capacidade infinita, para melhor visualização do comportamento de LED.

mente melhor desempenho quanto à LB requerida pelos CA.

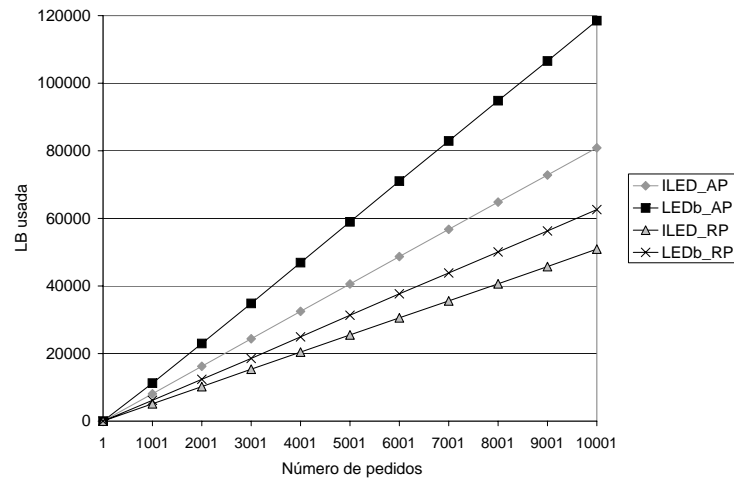


Figura 3.15: Largura de banda usada por CA e CR, numa rede de capacidade infinita com tráfego incremental.

No final das corridas de simulação o comprimento médio dos CA (número de nós) é cerca de 2,3 para ILED e 3,5 para LEDb. O número médio de ramos diferentes necessários para proteger cada CA é 6,4 para ILED e 9 para LEDb. Por conseguinte ILED reserva menos LB do que LEDb porque utiliza CA mais curtos (com menor número de ramos) e embora ILED use nos CR menos ramos (diferentes) do que LEDb, ambos os algoritmos apresentam consumo idêntico de LB de protecção.

Na experiência com do tráfego dinâmico e capacidade finita nos ramos, cada corrida de simulação termina após 10001 pedidos. As chegadas seguem um processo de Poisson de intensidade λ , e a duração da ligações são exponencial-

mente distribuídas com valor médio igual a $1/\mu$. Para avaliar o desempenho dos algoritmo com $\lambda/\mu = 150$ foram igualmente utilizadas 10 corridas de simulação, assim como pedidos de LB uniformemente distribuídos no intervalo [6, 9] e com origem e destino dos LSP escolhidos aleatoriamente.

Nesta experiência os resultados obtidos mostram que ILED tem probabilidade de rejeição de novos pedidos, P_b , significativamente inferior a LEDb e LED, como pode ser visto na Figura 3.16 (foi usado um período de aquecimento de 2500 pedidos). Por sua vez LEDb apresenta melhor desempenho do que LED o que vem confirmar a pertinência da remoção da restrição $\delta_{mn} \leq b$ no passo 2 de **SHORT_PRED_PATH**(k, u, j). ILED utiliza em média menos ramos nos CA, do que LEDb ou LED, o que tende a reduzir o tempo de propagação do tráfego na rede. Por outro lado o número de ramos diferentes utilizados por ILED nos CR é em média também inferior aos requeridos por LEDb e LED.

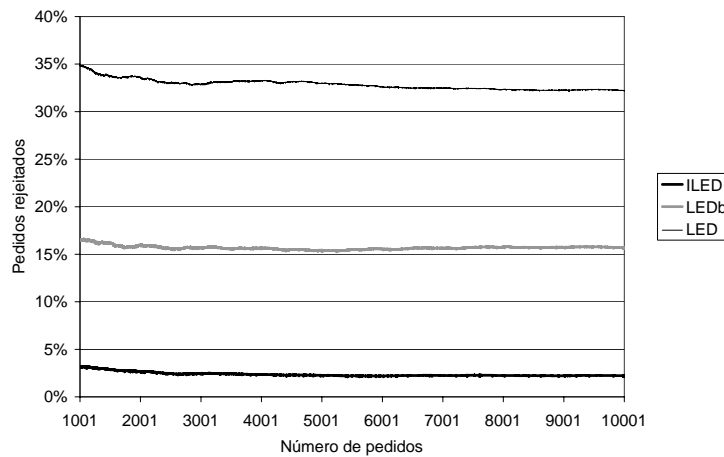


Figura 3.16: Probabilidade de rejeição de pedidos, numa rede de capacidade finita com tráfego dinâmico.

Na Tabela 3.31 são apresentados resultados para P_b , para o número médio de ramos usados no CA e para o número de ramos diferentes utilizado pelos CR para os 10001 pedidos considerados. Nesta tabela é apresentado o valor médio \pm metade do intervalo de confiança. Os intervalos de confiança apresentados foram determinados de acordo com a Fórmula (F.10) apresentada no Apêndice F, o grau de confiança utilizado foi de 95%.

Observamos que é frequente o algoritmo obter CR longos. Este aspecto poderá ser melhorado: (a) através da escolha do caminho com menor número de ramos, entre caminhos alternativos (de igual custo) no algoritmo **ALT_PATH_COST**(k, j); (b) permitindo em **SHORT_PRED_PATH**(k, u, j) a alteração da etiqueta e do predecessor de um nó, quando os custos são iguais, garantindo desta forma

| | ILED | LEDb | LED |
|-----------|-----------------|------------------|------------------|
| P_b (%) | 2.23 ± 0.22 | 15.69 ± 0.71 | 32.22 ± 1.57 |
| #CA | 2.47 ± 0.01 | 3.74 ± 0.03 | 3.91 ± 0.05 |
| #CR | 6.97 ± 0.15 | 9.89 ± 0.26 | 10.05 ± 0.31 |

Tabela 3.31: Tráfego dinâmico com capacidade finita na rede, quando $\lambda/\mu = 150$. #AP é o número médio de ramos nos CA; #RP é o número médio de ramos diferentes nos CR.

que o caminho obtido até ao nó j é um caminho de custo mínimo com o menor número de ramos possível.

Análise dos resultados

Foram corrigidas algumas gralhas e imprecisões de um algoritmo de encaminhamento dinâmico com protecção local e foi proposta uma nova versão para esse algoritmo. As alterações introduzidas foram todas devidamente justificadas e ilustradas e conduziram a uma versão mais eficiente.

O desempenho do algoritmo original e da versão melhorada do algoritmo, utilizando reserva mínima de LB de protecção foi analisada usando tráfego incremental e dinâmico. Em ambas as experiências, o número médio de ramos utilizado nos CA é menor em ILED do que em LEDb ou LED. Usando tráfego incremental, os resultados experimentais mostraram que a nova versão utiliza menos LB, nomeadamente nos CA. No estudo experimental utilizando tráfego dinâmico o bloqueio médio na rede foi significativamente menor quando se utiliza ILED. As conclusões anteriores relativamente ao número médio de ramos utilizado nos CA podem neste caso verificar-se também no comprimento médio do CR mais longo de cada CA.

As várias versões do algoritmo obtêm com frequência CR longos, embora tal seja menos visível em ILED. Foi sugerida uma estratégia para mitigar essa característica.

Em resumo, pode afirmar-se que as alterações propostas tornam o algoritmo mais eficiente em termos de consumo de LB e também relativamente aos tempos de propagação do tráfego nos CA. O algoritmo mesmo com as alterações mantém-se uma heurística "greedy" uma vez que as escolhas iniciais podem impedir conseguir uma maior partilha no futuro.

3.3 Caminhos disjuntos de custo mínimo com custos duais nos arcos

Esquemas que fornecem protecção global precisam de encontrar dois caminhos (o CA e o CR) disjuntos entre a origem e o destino do pedido. Se a intenção do esquema for simplesmente minimizar a LB usada por estes caminhos, Suurballe e Tarjan (1984) apresentam um algoritmo que pode ser usado para determinar a qual par (CA, CR) corresponde o mínimo de LB total usada, num tempo que cresce polinomialmente com o número de nós da rede. No entanto, se se assumir que as falhas são raras e não afectam mais do que um ramo de cada vez (e que portanto não existe em cada momento mais do que um ramo afectado por falha na rede), é possível reutilizar LB já reservada para um determinado ramo de um CR para proteger um ramo de um CA que seja distinto do CA original para o qual a LB de protecção foi reservada. Isto significa que o “custo” em LB associado a usar determinado ramo num CR, para proteger um ramo a usar no CA, poderá ser um valor diferente (e inferior) ao “custo” de usar o mesmo ramo no CA. O problema de encontrar qual o par óptimo (CA, CR) que minimiza a LB usada neste caso é um problema mais complexo, que se demonstrou ser NP-completo (Laborczi *et al.*, 2001; Xu *et al.*, 2004).

3.3.1 Partilha de Largura de Banda na Protecção Global

Em Kodialam e Lakshman (2003) são propostos diversos esquemas que pretendem encontrar soluções aceitáveis para o problema de encontrar o par (CA, CR) que minimiza a LB usada, assumindo partilha de LB de protecção entre diversos CA. Estes esquemas diferem entre si no conhecimento requerido sobre a relação entre os arcos de cada CR e os arcos de cada CA por aqueles protegidos.

Como já indicado na Secção 2.4.1, se essa informação não estiver disponível de todo (no cenário chamado de “sem informação”), o problema não permite partilha, e portanto o algoritmo de Suurballe e Tarjan (1984) pode ser usado para encontrar dois caminhos disjuntos usando apenas o conhecimento da LB total disponível em cada ramo. Se se souber exactamente δ_{ij}^{uv} , ou seja, qual é a LB usada no arco (i, j) para os CA que é protegida no arco (u, v) pelos CR para todos os arcos (i, j) e (u, v) , o problema de minimizar os custos nesta situação, designada de cenário de informação completa, pode ser formulado e resolvido como um problema de programação linear inteira.

A quantidade de informação que é necessário partilhar para conseguir informação completa (informação por cada LSP criado) torna esta situação pouco re-

alista. Por esta razão, é proposto também em Kodialam e Lakshman (2003) um algoritmo para a situação em que não sabendo exactamente qual ramo do CA é protegido por qual ramo(s) do CR, se sabe porém qual o total de LB que é usada em cada arco (i, j) para CA (F_{ij}) e qual é usada no mesmo ramo para CR (B_{ij}). Usando apenas esta informação (no cenário chamado de informação agregada) não é possível partilhar tanta LB de protecção como perante informação completa, mas é ainda possível formular o problema a resolver como um de programação linear inteira, apenas com valores menos precisos (e portanto desperdiçadores de LB que poderia ser partilhada).

Por esta formulação ser pouco eficiente, é reformulada como um processo iterativo que resolve sucessivamente o problema de calcular o par de caminhos disjuntos de custo mínimo. As iterações diferem entre si porque em cada iteração os arcos que podem constituir o CA são limitados de forma a que o valor máximo de F_{ij} no CA seja inferior ou igual a um valor $M = \max_{(i,j) \in AP} F_{ij}$ (ou seja, para um determinado valor de M tiram-se de consideração para o CA todos os arcos para o qual $F_{ij} > M$). O algoritmo funciona calculando o custo agregado do par de caminhos de custo mínimo para todos os valores de M admissíveis (de 0 a $\max_{(i,j)} F_{ij}, \forall (i, j) \in E'$) e escolhendo o menor destes. O custo, c_{uv} , de usar o arco (u, v) no CR é em cada iteração calculado como função da LB do pedido, de M e de B_{uv} , enquanto o custo, a_{ij} , de usar o arco (i, j) no CA, é simplesmente a LB do pedido (ou infinito, se o arco não estiver disponível). Por conseguinte em Kodialam e Lakshman (2003) foi abordado o problema do cálculo de pares de caminhos disjuntos de custo mínimo com custos duais, ou seja em que associados a cada arco existem dois custos: o custo de utilização no CA e o custo de utilização no CR. Esse problema foi aí resolvido usando uma resolução heurística do problema de programação linear inteira através da relaxação do mesmo como um problema de programação linear e calculando aproximações sucessivas ao óptimo do primal e do dual do problema relaxado. Estas aproximações, requerendo apenas a resolução de problemas simples de caminho mais curto, podem ser resolvidas rapidamente. Tratando-se de uma abordagem heurística, não garante mais do que a possibilidade de aproximação assintótica ao óptimo, mas os autores consideram que tal é compensado pela velocidade de resolução.

3.3.2 Um algoritmo para obter um par de caminhos disjuntos de custo mínimo com custos duais nos arcos

Uma abordagem alternativa ao problema de calcular pares de caminhos disjuntos de custo mínimo com custos duais (c^1 e c^2) foi proposta em Gomes *et al.* (2006b)³. Nesta abordagem pretendeu-se resolver o problema usando um algoritmo de k -caminhos mais curtos (foi usado o algoritmo MPS, descrito em Martins *et al.*, 1999) e o algoritmo de *Dijkstra*. O MPS foi usado para obter um caminho (p) usando um conjunto de custos e o algoritmo de *Dijkstra* usou o outro conjunto de custos para obter outro caminho ($q(p)$) num conjunto de arcos aos quais foram retirados os arcos presentes em p .

O algoritmo (designado por DP2LC e descrito em Gomes *et al.*, 2006a,b) funciona usando simultaneamente a abordagem para os dois conjuntos de custos. Para tal são criados dois conjuntos (A e B) de pares de caminhos, A usando c^1 com o algoritmo MPS e c^2 com o algoritmo de *Dijkstra*, e B usando c^2 com o MPS e c^1 com o *Dijkstra*. É possível provar (Gomes *et al.*, 2006b) que usando em cada momento os elementos de A e B que apresentam menor custo agregado (sejam estes A_h com custo agregado $C(A_h) = c^1(p_{A_h}) + c^2(q(p_{A_h}))$ e B_k com custo agregado $C(B_k) = c^2(p_{B_k}) + c^1(q(p_{B_k}))$) é possível obter a partir destes um limiar superior (*upper bound*) e um limiar inferior (*lower bound*) para o valor da solução de menor custo agregado. Após encontrar um primeiro par de caminho admissíveis em cada conjunto é possível verificar se se encontrou o óptimo (se os dois limiares forem coincidentes) ou então procurar estreitar os limiares, procurando novos pares (admissíveis) em qualquer dos conjuntos que apresentem custos agregados menores que o custo agregado já encontrado para esse conjunto. A pesquisa de novos pares efectua-se de forma alternada nos dois conjuntos, para que de cada vez que se encontre um novo par com custos agregados menores que o menor desse conjunto, se possa obter um novo limiar inferior, maior ou igual ao anterior. Se o custo agregado encontrado for menor que todos os encontrados até então obtêm-se também um novo limiar superior, inferior ao anterior, e a pesquisa passa a efectuar-se no outro conjunto.

Na presença de recursos computacionais (memória e tempo de CPU) ilimitados, o algoritmo permite resolver de forma exacta o problema. No entanto, se se considerarem limitações nestes recursos (algo que é provável em aplicações de encaminhamento com um tempo limitado para chegar à solução) o algoritmo pode chegar a soluções sub-óptimas: aquelas em que não foi possível atingir a

³A autora desta dissertação esteve envolvida neste e no próximo algoritmo na área da definição das experiências para validação dos mesmos e na obtenção e análise dos resultados, mas não na criação dos algoritmos.

| | \mathcal{E} | $\bar{\mathcal{E}}$ |
|---------------------|--|---|
| \mathcal{Z} | $([0, 100]; [0, 100])$ $([0, 10000]; [0, 10000])$ | $([0, 10]; [0, 10000])$ $([0, 100]; [0, 10000])$ |
| $\bar{\mathcal{Z}}$ | $([1, 100]; [1, 100])$ $([1, 10000]; [1, 10000])$ | $([1, 10]; [1, 10000])$ $([1, 100]; [1, 10000])$ |

Tabela 3.32: Símbolos correspondentes a gamas de custos.

solução óptima (ou mesmo que tenha sido possível atingi-la não se conseguiu verificar a optimalidade da mesma). O algoritmo permite mesmo nestes casos obter uma medida aproximada da qualidade de uma solução sub-óptima, através do cálculo da distância entre os limiares superior e inferior. O número total de iterações a realizar para atingir o óptimo depende da topologia da rede (em princípio, se existirem muitos caminhos alternativos com o mesmo custo a progressão em direcção ao óptimo será mais lenta) mas a existência de limiares permite também considerar a hipótese de parar quando o benefício potencial máximo de atingir o óptimo for inferior a um determinado valor predefinido.

Este algoritmo foi validado através de um conjunto de experiências que pretenderam verificar a adequação do algoritmo para utilização prática, quer em termos de tempo necessário para o cálculo quer em termos de distância máxima ao óptimo nas soluções sub-óptimas e qual o seu número. Para tal foram geradas aleatoriamente conjuntos de redes com diferentes números de nós, densidades diferentes e diferentes valores para as gamas de custos. A análise foi efectuada tanto em redes dirigidas com em redes não dirigidas. Nesta secção são apresentados alguns resultados para o caso de redes dirigidas, do qual um conjunto mais alargado de resultados pode ser encontrado em Gomes *et al.* (2006a). Resultados para o caso particular de redes não dirigidas foram também publicados em Gomes *et al.* (2006c).

Numa primeira experiência, foram geradas redes dirigidas conexas com um número de nós n com valores de 50, 100, 150, 200, 250, 300, 350, 400, 450, 500, 600 e 800. Para cada uma destas foram gerados um número de arcos m correspondente a $3n$, $4n$ e $6n$. Para estas redes foram usados custos duais gerados aleatoriamente por ramo usando gamas predefinidas de valores possíveis para cada um dos conjuntos de custos. As combinações de gamas escolhidas (8 no total) encontram-se referenciadas na Tabela 3.32, e classificadas em função das suas propriedades: assim, foram geradas gamas de custos “iguais” – ou melhor, aleatórias sobre um mesmo domínio – denotadas por gamas \mathcal{E} – e gamas em que tal não ocorre – gamas $\bar{\mathcal{E}}$; e foram geradas gamas que permitem arcos de custo

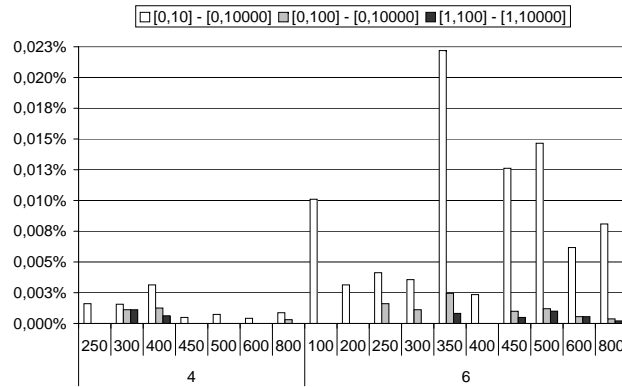


Figura 3.17: Percentagem de soluções sub-óptimas, em rede com soluções sub-óptimas para $m = 4n, 6n$ e gamas $\bar{\mathcal{E}}$.

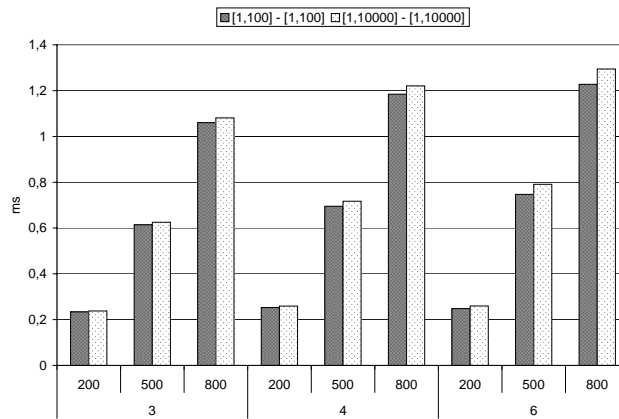
nulo⁴ – gamas \mathcal{Z} – e gamas que não incluem a possibilidade de custos nulos – gamas $\bar{\mathcal{Z}}$. Nesta experiência foram calculados caminhos disjuntos de custo mínimo para todos os pares existentes (no total $n \times (n - 1)$ em cada rede). Os valores apresentados para cada valor de m, n e gama de custos são médias obtidas em conjuntos de 10 redes.

A experiência foi executada com limitações de memória máxima usada, mas sem limitações de tempo de processamento. No respeitante à capacidade de atingir o óptimo, os resultados das experiências mostram que tal apenas não ocorreu para a gama $\bar{\mathcal{E}}$ com densidades $m = 4n$ e $m = 6n$. Mesmo nestas redes o óptimo foi atingido em mais de 99.97% dos casos (ver Figura 3.17, sendo de notar que apenas se calcularam percentagens para as redes onde existiram soluções sub-óptimas).

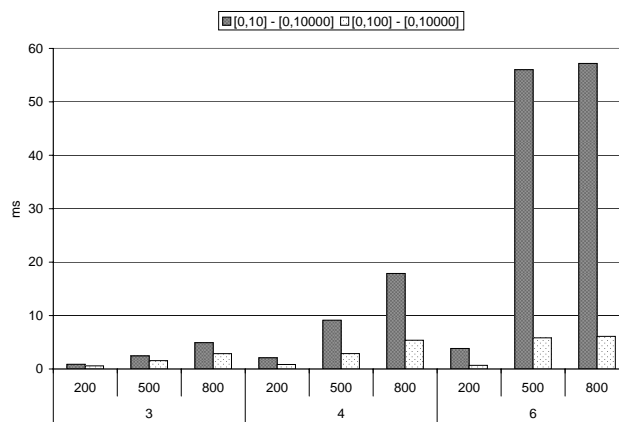
Em termos de tempo de processamento, mesmo incluindo o tempo dos pares em que não se chegou ao óptimo (e que demoraram até terminar muito mais que a média) o tempo por par calculado foi bastante pequeno. Nas gamas E o tempo foi inferior a $2ms$ mesmo nas redes mais densas e com mais nós (como se pode verificar na Figura 3.18a, sendo de notar que esta gama não apresentou soluções sub-óptimas). Mesmo em gamas que apresentaram soluções sub-óptimas, como foi o caso das gamas $\mathcal{Z}\bar{\mathcal{E}}$ apresentadas na Figura 3.18b o tempo médio de CPU por par não excedeu os $60ms$.

No entanto, como podemos concluir combinando estes resultados com os anteriores (e foi verificado noutros resultados), tal significou que nalguns pares o algoritmo demorou tempo significativo, antes de atingir o óptimo ou de terminar com solução sub-óptima. Estes resultados parecem apontar para a vantagem

⁴Deve ser notado que embora os custos nulos sejam possíveis nas gamas \mathcal{Z} , as dimensões efectiva das gamas tornam-os na realidade pouco frequentes.



(a) Gamas $\bar{Z}\bar{\mathcal{E}}$



(b) Gamas $\bar{Z}\bar{\mathcal{E}}$

Figura 3.18: Tempo médio de CPU por par de nós para $n = 200, 500, 800$ e $m = 3n, 4n, 6n$.

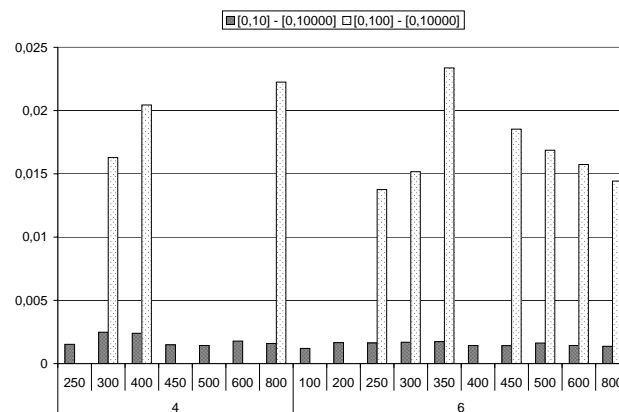


Figura 3.19: Desvio normalizado máximo associado às soluções sub-óptimas, para $m = 4n, 6n$ e gamas $\bar{Z}\bar{\mathcal{E}}$.

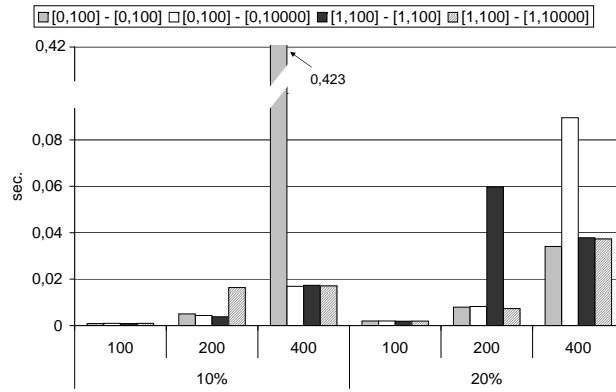
de limitar em aplicações reais o tempo de CPU a atribuir ao algoritmo para a resolução do problema, no pressuposto de que o valor máximo de desvio do ótimo associado às soluções sub-ótimas não é elevado, o que se verificou nesta experiência, ver Figura 3.19. Nesta figura o valor apresentado corresponde ao intervalo entre limiares na solução sub-ótima, dividido pelo valor do limiar inferior, o que permite de alguma forma apresentar um valor normalizado para o desvio máximo em relação ao ótimo, e que se verifica nunca ser superior a 3% do valor do limiar inferior

Uma segunda experiência realizada com este algoritmo comparou o algoritmo DP2LC com a formulação do problema como um problema de programação linear inteira (designado algoritmo PLI), resolvido usando a ferramenta *ILOG CPLEX* 10.110 na mesma máquina (Pentium IV a 3.2GHz com 2GB de RAM). Para esta experiência usaram-se redes mais densas (densidades iguais a 10% e 20% do número de arcos totais possíveis) e valores de n mais espaçados ($n = 50, 100, 200, 400, 800, 1600, 3200$), com as mesmas gamas de custos referidas anteriormente. Para este exemplo foram testados apenas 100 pares de nós para cada rede, gerados aleatoriamente.

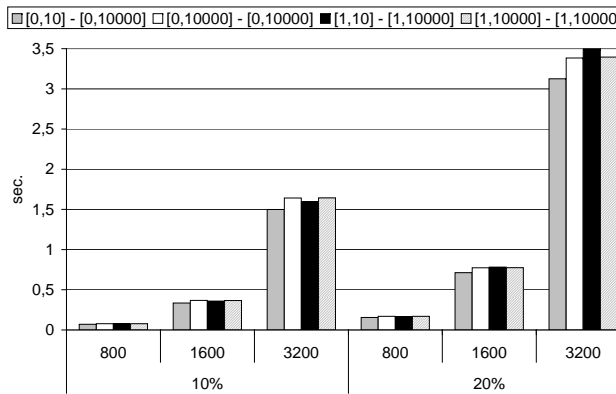
Esta comparação permitiu verificar que o algoritmo DP2LC obtém resultados mais rapidamente que a implementação PLI (desde cerca de 10 vezes mais rápido em redes maiores e mais densas até cerca de 25 vezes mais rápido em redes menores e menos densas). A Figura 3.20 ilustra estes resultados, mas também uma maior volatilidade em termos de tempo necessário por resolução no algoritmo DP2LC, em contraste com durações muito mais uniformes na resolução usando o CPLEX. O algoritmo DP2LC gerou também neste caso algumas soluções sub-ótimas, mas raramente (115 casos em 112000 pares testados, ou seja, perto de 0,1%). A implementação PLI não conseguiu resolver o caso da maior rede com maior densidade.

3.3.3 Um algoritmo para obter o conjunto de pares de caminhos disjuntos de custo mínimo com custos duais nos arcos

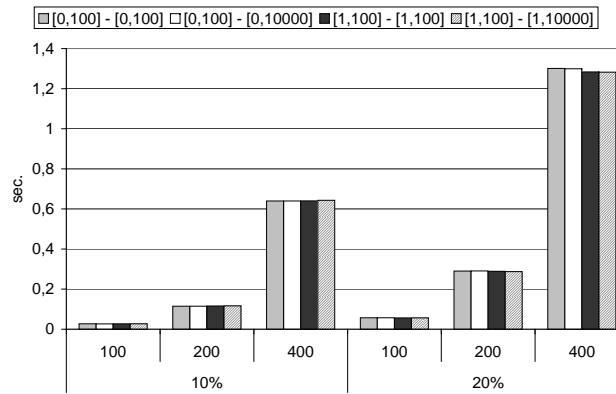
Embora o algoritmo anterior funcione adequadamente para obter uma solução para o problema de obtenção de pares de caminhos disjuntos de custo mínimo com custos duais nos arcos, uma solicitação adicional que pode ocorrer em recuperação pode passar por determinar não apenas uma solução ótima para o problema mas sim o conjunto de todas as soluções ótimas (por exemplo, se se pretender fazer uma análise posterior dos caminhos obtidos usando outros critérios que não apenas a LB ocupada).



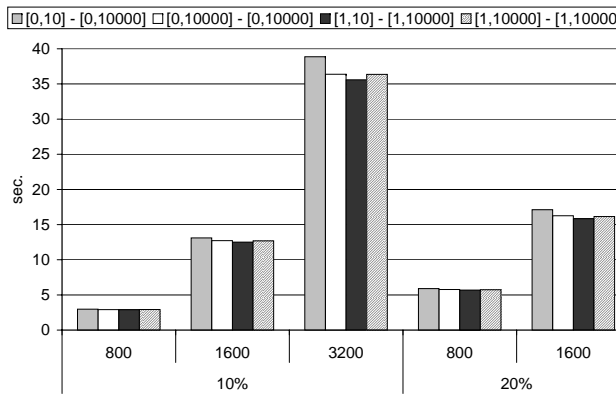
(a) Algoritmo DP2LC



(b) Algoritmo DP2LC



(c) Algoritmo PLI



(d) Algoritmo PLI

Figura 3.20: Tempo médio de CPU por par de nós para $n = 100, 200, 400, 800, 1600, 3200$ e $m = 10\%, 20\%$.

Um algoritmo (designado por Set2LC e descrito em Gomes *et al.*, 2006a) para obter o conjunto de todos os caminhos de custo agregado mínimo pode ser obtido por modificação do algoritmo anterior. Agora, em vez de guardar nos conjuntos A e B , em cada momento, apenas um par de caminhos com o custo agregado mínimo obtido até então para cada um destes, ir-se-á guardar todo o conjunto de pares de caminhos com esse mesmo custo mínimo, em S_A e S_B respectivamente. Os conjuntos S_A e S_B são usados pelo algoritmo de forma semelhante ao definido para A_h e B_k no algoritmo anterior (apenas tendo em consideração que os conjuntos irão potencialmente guardar em cada momento mais do que um par de caminhos, mas todos com o mesmo custo agregado – o menor encontrado até ao momento para esse conjunto) até conterem um cada um destes pelo menos um par de caminhos de custo agregado óptimo. Nesta altura será efectuada uma pesquisa por outros pares de caminhos que possuam o mesmo custo agregado (através da obtenção pelo algoritmo MPS de novos caminhos com o mesmo custo e pelo *Dijkstra* de novos caminhos mínimos para a rede restante), sendo estes de seguida adicionados a S_A ou S_B . De seguida os conjuntos são unidos (para eliminar potenciais duplicados, ou seja pares que apareçam em ambos os conjuntos). Finalmente são gerados pares de caminhos que eventualmente não tenham sido detectados (recombinando os caminhos já existentes nos pares do conjunto, de forma a criar novos pares de caminhos disjuntos com custo agregado óptimo – embora tenha de ser feita filtragem porque nem todas as combinações resultam em pares disjuntos).

Este algoritmo foi validado através de uma experiência que pretendeu verificar a adequação do algoritmo para utilização prática, quer em termos de tempo necessário para o cálculo quer em termos de dimensão dos conjuntos obtidos. Para tal foram geradas aleatoriamente conjuntos de redes com diferentes números de nós, densidades diferentes e diferentes valores para as gamas de custos apresentadas na Tabela 3.32. Para esta experiência, foram geradas redes dirigidas conexas com um número de nós n com valores de 50, 100, 150, 200, 250, 300, 350, 400, 450, 500, 600, 800 e 1000. Para cada uma destas foram gerados um número de arcos m correspondente a $3n$ e $4n$ (novamente foram geradas 10 redes para cada valor de n , m e gama dos custos). Foram gerados os conjuntos de soluções para todos os pares de nós de cada rede. Para evitar tempos de cálculo muito elevados, o tempo de CPU associado a cada par foi limitado a 50ms (esta duração foi escolhida por ser uma duração comum do tempo máximo de recuperação esperado em redes de telecomunicações). Isto significa que o algoritmo pode terminar com o conjunto completo de soluções óptimas, com um conjunto que pode não estar completo ou mesmo sem ter a certeza de ter soluções óptimas no

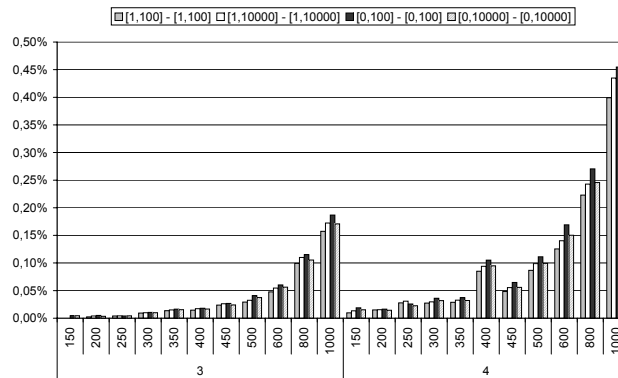
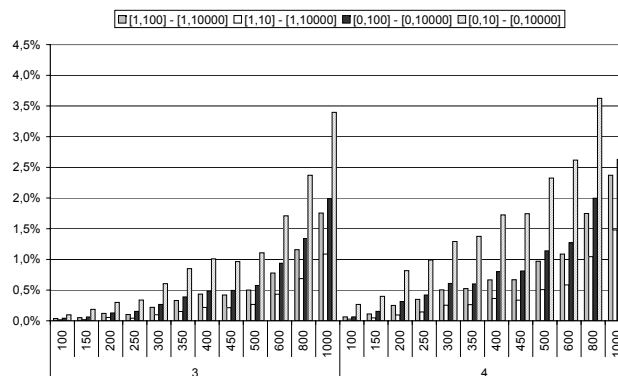
(a) Gamas \mathcal{E} (b) Gamas $\bar{\mathcal{E}}$

Figura 3.21: Percentagem de soluções sub-óptimas para redes com soluções sub-óptimas para $m = 3n$ e $m = 4n$.

conjunto final (a última condição é considerada como a obtenção de um conjunto de soluções sub-óptimas). Apenas alguns resultados são apresentados aqui, um conjuntos mais alargado pode ser encontrado em Gomes *et al.* (2006a).

Como é visível na Figura 3.21a obtenção de conjuntos de soluções sub-óptimas é relativamente rara para custos na gama \mathcal{E} (não ocorrem de todo para $n < 150$ e nunca é superior a 0,5% dos pares). No entanto, para a gama $\bar{\mathcal{E}}$, representada na Figura 3.21b, podemos verificar que as soluções sub-óptimas são mais comuns (mesmo assim não ocorrem para $n = 50$ e a sua frequência nunca é superior a 5% dos pares).

Na Figura 3.22 pode observar-se que o tempo médio de CPU necessário por par é reduzido (inferior em geral a $4ms$ nas redes com gamas de custo \mathcal{E} e inferior a $8ms$ nas redes com gamas de custo $\bar{\mathcal{E}}$). Isto ocorre porque embora as soluções sub-óptimas tenham, por definição, uma duração de $50ms$ de tempo de CPU, a sua relativa raridade não afecta de forma significativa a média nas redes \mathcal{E} e o

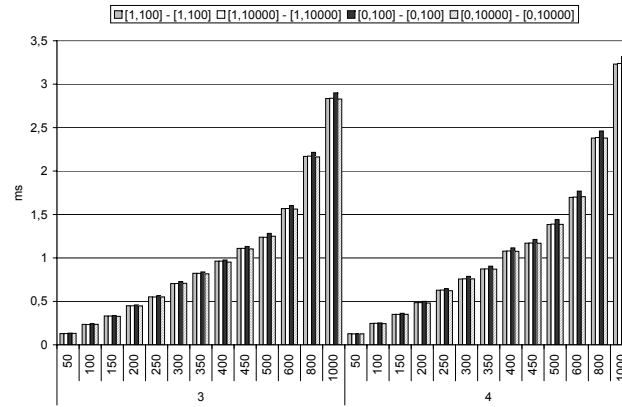
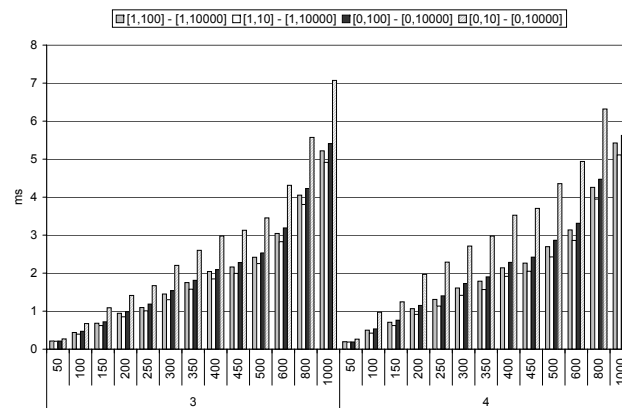
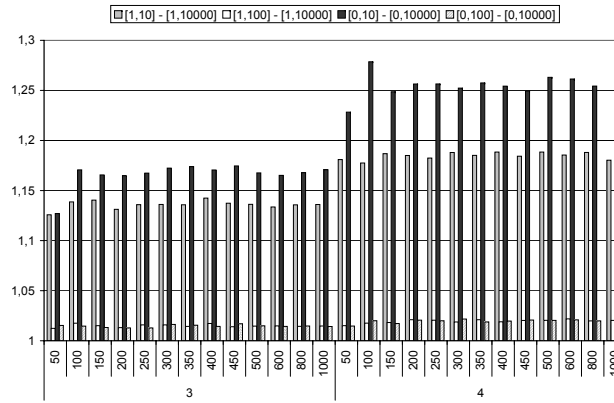
(a) Gamas \mathcal{E} (b) Gamas $\bar{\mathcal{E}}$

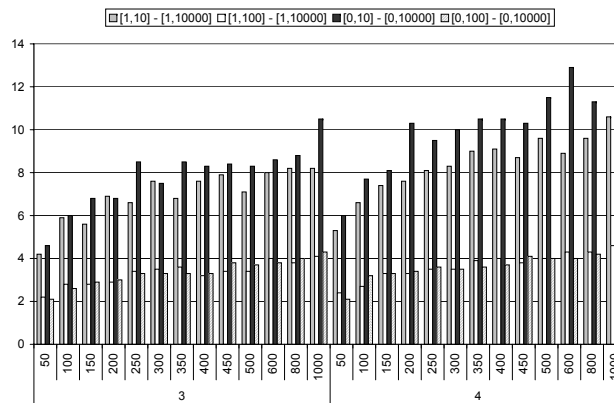
Figura 3.22: Tempo médio de CPU por par de nós (incluindo redes com soluções sub-ótimas) para $m = 3n$ e $m = 4n$.

seu efeito é apesar de tudo limitado nas redes $\bar{\mathcal{E}}$.

Finalmente, é possível observar que os conjuntos gerados por estas redes são em geral de dimensão reduzida (como visível na Figura 3.23a, para todas as gamas de custos $\bar{\mathcal{E}}$ a dimensão é muitas vezes unitária, e a sua média não ultrapassa 1,3). No entanto, existem pares para os quais essa dimensão é significativamente superior (como visível na Figura 3.23b, que apresenta a média das dimensões máximas dos conjuntos obtidos, a qual é mais afectada pela diferença entre as dimensões das gamas do que por essa gama ter ou não a possibilidade de custos nulos). Tal pode ser justificado pelas gamas mais estreitas de custos facilitarem a obtenção de valores de custos iguais em ramos, e portanto a existência de mais possibilidades de atingir o destino com o mesmo custo por caminhos distintos.



(a) Média



(b) Média dos máximos

Figura 3.23: Dimensão dos conjuntos óptimos para gamas \mathcal{E} para $m = 3n$ e $m = 4n$.

3.4 Conclusões

Neste capítulo foram apresentadas algumas melhorias que podem ser aplicadas a esquemas de recuperação que usem partilha de LB de protecção.

Foi apresentado um algoritmo de encaminhamento *online* para protecção local, que é mais eficiente do que o publicado em Kodialam e Lakshman (2001). Os desempenhos da versão original e da versão melhorada do algoritmo, foram analisado com tráfego incremental e tráfego dinâmico. Os resultados mostram que a nova versão usa menos LB e menor número de ramos no CA que a versão original, e que o número de pedidos rejeitados foi significativamente menor na nova versão quando se considerou tráfego dinâmico. O melhor desempenho do algoritmo resulta do uso de uma melhor estimativa da LBP adicional necessária pelos CR (sem aumentar a sobrecarga do protocolo de sinalização), a utilização de reserva exacta da LBP e de algumas modificações na heurística publicada em

Kodialam e Lakshman (2001).

Foi também desenvolvida uma adaptação de um método para reserva de largura de banda de protecção mínima, na protecção local com partilha de LB, proposto originalmente no contexto da protecção global, cuja complexidade espacial foi avaliada.

Finalmente, foi descrito um algoritmo para encontrar o par de caminhos disjuntos, de custo agregado mínimo, numa rede com custos duais nos arcos. Os resultados das experiências mostram que o algoritmo encontra soluções exactas em praticamente todos os casos das experiências com redes dirigidas. Para as soluções sub-óptimas é possível obter um limiar superior e um limiar inferior para o custo óptimo. O tempo de CPU médio para obter uma solução óptima por par de nós, para redes de densidade baixa, foi de poucos milissegundos (com pequenas excepções onde foram necessárias dezenas de milissegundos). Para redes de 10% e 20% de densidade o algoritmo obteve soluções óptimas em tempos de CPU muito menor que a resolução como programação linear usando CPLEX. Este algoritmo foi também adaptado para obter o conjunto de pares de caminhos disjuntos.

4 Um Novo Sistema de Recuperação

Na Secção 4.1 é apresentada uma metodologia de recuperação multi-esquema para redes MPLS, que pretende aumentar a sua resiliência, e ao mesmo tempo usar os seus recursos de forma eficiente tendo em consideração as necessidades das diferentes classes de serviço. O objectivo da metodologia é oferecer protecção a um conjunto de serviços, escolhendo o esquema de recuperação mais adequado, tomando em consideração: classe de serviço, o estado da rede, e as características dos esquemas de recuperação disponíveis. Portanto, o esquema de recuperação apropriado será escolhido com base numa combinação de medidas quantitativas e classificações qualitativas. Na Secção 4.2 é descrito um exemplo de aplicação do sistema proposto e na Secção 4.3 são apresentadas algumas conclusões.

Este capítulo baseia-se no trabalho publicado em Jorge e Gomes (2006a).

4.1 Estrutura do sistema

O sistema de recuperação escolherá de entre um conjunto de esquemas de recuperação o mais apropriado a cada situação particular. O sistema poderá ser implementado num conjunto de LSR da rede, e como tal deverá ter necessidades modestas ou ser capaz de escalar de acordo com as capacidades disponíveis nesses LSR. Portanto, o sistema não será monolítico, mas em vez disso será construído por uma série de componentes que podem ser expandidos, por exemplo incorporando esquemas adicionais ou características a ser incorporadas.

O objectivo da metodologia será oferecer protecção a um conjunto de serviços através da escolha do esquema de recuperação mais apropriado. Esta escolha será feita tendo em consideração a classe de serviço (a qual geralmente poderá requerer um dado nível de protecção), o estado da rede e as características dos esquemas de recuperação considerados ou disponíveis. Isto resultará numa abordagem de recuperação pré-planeada para cada pedido, a qual poderá ser revista periodicamente, porque o comportamento da rede mudou ou por questões de optimização do desempenho.

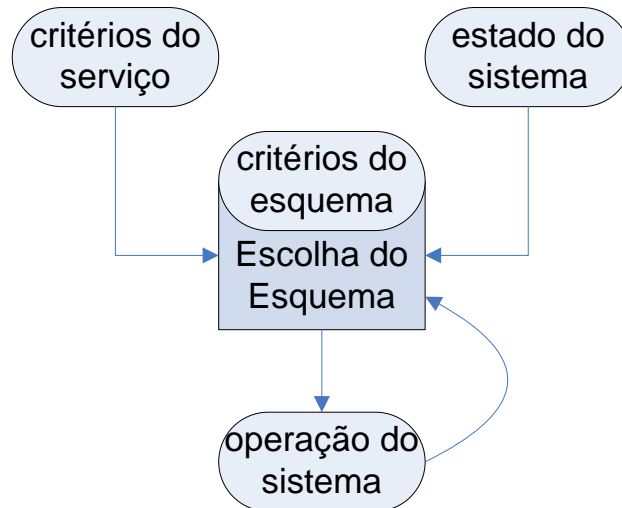


Figura 4.1: Diagrama principal.

Um esquema de recuperação será escolhido baseado numa combinação de medidas quantitativas e classificações qualitativas. Exemplos de medidas quantitativas são a probabilidade de falha de um ramo e a utilização observada de recursos, incluindo partilha de LB. Outras medidas quantitativas, tais como o tempo de recuperação, também poderão ser usadas. As classificações qualitativas usadas podem ser, por exemplo, se um esquema de recuperação usa protecção por comutação ou reencaminhamento, e o tipo de falhas que recupera.

O estado da rede será caracterizado pela descrição dos fluxos de tráfego transportados localmente, LB disponível em todos os ramos (e possivelmente a LB reservada para caminhos activos e para caminhos de protecção), fiabilidade dos ramos e nós, e a topologia exacta da rede (os nós podem falhar e a capacidade dos ramos pode ser reduzida, também devido a falhas) incluindo a descrição dos SRLG.

A estrutura do sistema proposto é apresentada na Figura 4.1. Nela pode observar-se que, a escolha do esquema de recuperação é função de um grupo de critérios:

Critérios de serviço A classe de serviço, como definida para MPLS em Faucheur *et al.* (2002) e mencionada na secção 2.3.5, pode ser a base para o tratamento diferenciado (tratamento por classe de serviço) do tráfego oferecido.

Estado do sistema Um conjunto de parâmetros que descrevem o estado do sistema no instante da escolha. Alguns dos critérios baseados no estado da rede podem necessitar de cálculo de medidas de desempenho as quais podem ser difíceis de obter e/ou necessitarem de dados volumosos ou de

| Nome | Descrição |
|------|--|
| PG1 | Com partilha de reserva |
| PG2 | Sem reserva |
| PL1 | Reserva, sem partilha <i>interdemand</i> |
| PL2 | Reserva, com partilha <i>interdemand</i> |
| PL4 | Sem reserva |
| RG | Sem reserva |
| RL1 | Com pre-cálculo |
| RL2 | Sem pre-cálculo |
| RL3 | Como RL2, mas algoritmo de escolha de caminhos diferente |

Tabela 4.1: Esquemas de recuperação.

aquisição complexa, acerca do estado da rede. Neste caso o sistema pode ser construído de forma a usar apenas a informação disponível, e um subconjunto de critérios possível baseados nas medidas de desempenho que possam ser obtidas, as quais podem depender das capacidades e carga de processamentos dos LSR no instante do cálculo.

Critérios do esquema Os esquemas disponíveis são descritos em função das suas características globais - tais como as apresentadas na secção 2.4 (e em Jorge e Gomes, 2006b). Algumas escolhas preliminares podem focar apenas nas características gerais, com a escolha do esquema exacto feita em etapas posteriores, respeitando (ou não) as escolhas tomadas previamente.

Operação do sistema Depois de ter sido escolhido o esquema de recuperação, mudanças no estado da rede (por exemplo, falhas posteriores) ou nos objectivos de desempenho podem pedir a sua alteração, ou pelo menos a sua revisão.

Todos os elementos anteriores influenciam a escolha do esquema a ser usado para proteger um serviço num LSP particular. Os detalhes da escolha podem ser operacionalizados por um conjunto de tabelas de decisão, que codificam as preferências de decisão de acordo com os critérios anteriores.

É assumido que o conjunto de esquemas de recuperação que pode ser usado é conhecido à partida, embora haja uma certa liberdade na escolha do esquema de recuperação que será usado. Na Tabela 4.1 apresentamos um conjunto de 9 esquemas de recuperação que podem ser usados no sistema. As siglas utilizadas nesta tabela, [P,R][G,L]x, têm o seguinte significado: P – protecção, R – Reencaminhamento, G – Global, L – Local, x – uma implementação particular de um esquema de recuperação. Por exemplo, PL2 representa um esquema de protecção

local com partilha *interdemand*. A Tabela 4.1 pretende apenas dar uma ideia da variedade possível de esquemas de recuperação que poderiam ser considerados.

Muitos dos esquemas apresentados diferem em características observadas externamente. No entanto, alguns esquemas possuem características similares, diferindo apenas nos mecanismos internos, como é o caso dos esquemas RL2 e RL3 apresentados na Tabela 4.1. Nesta situação, a escolha entre eles exige um conhecimento aprofundado de como as características internas se traduzem em desempenho efectivo na rede. De facto, poderá ser mais fácil escolher simplesmente um deles para utilização continuada.

Repare também que a reserva indicada nas características dos esquemas refere-se à LB reservada para os caminhos de recuperação, quando é estabelecido o caminho activo (poderia ser referida por pré-reserva). Alguns esquemas de recuperação apresentados podem não ser apropriados para uso DS-TE, uma vez que, por exemplo usar protecção sem reserva (o que significa que se efectua apenas a sinalização do caminho) pode ser inadequado se as propriedades da classe requerem que seja efectuada reserva de LB antes da sua utilização.

O mecanismo de escolha do esquema consistirá em várias etapas. Estas etapas serão aplicadas sequencialmente embora algumas etapas possam ser saltadas (se assim for desejado), de acordo com o esquema de recuperação seleccionado nas etapas anteriores. Todas as etapas conterão um conjunto de tabelas de decisão. O número de etapas é pré-definido, mas o conteúdo das tabelas não é, nem o número de tabelas a ser usado, embora algum cuidado deva ser tomado para não criar um conjunto incoerente de tabelas.

Em cada tabela, é apresentado um conjunto de condições. As tabelas são de dupla entrada com as condições das classes de serviço representadas nas colunas e com as restantes condições representadas nas linhas. Quando todas as condições se verificarem dizemos que ocorreu um *match* (mesmo que ao *match* não corresponda nenhuma acção, sendo este caso referido por *null match*). Numa tabela será seleccionada uma acção sempre que o critério (serviço, esquema ou estado do sistema) *match* as condições nela apresentadas. A acção será normalmente a selecção do esquema a ser usado, embora algumas vezes seja simplesmente a selecção de algumas características ou parâmetros adicionais, os quais conduzirão eventualmente ao esquema final. Qualquer *match* (incluindo *null matches*) termina a pesquisa na tabela corrente, e por omissão permitirá a pesquisa da próxima tabela na mesma etapa. Contudo, uma acção pode também indicar qual a etapa que se seguirá, portanto permitindo saltos para a etapa seguinte, ou para qualquer outra das próximas etapas. No fim de cada etapa poderá existir uma tabela por "omissão", a qual será usada nas situações seguintes:

- i) sempre que falhar o *match* na tabela anterior da mesma etapa;
- ii) sempre que a escolha anterior possa ser modificada (esta situação ocorre quando não for efectuado nenhum salto para uma etapa posterior, depois da acção na tabela anterior desta etapa);
- iii) e finalmente para fornecer valores por omissão.

É desejado que, em instalações básicas do multi-esquema proposto, a utilização apenas da tabela por omissão ofereça um serviço aceitável.

A sobreposição de escolhas anteriores pode surgir de condições em conflito. Nesse caso, a escolha final será a última efectuada, o que leva a que se deva tomar atenção à ordem pela qual são colocadas as tabelas (um exemplo será apresentado posteriormente). Podem ser usados outros métodos mais elaborados para tratar das condições em conflito, desde que resolvam o conflito dentro de um tempo aceitável.

O sistema de recuperação possui pelo menos uma etapa e no máximo cinco etapas:

- A primeira etapa (A), será realizada quando o serviço surgir, podendo resultar na selecção do modelo de recuperação, a selecção inicial do esquema de recuperação e algumas vezes a selecção do esquema exacto;
- A segunda etapa (B), caso exista, será efectuada imediatamente a seguir à etapa A, culmina no esquema exacto a ser activado para o pedido em causa;
- A terceira etapa (C), será efectuada imediatamente a seguir à etapa B (ou à etapa A caso a etapa B não exista), e permite em função do(s) caminho(s) obtidos com base no esquema de recuperação previamente seleccionado poder escolher um segundo esquema de recuperação para ser utilizado em parte específicas do CA (este segundo esquema de recuperação assume o CA determinado à partida);
- A quarta etapa (F), será avaliada quando ocorre uma falha, tentará usar a informação relativa às características da falha para, se necessário, melhorar ou alterar o esquema seleccionado;
- A quinta etapa (O), ocorrerá algures depois da ocorrência da falha, ou possivelmente quando escalonada, e permite optimização dos caminhos usados pelo pedido, se tal for desejado.

| Classe | Descrição |
|--------|--|
| BE | <i>Best effort</i> |
| AF21 | Tolerância alta de <i>jitter</i> ; Tempos de recuperação médios/altos |
| AF11 | Tolerância média de <i>jitter</i> ; Tempos de recuperação pequenos |
| EF | Tolerância baixa de <i>jitter</i> ; Tempos de recuperação muito exigentes |

Tabela 4.2: Classes de serviço suportadas.

| Condição | BE | AF21 | AF11 | EF |
|----------------------|--------|------|-------|--------|
| <i>Carga Leve</i> | RL1[B] | | PL[B] | PL2[F] |
| <i>Carga Elevada</i> | G[] | | | |

Tabela 4.3: #A.10 – De acordo com o tráfego na rede.

A divisão da escolha inicial de esquema em duas (ou três) etapas permite maior flexibilidade na construção das tabelas que definem a escolha, permitindo inclusivamente estruturar as tabelas de forma a que um grupo destas seja considerado/testado apenas perante determinadas condições (o que pode ser útil para o tratamento de excepções).

4.2 Um exemplo de aplicação

Vai utilizar-se um exemplo para ilustrar como este sistema multi-esquema poderá ser usado numa aplicação suportando quatro classes de serviço (Tabela 4.2).

Cada etapa apresentará algumas tabelas. Por exemplo, para a primeira etapa considerar-se-á a Tabela 4.3 (#A.10), a Tabela 4.4 (#A.20) e a Tabela 4.5 (#A.99). As tabelas apresentadas são apenas ilustrativas das possíveis interacções entre elas. No Capítulo 6, Secção 6.5.5 encontra-se a descrição de uma aplicação do modelo proposto em duas redes.

Estas tabelas serão, como dito anteriormente, pesquisadas sequencialmente. Note-se que se um SLA pré-definido tiver de ser suportado, outras tabelas iniciais, correspondentes às condições desse tráfego e tratamento desejado, podem ser colocadas antes de #A.10 (Tabela 4.3), de modo que o acordo seja cumprido.

Sempre que exista um *match*, são tomadas as acções definidas pelo *match*, que podem incluir um salto para uma etapa posterior (o destino do salto é indicado entre parêntesis rectos, []). Na Tabela 4.3 se a carga de tráfego for leve, para AF11 ocorre um salto para a etapa B depois de ter sido seleccionado usar recuperação

| <i>LB</i> | <i>Condições</i> <i>Comprimento do LSP</i> | BE | AF21 | AF11 | EF |
|----------------------|---|-----------|-------------|-------------|-----------|
| <i>Garantida</i> | <i>Curto</i> | | PG1[F] | | |
| | <i>Longo</i> | | PL2[F] | PL1[F] | |
| <i>Não garantida</i> | <i>Curto</i> | | | | |
| | <i>Longo</i> | L[] | | | |

Tabela 4.4: #A.20 – De acordo com a garantia de LB e comprimento do LSP activo mínimo.

| <i>Condição</i> | BE | AF21 | AF11 | EF |
|-----------------|-----------|-------------|-------------|-----------|
| <i>Sempre</i> | R[] | R[] | P[] | P[] |

Tabela 4.5: #A.99 - Regras aplicadas por omissão para a etapa A.

por protecção local, por outro lado se a classe de serviço for EF, será seleccionado o esquema de recuperação PL2 e ocorrerá um salto para a etapa F, o que significa que para este tráfego em particular só serão feitas configurações adicionais quando houver uma falha no caminho activo. Frequentemente, quando ocorre um *match* não será seleccionado nenhum salto (como é mostrado na Tabela 4.3, para o tipo de serviço BE em redes com carga elevada), o que significa que o sistema prosseguirá para a tabela seguinte, tendo em consideração as acções seleccionadas (neste exemplo, o modelo de recuperação Global será seleccionado para esse tráfego, embora esta escolha possa sofrer alterações posteriores).

Sempre que uma acção for seleccionada como consequência de um *match*, esta nova acção sobreporá as escolhas prévias. Esta sobreposição poderá ser apenas parcial, se a acção apenas descrever alguma característica do esquema, em vez do esquema completo – neste caso, apenas a característica seleccionada reflecte a nova acção, e será combinada com as escolhas prévias. Na Tabela 4.4 uma sobreposição poderá ocorrer para BE, quando o caminho activo for longo escolhendo neste caso recuperação Local (assumindo que tinha sido seleccionada previamente recuperação Global para essa classe). O comprimento do LSP utilizado nessa tabela pode obter-se como sendo o comprimento do LSP mais curto possível para o pedido, uma vez que o LSP activo para o pedido que será utilizado efectivamente, só será determinado quando o esquema de recuperação for completamente conhecido, pois em alguns esquemas de recuperação, como descrito no Capítulo 3, os LSP activo e de recuperação são calculados conjuntamente.

Na Tabela 4.4, repare também que são escolhidos dois esquemas semelhantes mas diferentes para AF21 e AF11 quando o tamanho do LSP for longo e for exigida LB garantida, uma vez que diferenças nos esquemas podem fazer PL1 um pouco mais rápido que PL2.

| <i>Condições</i> | | BE | AF21 | AF11 | EF |
|------------------|-------------------|-----------|-------------|-------------|-----------|
| <i>R/P</i> | <i>L/G</i> | | | | |
| <i>P</i> | <i>L</i> | PL4[] | PL1[] | PL1[] | PL1[] |
| | <i>G</i> | PG2[] | PG2[] | PG1[] | PG1[] |
| | <i>indefinido</i> | PG2[] | PG2[] | PL1[] | PL1[] |
| <i>R</i> | <i>L</i> | RL2[] | RL2[] | RL1[] | RL1[] |
| | <i>G</i> | RG[] | RG[] | RG[] | RG[] |
| | <i>indefinido</i> | RG[] | RG[] | RL1[] | RL1[] |

Tabela 4.6: #B.99 - Tabela por omissão para a etapa B.

Uma vez terminada a pesquisa na tabela por omissão de uma etapa, o sistema pode seguir para a primeira tabela da próxima etapa se as condições para essa etapa forem satisfeitas. Por exemplo, depois da Tabela 4.5, o sistema pode prosseguir imediatamente para a Tabela 4.6. Contudo, transições para a etapa F requerem a detecção de uma falha no caminho, e para a etapa O que alguma condição – por exemplo um temporizador interno – indique para isso ser feito. Note-se que embora não seja exigida a existência de uma tabela por omissão é aconselhável a definição dessa tabela pelo menos para a etapa B, de forma a garantir que a todos os pedidos seja atribuído um esquema de recuperação.

Algumas condições, nas tabelas propostas, serão apenas verdadeiras se os valores testados forem indefinidos – até esse instante – por exemplo na Tabela 4.6 algumas acções apenas poderão vir a ser activadas se não tiver sido tomada nenhuma decisão prévia relativa ao modelo de recuperação (Local ou Global) utilizado.

Poderão ser colocadas outras tabelas antes da Tabela 4.6 (#B.99). A Tabela 4.6, em combinação com a Tabela 4.5, deverá fornecer um esquema de recuperação por omissão para todos os pedidos para os quais não tenha ocorrido um *match* em qualquer uma das tabela especificada até então. Depois desta etapa, estará supostamente seleccionado um esquema de recuperação para cada pedido. Na etapa C este esquema de recuperação poderá vir a ser complementado por um segundo esquema de recuperação (a ser aplicado em alguns ou em todos os ramos do CA) se o CA obtido pela aplicação do primeiro esquema de recuperação não cumprir determinadas condições.

A Tabela 4.7 mostra que as condições podem ser bastante complexas. Nesta tabela PFP1 será verdadeiro se a probabilidade de falha do CA for maior que algum valor pré-determinado, LFP1 será activado sempre que a probabilidade de falha de um ramo do CA perto do fim for superior que alguma outra constante, e LFP2 será verificado se a probabilidade de falha de um ramo, para algum

| <i>Condições</i> | | | BE | AF21 | AF11 | EF |
|------------------|------------|-----------------------|-----------|----------------|----------------|----------------|
| <i>P/R</i> | <i>L/G</i> | <i>Prob. de falha</i> | | | | |
| <i>P</i> | <i>G</i> | <i>PPF1</i> | | +PL4[F]{todos} | +PL4[F]{todos} | +PL1[F]{todos} |
| | | <i>LFP1</i> | | | | +PL1[]{LFP1} |
| | | <i>LFP2</i> | | +PL4[]{LFP2} | +PL4[]{LFP2} | +PL1[]{LFP2} |
| <i>R</i> | <i>G</i> | <i>PPF1</i> | | +RL2[F]{todos} | +RL2[F]{todos} | +RL1[F]{todos} |
| | | <i>LFP1</i> | | | | +RL1[]{LFP1} |
| | | <i>LFP2</i> | | +RL2[]{LFP2} | +RL2[]{LFP2} | +RL1[]{LFP2} |

Tabela 4.7: #C.10 - De acordo com a probabilidade de falha do CA.

| <i>Condições</i> | | | BE | AF21 | AF11 | EF |
|------------------|------------|--------------------|-----------|----------------|----------------|----------------|
| <i>P/R</i> | <i>L/G</i> | <i>Comp. do CA</i> | | | | |
| <i>P</i> | <i>G</i> | <i>CAP</i> | | +PL2[]{todos} | +PL2[]{todos} | +PL1[]{todos} |
| <i>R</i> | <i>G</i> | <i>CAP</i> | | +RL2[]{todos} | +RL2[]{todos} | +RL1[]{todos} |

Tabela 4.8: #C.99 - De acordo com o comprimento do CA.

ramo no CA for maior que duas vezes a probabilidade média de falha do ramo considerando todos os ramos do CA. Na Tabela 4.8 CAP será verdadeiro se o comprimento do CA for maior que duas vezes o comprimento médio dos CA. Como será esperado, alguns destes testes poderão ser computacionalmente dispendiosos em termos da quantidade de dados necessários ou do tempo que será necessário despendar para os adquirir, e como tal a sua utilização deverá ser ponderada com os benefícios esperados da sua utilização.

Nas Tabelas 4.7 e 4.8 a acção seleccionada será muitas vezes indicada por “+Esquema de recuperação[etapa]{condição}”. Isto indica que serão seleccionados dois esquemas para esses pedidos. O primeiro esquema de recuperação (que resultou no final da etapa B) será aplicado a todo o caminho activo, excepto nos ramos “excepcionais” que necessitarão de um tratamento diferente. O segundo esquema de recuperação local (que será apresentado nas acções destas tabelas depois do sinal “+”) será aplicado apenas em alguns dos ramos do CA, portanto criando a necessidade de um esquema que designamos por esquema híbrido. De notar que nem todos os pedidos recebem novos esquemas nesta etapa. Os ramos que precisem de tratamento “excepcional” serão seleccionados através da informação da condição que devem verificar (condição colocada entre chavetas nas acções dessas tabelas). Esta notação significa que existem duas condições: a primeira, aplicada ao pedido, indica se um segundo esquema deve ser considerado; a segunda, colocada entre chavetas, é aplicada a cada ramo do CA testa se para esse ramo em particular deve ser aplicado o segundo esquema. Na notação usada se a condição a aplicar aos ramos for “todos” isso significa que todos os ramos do

| <i>Local</i> | <i>Condições</i> | | BE | AF21 | AF11 | EF |
|--------------|------------------|-----------------|-----------|-------------|-------------|-----------|
| | <i>R/P</i> | <i>L/G</i> | | | | |
| <i>La1</i> | <i>R</i> | <i>Qualquer</i> | RG[] | RG[] | RG[] | RG[] |
| <i>La2</i> | <i>R</i> | <i>Qualquer</i> | RG[] | RG[] | RG[] | RL3[] |
| <i>La3</i> | <i>R</i> | <i>Qualquer</i> | RG[] | RL2[] | RL3[] | RL3[] |

Tabela 4.9: #F.10 - Mudanças no esquema de recuperação por reencaminhamento de acordo com a localização da falha.

| <i>Local</i> | <i>Condições</i> | | BE | AF21 | AF11 | EF |
|--------------|------------------|-----------------|-----------|-------------|-------------|-----------|
| | <i>R/P</i> | <i>L/G</i> | | | | |
| <i>Lr1</i> | <i>R</i> | <i>L</i> | RG[O] | RL2[O] | RL3[O] | RL1[O] |
| | <i>R</i> | <i>G</i> | RG[O] | RG[O] | RL3[O] | RL1[O] |
| | <i>P</i> | <i>Qualquer</i> | RG[O] | RL2[O] | RL1[O] | RL1[O] |

Tabela 4.10: #F.99 - Tabela por omissão para a etapa F.

CA desse pedido usarão para recuperação o segundo esquema. O segundo esquema de recuperação não deve determinar os caminhos de forma agregada uma vez que ele será utilizado apenas na determinação dos caminhos de recuperação. Repare que se algum dos dois esquemas de recuperação escolhidos determinar os caminhos de forma agregada o resultado obtido poderá ser pouco eficiente em termos de consumo de LB, uma vez que poderão existir recursos reservados em duplicado.

Quando ocorrer uma falha, a etapa F será usada e poderá conduzir a uma modificação do esquema de recuperação escolhido previamente para os LSP afectados. Na Tabela 4.9 a localização da falha no caminho activo é codificada. La1 refere-se a falhas em caminhos activos curtos, La2 refere-se a falhas próximas do início em caminhos activos longos e La3 refere-se a falha próximas do fim em caminhos activos longos. Como poderá ser visto, a localização da falha pode influenciar o tipo de reencaminhamento a ser tentado.

Na Tabela 4.10 (tabela por omissão para a etapa F) uma falha localizada no caminho de recuperação é codificada como Lr1. A pesquisa nesta tabela será apenas desencadeada se falhas múltiplas simultâneas (ou falhas sequenciais muito próximas) causarem também a falha do caminho de recuperação antes desta tabela ser testada. Esta tabela pode então ser usada para seleccionar um esquema de recuperação adicional no caso do esquema de recuperação anterior não conseguir tratar da situação. Pode ser encontrado um exemplo na Tabela 4.10 onde para (Lr1, R, G) e AF11 um esquema de recuperação por reencaminhamento local será a solução obtida.

Não é apresentada nenhuma tabela para a etapa O, mas essas tabelas podem

ser construídas como as anteriores. Serão usadas para oferecer uma oportunidade para decidir quando e como otimizar o “novo” caminho activo e o “novo” caminho de recuperação ou talvez seleccionar caminhos de recuperação depois de uma falha, de acordo com a classe de serviço. Falhas sequenciais múltiplas, que não são geralmente consideradas pela etapa F também poderiam ser resolvidas nesta etapa.

4.3 Conclusões

Foi proposta uma metodologia de recuperação multi-esquema, para aumentar a resiliência da rede, e ao mesmo tempo utilizar os recursos da rede de forma eficiente tomando em consideração as diferentes necessidades das várias classes de serviço. A metodologia tenta oferecer protecção a um conjunto de serviços através da escolha do esquema de recuperação mais adequado, tomando em consideração a classe de serviço, o estado da rede, e as características dos vários esquemas de recuperação considerados. Essa escolha do esquema de recuperação foi mecanizada através de um conjunto de etapas cada uma das quais consistindo em várias tabelas de decisão. Foram descritos os detalhes do sistema que incorpora essa metodologia.

A implementação deste multi-esquema necessitará de aferir o conteúdo e número de tabelas. Esta proposta vai ser sujeita a análise no Capítulo 6, onde se ilustra o seu funcionamento. Essa análise será feita recorrendo a um estudo simulacional do sistema.

5 Aspectos da Implementação

Na primeira secção deste capítulo é feita uma descrição das características mais importantes do ambiente de simulação utilizado, necessárias para a definição do modelo que representa o sistema a simular. Seguidamente, na Secção 5.2, é feita uma descrição geral do funcionamento do programa, seguida, na Secção 5.3, de uma descrição do funcionamento da troca de mensagens (e máquina de estados) que suporta o tratamento de cada pedido de estabelecimento de ligação e a simulação de falhas. Depois disso, na Secção 5.4, é explicada a implementação do programa de simulação, sem entrar em demasiados pormenores de implementação (esses pormenores podem ser obtidos através da consulta do Apêndice A).

5.1 Breve descrição do ambiente de simulação utilizado – OMNeT++

De acordo com a filosofia de modelação que o *Object-oriented Modular discrete event NeTwork simulator* (OMNeT++) implementa, um modelo/rede consiste em várias entidades designadas por módulos embutidos hierarquicamente. O módulo no nível de topo é designado por módulo sistema e conterá normalmente submódulos. Os módulos que possuem submódulos são designados por módulos compostos e os módulos que estão no nível mais baixo da hierarquia são designados módulos simples, sendo estes os módulos activos. Não existindo limite para a profundidade dos módulos, é permitido ao utilizador reflectir a estrutura lógica do sistema na estrutura do modelo.

O código dos módulos simples é escrito em linguagem C++ usando a biblioteca de classes do OMNeT++. Os módulos simples são criados por derivação da classe `C_SIMPLEMODULE` da biblioteca de classes do OMNeT++. Os módulos comunicam entre si pela passagem de mensagens as quais podem conter estruturas de dados complexas. Os módulos podem enviar mensagens, directamente para os destinos ou por um caminho pré-definido, através de portas e ligações. As portas são os interfaces de entrada e saída dos módulos, sendo as mensagens enviadas através de uma porta de saída e recebidas através de uma porta de entrada. Podem ser criadas ligações entre portas correspondentes de dois

submódulos, ou entre a porta de um submódulo e a porta correspondente do módulo composto. Os módulos podem ter parâmetros, que podem ser usados para construir topologias flexíveis para o modelo, para personalizar o comportamento dos módulos e para comunicação entre os módulos através de variáveis partilhadas. A estrutura do modelo é definida em linguagem *NEtwork Descrip-tion* (NED), manualmente num ficheiro de texto ou através da utilização do editor gráfico fornecido, *Graphical NEtwork EDitor* (GNED), resultando em ambos os casos ficheiros NED. Nos ficheiros NED podem existir declarações de módulos, que correspondem simplesmente à definição dos tipos dos módulos, e a definição da rede, que declara o modelo da simulação como instâncias dos tipos de módulos previamente definidos. No ficheiro ou ficheiros NED utilizados pela aplicação podem ser definidos vários modelos/redes, o que irá ser executado é aquele que for indicado no ficheiro *ini* (este ficheiro será descrito posteriormente).

Para criar um programa de simulação executável é necessário ligar o código produzido (código C++ correspondente à implementação dos módulos simples), ao núcleo (*kernel*) do OMNeT++, à biblioteca de classes auxiliares do OMNeT++ e a um dos interfaces com o utilizador oferecidos pelo OMNeT++ (e, nas versões anteriores do OMNeT++, ao código C++ correspondente aos ficheiros NED). Na versão actual do OMNeT++ os ficheiros NED, embora possam ser ligados com o restante código da aplicação normalmente não o são, em vez disso são *preloaded* para memória (ou seja, compilados para memória no arranque).

É necessário definir um ficheiro de configuração associado a cada aplicação, designado ficheiro *ini* (normalmente com o nome "*omnetpp.ini*"). A lista (ou listas) de ficheiros NED que são *preloaded* para memória é aquela que tiver sido colocada no ficheiro de configuração. Desses ficheiros NED são apenas inicializados/instanciados aqueles que forem utilizados na rede/modelo. As opções que controlam como são executadas as várias corridas de simulação (e os valores para os parâmetros do modelo) são também especificadas no ficheiro *ini*, e podem ser especificadas globalmente ou individualmente para cada corrida. Nesse ficheiro podem ser seleccionadas as sementes para os geradores de números aleatórios, deve ser especificado o modelo/rede da simulação a ser usado, podem ser seleccionados os nomes dos ficheiros de saída dos resultados, etc. Actualmente o OMNeT++ usa por omissão o gerador de números pseudo-aleatórios Mersenne Twister RNG de Matsumoto e Nishimura (1998), que possui um período de $2^{19937} - 1$.

5.2 Descrição geral do funcionamento do programa

Nesta secção são descritos os aspectos externos do programa de simulação, ou seja os aspectos do programa de simulação específicos do ambiente de simulação usado.

5.2.1 Descrição da topologia e ficheiros NED

O comportamento do modelo foi implementado em C++. Todo o código desenvolvido foi encapsulado no módulo simples principal. Este código inclui a redefinição de três funções membro virtuais, invocadas pelo *kernel* da simulação. Uma função invocada no início da simulação, contém as diversas inicializações, incluindo a obtenção de vários parâmetros do ficheiro de configuração e a criação de mensagens iniciais. Outra função invocada imediatamente antes da simulação terminar, contém código que faz a consolidação e registo das estatísticas obtidas ao longo da simulação. A função restante é a função principal, a qual é invocada várias vezes ao longo do programa de simulação, sempre que é necessário efectuar o processamento de um acontecimento/mensagem.

No modelo desenvolvido não foi definido apenas um módulo simples, mas sim vários, para permitir que o simulador possa ser utilizado no estudo de outros esquemas de recuperação que não envolvam a utilização de tabelas, e para facilitar a extensão do simulador à simulação ao pacote. No entanto, pensando utilizar o simulador apenas para o estudo específico do novo sistema de recuperação com simulação ao fluxo, essa simplificação poderia existir.

Na implementação presente do modelo é definido um ficheiro NED com as características da rede que se queira utilizar. Actualmente existe um ficheiro NED para a rede COST239 e um para a rede KL-15 (ambas as redes serão apresentadas no Capítulo 6). Nesses ficheiros existe um módulo simples por cada nó, LSR, da rede e são definidas as ligações correspondentes às ligações reais entre eles, o que em conjunto corresponde à declaração de um módulo composto. De uma forma simplificada ir-se-á referir apenas mais outro ficheiro NED, aquele em que foi declarado o modelo da simulação. Este modelo consiste na instanciação de um módulo composto definido previamente nesse ficheiro. O módulo composto consiste em dois submódulos.

Um dos submódulos funcionará como um ponto centralizado de informação e de cálculo, o submódulo que corresponde ao principal módulo simples definido (na realidade existe a possibilidade de escolher entre dois módulos simples, o que está a ser utilizado na simulação do novo esquema de recuperação e outro que

permite utilizar o simulador na simulação de esquemas de recuperação simples; os módulos partilham grande parte do código). Pensa-se que muito do código deste módulo se poderá manter numa simulação ao pacote, mas a quantidade dependerá efectivamente da especificidade da implementação.

O outro submódulo é uma instância do módulo composto que contém a definição da rede (aqui será escolhida a declaração do módulo que corresponde à rede que se deseja simular, de entre as várias que podem ter sido declaradas).

5.2.2 Descrição da saídas e ficheiros que as suportam

De nada serviria a aplicação se não permitisse a obtenção de resultados. Os resultados são registados nos dois tipos de ficheiros permitidos pelo OMNeT++, ficheiros de escalares para o registo de valores individuais, geralmente no final da simulação e ficheiros de vectores para o registo de séries temporais, geralmente produzidas ao longo de toda a simulação. As Tabelas 5.3 e 5.1 ilustram as várias saídas da simulação e as respectivas descrições, respectivamente no ficheiro escalar e vectorial.

Para cada pedido gerado é registada essa ocorrência (NUMPEDIDOS), o CT a que pertence (CTRAFEGO), a sua duração prevista (DPEDIDO) e a sua duração real (DEFECPEDIDO), a sua LB (LPEDIDO) e a indicação de se o pedido foi estabelecido (NPEDESTABL) ou recusado (NPEDRECUSAD)¹. Se o pedido foi estabelecido com sucesso é também registada a sua LB (lbPedidoEst), o comprimento do CA (DIMCAMACT), a LB necessária para o CA (LBCAMACT)² (obtida através do produto dos dois últimos valores) e a LB necessária no caminho de comprimento mínimo (LBCAMACTMINIMO)³. Para os pedidos em que é determinado um (ou vários) CR é registada essa situação (NPEDCRP), a LB necessária para os CR (LBCAMBAK)⁴ (calculada como a soma da LB dos ramos usados no/nos CR do pedido, sem incluir a LB já fornecida por partilha), o número de CR (NCAMINHOSBACKUP)⁵, o número de ramos distintos no/nos CR (NRAMOSBACKUP)⁶ e o número de ramos do CR com mais ramos (NRAMOSCAMMAXBAK). É registada a ocorrência do fim de um pedido (NPEDTERM), qualquer que seja a razão, e também do término com sucesso (NPEDTERMSUC) caso tenha sido essa a situação. Durante o tempo em que o pedido está a ser servido pode ser afectado por uma falha, se tal acontecer,

¹Um pedido será recusado se não for possível determinar o CA. Nos esquemas de protecção pode também ser recusado se não for possível determinar o/os CR.

²Pode ser ou não reservada e pode estar ou não ocupada.

³Caminho entre a origem e o destino do pedido, obtido pelo algoritmo de *Dijkstra*.

⁴Nos esquemas de reencaminhamento só será diferente de zero em situação de falha.

⁵Nos esquemas de âmbito global será 1.

⁶Os ramos do CA usados pelo/s CR não são contados.

| Descrição | Nome |
|--|-----------------|
| Pedido tentado | NUMPEDIDOS |
| Número da CT do pedido | CTRAFEGO |
| Duração prevista do pedido | DPEDIDO |
| Duração real do pedido | DEFECPEDIDO |
| Largura de banda do pedido | LPEDIDO |
| Pedido estabelecido | NPEDESTABL |
| Pedido recusado | NPEDRECUSAD |
| Largura de banda do pedido estabelecido | LPEDIDOEST |
| Comprimento do CA do pedido | DIMCAMACT |
| LB do CA do pedido | LCAMACT |
| LB do caminho (CA) de comprimento mínimo | LCAMACTMINIMO |
| CR calculado | NPEDCRP |
| LB para os CR do pedido | LCAMBAK |
| Número de CR do pedido | NCAMINHOSBACKUP |
| Número de ramos do/s CR do pedido | NRAMOSBACKUP |
| Número de ramos do maior CR do pedido | NRAMOSCAMMAXBAK |
| Pedido terminado | NPEDTERM |
| Pedido terminado com sucesso | NPEDTERMSUC |
| Pedido afectado por falha | NFALHA |
| Pedido recuperado após falha | NRFALHA |
| Comprimento do LSP usado em falha | DIMCAMUSAFAI |
| Pedido terminado por falha | NTFALHA |

Tabela 5.1: Saídas do programa de simulação - vectoriais.

para esse pedido é também registada essa ocorrência (NFALHA). Se o pedido foi recuperado de uma falha será também registada essa indicação (NRFALHA) e o comprimento do caminho⁷ usado na recuperação da falha (DIMCAMUSAFAI), caso contrário será registada a indicação de que o pedido terminou como consequência da falha (NTFALHA). Um pedido terá uma duração inferior à prevista se for afectado por uma falha da qual não foi possível ser recuperado.

Alguns dos valores registados conduzem ao registo de informação que embora redundante se considerou adequado fazer, para validação.

Uma vez que se pode considerar que o sistema a modelar é *non terminating*, a escolha da altura em que se termina a simulação não é trivial (Law e Kelton, 1991). O simulador construído permite escolher arbitrariamente o “tempo de aquecimento” e “tempo de término”. Uma vez que indicar esses tempos usando uma medida pré-definida poderia ser difícil para simulações com condições diferentes (redes diferentes) foi proposto usar o número de pedidos em vez de uma

⁷Na recuperação local o caminho usado na falha pode incluir ramos do caminho activo.

medida de tempo para este objectivo. Esta escolha permite criar e comparar simulações nas quais ocorreu a mesma quantidade de um determinado tipo de “acontecimentos”, independentemente do tempo simulado. No simulador usado não é necessário especificar a unidade de tempo usada. No entanto, se se pretender, pode-se considerar a unidade de tempo usada o minuto (o que leva a valores relativamente realísticos de duração de pedidos).

As intensidades de falhas estudadas são muitas vezes superiores às expectáveis numa rede real. Esta decisão foi tomada uma vez que se pretende estudar o comportamento do sistema perante falhas, e não apenas em funcionamento normal, e usar os MTBF usuais implicaria quer tempos de simulação muito maiores quer efeitos muito menos visíveis das falhas no resultado do sistema.

| Nome (saídas do ficheiro de vectores) | Medida a obter |
|--|--|
| NUMPEDIDOS | Número de pedidos tentados |
| DPEDIDO | Duração prevista dos pedidos |
| NPEDESTABL | Número de pedidos estabelecidos |
| NPEDTERMSUC | Número de pedidos terminados com sucesso |
| DEFECPEDIDO | Duração real dos pedidos |
| LPEDIDO | LB dos pedidos |
| NPEDRECUSAD | Número de pedidos (de ligação) recusados |
| NFALHA | Número de pedidos afectados por falhas |
| NRFALHA | Número de pedidos recuperados após falha |
| NTFALHA | Número de pedidos terminados por falha |
| LCAMACT | LB de todos os CA |
| DIMCAMACT | Comprimento dos CA |
| LCAMBAK | LB dos CR |
| DIMCAMUSAFA | Comprimento dos CR usados na falha |
| CTRAFEGO | Número de pedidos de cada CT |
| LCAMACTMINIMO | LB dos caminhos (CA) de comprimento mínimo |
| NCAMINHOSBACKUP | Número total de CR |
| LPEDIDOEST | LB dos pedidos efectivamente estabelecidos |
| NPEDCRP | Número de pedidos em que foi calculado um CR |

Tabela 5.2: Possíveis medidas obtidas a partir do ficheiro de vectores.

A Tabela 5.2 apresenta algumas medidas simples (cálculo de somas) que podem ser obtidas a partir dos valores registados no ficheiro de vectores (apresentados na Tabela 5.1). Em geral, as medidas podem ser obtidas para um agregado de todos os CT e também para cada um dos CT individualmente. Para identificar estas medidas vamos usar o nome usado no ficheiro de saídas seguido de $_c$ para designar a soma de todas as ocorrências desse valor e seguido de $_cCT_x$ para designar a soma de todas as ocorrências desse valor para o CT_i , com $i \in I_T$, onde I_T é o conjunto definido pelos índices de todos os CT no sistema. Por exemplo, para

o número total de pedidos gerados usaremos a designação $NUMPEDIDOS_c$, e por exemplo para o número total de pedidos gerados do CT3 usaremos a designação $NUMPEDIDOS_cCT3$.

A partir das medidas simples indicadas na Tabela 5.2, é possível criar um conjunto de medidas compostas, sendo algumas destas as realmente usadas na análise. De entre as medidas compostas pode-se realçar:

- O bloqueio no estabelecimento, definido como a probabilidade de rejeição (bloqueio) de pedidos (de ligação) aquando do estabelecimento do pedido, obtido através de:

$$\frac{NPEDRECUSAD_c}{NUMPEDIDOS_c} \quad (5.1)$$

- O bloqueio no estabelecimento para o CTx , onde $x \in I_T$:

$$\frac{NPEDRECUSAD_cCTx}{NUMPEDIDOS_cCTx} \quad (5.2)$$

- A probabilidade de desconexão na falha, definida como a probabilidade de desconexão na sequência de uma falha da qual não foi possível recuperar por reencaminhamento, obtida por:

$$\frac{NTFALHA_c}{NFALHA_c} \quad (5.3)$$

- A probabilidade de desconexão na falha para o CTx , onde $x \in I_T$:

$$\frac{NTFALHA_cCTx}{NFALHA_cCTx} \quad (5.4)$$

- A duração média dos pedidos estabelecidos, obtida por:

$$\frac{DEFECPEDIDO_c}{NPEDÉSTABL_c} = \frac{\sum_{i=1}^{NPEDÉSTABL_c} DEFECPEDIDO_i}{NPEDÉSTABL_c} \quad (5.5)$$

- A LB média dos pedidos estabelecidos:

$$\frac{LBPEDIDOEST_c}{NPEDÉSTABL_c} = \frac{\sum_{i=1}^{NPEDÉSTABL_c} LBPEDIDOEST_i}{NPEDÉSTABL_c} \quad (5.6)$$

- A LB média dos CA:

$$\frac{LBCAMACT_c}{NPEDESTABL_c} = \frac{\sum_{i=1}^{NPEDESTABL_c} LBCAMACT_i}{NPEDESTABL_c} \quad (5.7)$$

- A LB média dos caminhos (potenciais CA) de comprimento mínimo:

$$\frac{LBCAMACTMINIMO_c}{NPEDESTABL_c} = \frac{\sum_{i=1}^{NPEDESTABL_c} LBCAMACTMINIMO_i}{NPEDESTABL_c} \quad (5.8)$$

- O comprimento médio (também sendo possível obter o comprimento máximo e mínimo) dos CA:

$$\frac{DIMCAMACT_c}{NPEDESTABL_c} = \frac{\sum_{i=1}^{NPEDESTABL_c} DIMCAMACT_i}{NPEDESTABL_c} \quad (5.9)$$

- O comprimento médio dos CA para o CT x , onde $x \in I_T$:

$$\frac{DIMCAMACT_cCTx}{NPEDESTABL_cCTx} \quad (5.10)$$

- A LB média para o/s CR de um CA:

$$\frac{LBCAMBAK_c}{NPEDCRP_c} = \frac{\sum_{i=1}^{NPEDCRP_c} LBCAMBAK_i}{NPEDCRP_c} \quad (5.11)$$

- O número de ramos distintos médio (também sendo possível obter o número máximo e o número mínimo) no/nos CR dos CA:

$$\frac{NRAMOSBACKUP_c}{NPEDCRP_c} = \frac{\sum_{i=1}^{NPEDCRP_c} NRAMOSBACKUP_i}{NPEDCRP_c} \quad (5.12)$$

- O número médio de CR por CA:

$$\frac{NCAMINHOSBACKUP_c}{NPEDCRP_c} = \frac{\sum_{i=1}^{NPEDCRP_c} NCAMINHOSBACKUP_i}{NPEDCRP_c} \quad (5.13)$$

- O número médio de CR por CA para o CT x , onde $x \in I_T$:

$$\frac{NCAMINHOSBACKUP_cCTx}{NPEDCRP_c} \quad (5.14)$$

A Tabela 5.3 apresenta as medidas mais significativas registadas no ficheiro de escalares no instante imediatamente antes da simulação terminar. As medidas

| | Medida | Nome do campo |
|-------------------------|--|----------------------|
| <i>Totais</i> | Nº de falhas | FALHASTOTALS |
| | Duração das falhas (média, min., max.) | DURFALHASTOTALS |
| <i>Por Ramo</i> | Nº de falhas que afectam pedidos | FALHASRAMO |
| | Duração das falhas (média, min., max.) | DURFALHASRAMO |
| <i>Por CT</i> | Nº de pedidos afectados por falhas | PEDAFFECTFALHATOTALS |
| | Nº de pedidos terminados por falhas | PEDTERMFALHATOTALS |
| | LB reservada para CA | LBAPTOTALS |
| | LB reservada para CR | LB RPTOTALS |
| | LB ocupada sem reserva para CA | LB OCAPTOTALS |
| | LB ocupada sem reserva para CR | LB OCRPTOTALS |
| <i>Por CT e Ramo</i> | Nº de pedidos afectados por falhas | PEDAFFECTFALHARAMO |
| | Nº de pedidos terminados por falhas | PEDTERMFALHARAMO |
| | LB reservada para CA | LB APRAMO |
| | LB reservada para CR | LB RPRAMO |
| | LB ocupada | LB OCUPADARAMO |
| <i>Por CT e Esquema</i> | Nº de pedidos estabelecidos | ESQESTABECIDO |
| | Nº de pedidos tentados | ESQTENTADO |

Tabela 5.3: Saídas do programa de simulação - escalares.

globais registadas são o número total de falhas que ocorreram durante toda a corrida de simulação e a sua duração (mínima, média e máxima por corrida). Estes mesmos valores são também registados individualmente para cada um dos ramos (com a limitação de apenas serem contabilizadas as falhas que afectam pedidos nesse ramo).

Para cada CT são contabilizados o número de pedidos afectados por falha e o número de pedidos que uma falha levou a ser desconectados (neste caso incluem-se aqueles que não foi possível recuperar e aqueles afectados por uma falha enquanto estão a usar o caminho de recuperação – este último caso só é possível para esquemas que não possam implementar reversão após termo da falha, pois não se consideram falhas simultâneas). Estas mesmas medidas são também contabilizadas por ramo e CT.

Para cada CT também são contabilizadas as LB médias reservadas para CA e CR (estas mesmas medidas são também contabilizadas por ramo e CT). Para além disto, e para suportar esquemas que não utilizem reserva são também contabilizadas as LB médias ocupadas/usadas sem reserva para CA e CR (para manter um registo da utilização de LB dos pedidos sem reserva). Estas últimas medidas não estão disponíveis directamente por ramo e CT. Ao invés, é contabilizado por ramo e CT a LB média efectivamente usada nesse ramo (ou seja, a LB ocupada sem reserva somada com a LB efectivamente ocupada nesse ramo para tráfego

corrente que tenha sido reservada para CA ou para CR). Como indicado, o simulador distingue a LB reservada e a LB ocupada. A primeira corresponde àquela que é atribuída a pedidos que o solicitem (em CT que usem reserva), a segunda corresponde à LB efectivamente usada num determinado momento para passar tráfego dos pedidos. Isto significa que pode existir LB reservada mas não ocupada, e LB ocupada sem reserva (em CT que não usem reserva).

Finalmente, para determinar as quantidades de pedidos tratados por cada esquema de recuperação, é contabilizado por esquema e CT o número de pedidos se tentou estabelecer com determinado esquema, e quantos destes realmente conseguiram ser estabelecidos.

Deve notar-se que sempre que é indicado na Tabela 5.3 que os valores são obtidos por CT, existe também uma saída adicional correspondente ao total de todos os CT para essas saídas (contabilizado pelo sistema simultaneamente com as medidas por CT).

A partir das medidas obtidas directamente do ficheiro de escalares (descritas na Tabela 5.3), podem ser calculadas algumas agregações, nomeadamente para obter a LB gasta (por ramo e por CT ou simplesmente por ramo). São apresentadas duas medidas diferentes porque a forma de agregação por ramo usada não facilita a agregação por ramo e CT, e também para permitir análise das diferenças entre elas. Nestas expressões vamos redefinir a notação apresentada anteriormente de que Z_{CTx} representa a medida Z para CTx.

A primeira destas expressões trata de forma diferente os CT que exigem reserva (cujos índices vamos considerar que definem o conjunto I_R) daqueles que não usam reservas (cujos índices definem o conjunto $I_{\bar{R}}$). Assim, para calcular a LB gasta por ramo foi usada a seguinte expressão (equivalente à soma das reservas para os CT que exigem reserva com a soma das ocupações para o restante CT):

$$\sum_{x \in I_R} \text{LBAPRAMO}_{CTx} + \sum_{x \in I_{\bar{R}}} \text{LBRPRAMO}_{CTx} + \sum_{y \in I_{\bar{R}}} \text{LBOCUPADARAMO}_{CTy} \quad (5.15)$$

Já para calcular a LB gasta por ramo e CT foi usada a seguinte expressão (correspondente a uma aproximação, que soma à LB usada nesse ramo a LB reservada nesse ramo para os CR – embora esta medida seja uma aproximação porque não considera correctamente o tráfego que esteja nesse momento desviado devido a falha, obtém resultados bastante próximos da medida anterior no caso de se considerar a soma de todos os CT), definida para o CTx, onde $x \in I_T$:

| Nome | Descrição |
|--------------------|--|
| NUMMAXPEDIDOS | Número de pedidos gerados |
| NPEDIDOINICIALSTAT | Número de pedidos desprezado |
| N_CLASSES_TRAF | Número de CT |
| MAM_LIM_CLASSES | Proporções de tráfego (e de LB) para o MAM |
| MAR_FACTOR_CLASSES | Factores multiplicativos |
| MAR_LIMIAR | Limiar para o MAR |
| LAMBDA_POISSON | Intensidade dos pedidos |
| MIU_POISSON | Taxa de terminação dos pedidos |
| LAMBDA_FALHAS | Intensidade das falhas |
| MIU_FALHAS | Taxa de terminação das falhas |
| BW | LB dos pedidos |

Tabela 5.4: Parâmetros de entrada comuns a todas as corridas.

$$LBRPRAMO_{CTx} + LBOCUPADARAMO_{CTx} \quad (5.16)$$

Embora ambos estes valores sejam na simulação obtidos por ramo, na análise dos resultados é apresentada a média aritmética dos resultados de todos os ramos (para todas as corridas consideradas).

5.2.3 Descrição dos ficheiros de configuração

De seguida é apresentado o conteúdo do ficheiro de configuração (ficheiro *ini*). A dimensão deste ficheiro é considerável uma vez que contém uma secção por cada uma das corridas de simulação. Para evitar a construção fastidiosa deste ficheiro ele é construído automaticamente por um programa, desenvolvido em Perl, que toma como entrada um ficheiro em que são indicados de uma forma compacta os nomes dos parâmetros de entrada da simulação e as gamas de valores respectivos. O programa gera automaticamente todas as combinações possíveis de parâmetros nas gamas indicadas, sendo cada combinação gerada associada a uma corrida (*run*).

A Tabela 5.4 apresenta os parâmetros de entrada comuns a todas as corridas de simulação. Além dos parâmetros apresentados nessa tabela existem alguns mais, menos relevantes, que servem basicamente para indicar quais das opções da implementação devem ser utilizadas na execução. A regra (simple) de paragem do programa de simulação é a seguinte: cada replicação corre até ser gerado um determinado número de pedidos pré-fixado, NUMMAXPEDIDOS. Desses, um número inicial (tempo de transição) é desprezado, NPEDIDOINICIALSTAT. Só passado esse

| Nome | Descrição |
|--------------------|--|
| FACTOR_TRAFEGO | Factor multiplicativo para a geração de pedidos |
| TIPO_TABELAS | Número que indica o conjunto de tabelas a utilizar |
| OUTPUT-SCALAR-FILE | Nome do ficheiro de saída de escalares |
| OUTPUT-VECTOR-FILE | Nome do ficheiro de saída vectorial |

Tabela 5.5: Parâmetros de entrada específicos de cada corrida.

número de pedidos é que começa a ser feita a recolha de dados, para reduzir o efeito dos transitórios iniciais nas medidas de desempenho calculadas. O número de CT que se pretende nos casos em estudo é introduzido no parâmetro `N_CLASSES_TRAF`.

Foram implementados dois BCM para ser aplicados aos CT, o MAM e o MAR. O parâmetro `MAM_LIM_CLASSES` é utilizado simultaneamente para indicar as proporções de tráfego e de LB para os vários CT, a serem utilizadas no MAM. Uma vez que não se considerou utilizar os dois modelos simultaneamente, reutilizamos no MAR o parâmetro `MAM_LIM_CLASSES` para indicar as proporções de tráfego para os vários CT. Para o MAR é necessário indicar ainda o limiar de reserva de LB (parâmetro `MAR_LIMIAR`). No MAR as proporções de LB a serem utilizadas para os vários CT são obtidas multiplicando as proporções de tráfego por determinados factores. O parâmetro `MAR_FACTOR_CLASSES` contém esses factores. Segundo o RFC 4126 (Ash, 2005) o factor para CT0 deve ser zero, para CT de prioridade normal deve ser um e para CT de elevada prioridade dois ou três são valores típicos.

Tanto os pedidos como as falhas estão a ser gerados como processos de Poisson. O tempo entre dois pedidos consecutivos e a duração dos pedidos, bem como o tempo entre duas falhas consecutivas e duração das mesmas, são considerados com distribuição exponencial negativa. Os valores das durações médias (tempos de serviço médio) dos pedidos e das falhas, são obtidos respectivamente dos parâmetros `MIU_POISSON` e `MIU_FALHAS` (os tempos são o inverso dos parâmetros `MIU` correspondentes). As intensidade (frequência) de chegada dos pedidos e das falhas são indicadas respectivamente nos parâmetros `LAMBDA_POISSON` e `LAMBDA_FALHAS`. A LB dos pedidos gerados é um valor inteiro entre 1 e o valor do parâmetro `BW`.

Outros parâmetros que também fazem parte do ficheiro de configuração (parâmetros específicos de cada corrida), são apresentados na Tabela 5.5. O parâmetro `FACTOR_TRAFEGO` é utilizado para permitir estudos sob diversas condições de carga da rede, sendo usado como factor multiplicativo da frequência de gera-

ção de pedidos (`LAMBDA_POISSON`). O tempo simulado será tanto maior quanto mais pequeno for o valor do `FACTOR_TRAFEGO` uma vez que o número total de pedidos gerados é sempre o mesmo. Para cada corrida é necessário especificar um número que indica qual o conjunto de tabelas a utilizar na corrida de simulação, parâmetro `TIPO_TABELAS`. Fazem ainda parte do ficheiro de configuração os nomes dos ficheiros de saída (de vectores e de escalares) para as várias corridas. Esses nomes são construídos com informação que identifica facilmente o conteúdo dos ficheiros. Dessa informação faz parte a data de obtenção dos resultados, o factor de tráfego utilizado, etc. Para cada um dos casos estudados foram executadas várias replicações independentes.

Foram também desenvolvidas algumas aplicações auxiliares: uma já referida, para criar os ficheiros de configuração; outra foi desenvolvida para controlar a execução distribuída das várias corridas por várias máquinas, que é responsável por fazer um tratamento parcial de alguns dos resultados e por fazer a importação de todos os resultados para uma base de dados.

5.3 Funcionamento da simulação por troca de mensagens

O comportamento do programa de simulação é definido pelo código associado ao módulo simples principal. Sempre que surge um acontecimento a primeira função a ser invocada é a função principal associada directamente a esse módulo, referida na Secção 5.2.1. Esta função tem por objectivo fazer o processamento do acontecimento, determinar o instante do próximo acontecimento e escaloná-lo. No OMNeT++ os acontecimentos são representados por mensagens.

Existem dois geradores principais e independentes de mensagens, um gera pedidos e o outro gera falhas. Tanto os pedidos como as falhas são gerados como processos de Poisson com valores de parâmetros introduzidos no ficheiro de configuração (ficheiro *ini*). Na fase de inicialização da simulação são geradas mensagens iniciais, uma geradora de pedidos e uma geradora de falhas.

Na fila de acontecimentos existe sempre uma mensagem geradora de pedidos e uma mensagem geradora de falhas. Existirão também em geral várias mensagens correspondentes a pedidos, tantas quantos os pedidos activos. Quando existir uma falha activa (no máximo pode existir uma, restrição que traduz o cenário de falha isolada subjacente aos algoritmos com partilha de LB) existirá ainda na fila de acontecimentos uma mensagem de falha. A Figura 5.1 ilustra um estado possível da fila de acontecimentos. Essa figura corresponde a um instante em que existem 6 pedidos no sistema (em estados não especificados na figura). Neste exemplo existe uma falha activa. A mensagem de falha apresentada na figura

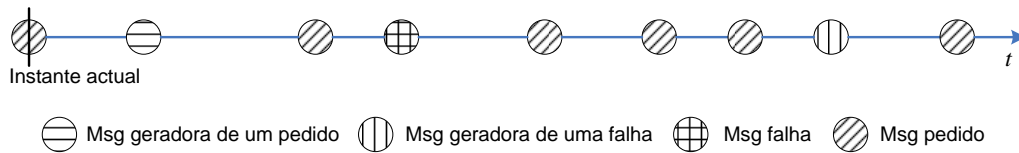


Figura 5.1: Fila de acontecimentos futuros - um estado possível.

sinaliza o fim dessa falha.

Sempre que é invocada a função que faz o tratamento de mensagens é porque aconteceu uma das três situações seguintes:

1. Surgiu uma mensagem geradora (de um pedido ou de uma falha);
2. Surgiu uma mensagem associada a um pedido, que passamos a designar por mensagem pedido;
3. Surgiu uma mensagem associada a uma falha, que passamos a designar por mensagem falha.

Quando surge uma mensagem geradora de uma falha para ser tratada, vai ser gerada efectivamente uma nova instância de falha e escalonada de novo a mensagem geradora de falha para o instante em que deve surgir a falha seguinte. Na geração da nova falha começa por ser determinado o ramo, de entre os vários da rede, que será afectado pela falha. Considerou-se que todos os ramos possuem a mesma probabilidade de falha (utilização de uma distribuição uniforme discreta). Na geração da falha é criada uma mensagem falha, à qual se associa informação relativa à mesma. Esta mensagem é escalonada para o instante actual da simulação. Quando ela for recebida pela função de tratamento de mensagens será verificado que se trata do início de uma falha. Nesta situação várias acções serão realizadas: O ramo afectado pela falha será marcado como desactivado; serão identificados todos os pedidos afectados pela falha (sendo construído um vector com todas as mensagens associadas a esses pedidos) e será alterado o estado desses pedidos para o sub-estado SP_EM_FALHA (como será referido adiante a cada pedido activo está associada uma máquina de estados); todas as mensagens dos pedidos afectados pela falha serão escalonadas para o instante actual da simulação; por último a mensagem falha será escalonada para o instante em que a falha for recuperada.

Comportamento similar vai ser desencadeado quando surge uma mensagem geradora de um pedido para ser tratada. De modo análogo, vai ser gerado uma nova instância de pedido e escalonada a mensagem geradora de um pedido para

o instante em que deve surgir o pedido seguinte. A geração do novo pedido envolve a determinação do nó origem e destino, a determinação da LB e a determinação do CT do pedido. A escolha do nó origem e do nó destino é feita através de uma distribuição uniforme discreta, considerando-se todos os nós com a mesma probabilidade de serem escolhidos (à semelhança do realizado em Calle *et al.*, 2004; Kodialam e Lakshman, 2002b; Qiao e Xu, 2002; Xiong *et al.*, 2003). Também foi utilizada uma distribuição uniforme discreta na determinação da LB (de entre os valores $\{1, 2, \dots, BW.\}$). Actualmente a determinação do CT está a ser efectuada pela aplicação do método de Monte Carlo tendo por base as proporções do BCM utilizado. A geração do novo pedido envolve também a atribuição de uma máquina de estados ao pedido e a criação de uma mensagem pedido (à qual é associada uma estrutura que armazena informação relativa ao pedido). Esta mensagem é escalonada para o instante actual da simulação. Posteriormente, a mensagem será sucessivamente escalonada aquando da saída dos vários estados estacionários da máquina de estados associada ao pedido.

Foi necessário definir uma máquina de estados para controlar as transições entre os vários estados dos pedidos. Cada pedido está sempre num determinado estado. Os estados podem ser estados estacionários ou transitórios. A diferença entre estes é que a transição entre estados transitórios se faz automaticamente sem passagem de tempo (e sem consultar a fila de acontecimentos) enquanto que um pedido num estado estacionário devolve o controlo ao simulador (que consulta a fila de acontecimentos e pode levar a passagem de tempo). Um acontecimento num pedido (mensagem pedido) faz passar o pedido para o próximo estado de acordo com a máquina de estados. O pedido sai do estado estacionário corrente, percorre um ou mais estados transitórios, e finalmente chega a um novo estado estacionário. Cada estado possui código que será executados quando se sai dele e pode também possuir código para ser executado quando se entra no estado. No estado FIM_PEDIDO o pedido é libertado.

A Figura 5.2 apresenta o diagrama de fluxo da máquina de estados. São visíveis na figura os vários estados (21 no total, incluindo estacionários e transitórios) em que um pedido se pode encontrar. Actualmente o único estado em que ocorre passagem de tempo é o estado S_EM_PROGRESSO, embora tenham sido definidos outros estados estacionários (estados em que poderá também ser gasto tempo). Outros estados em que se pretenda considerar passagem de tempo devem ser convertidos de transitórios para estacionários. Em parte a complexidade da máquina de estados resultou de se pretender que fosse adequada não só na simulação ao fluxo mas também na simulação ao pacote (e neste último caso permitir o funcionamento adequado do RSVP).

No início, cada pedido é colocado no estado INICIO_PEDIDO. Seguidamente, passará pelo grupo de estados S_TABELA_A, S_TABELA_B e T_DETERMINA_CAMINHOS para escolha do esquema e com este do CA (e possivelmente do CR). De seguida, passa pelo estado S_TABELA_C, usado para permitir suporte de esquemas híbridos, mas não definido na implementação actual. O pedido de recursos e verificação das reservas é efectuado nos estados T_ESTABREG_CAMINHOS, T_TESTA_TODOS_VERIF e S_VERIF_ESTAB_CAMINHOS. Após isto, o pedido é considerado realmente estabelecido, o que é conseguido nos estados T_EM_PROGRESSO e S_EM_PROGRESSO. Se tudo prosseguir sem problemas o pedido ficará no último estado referido até expirar a sua duração. Após isto, será então mudado para os estados T_REMOVREG_CAMINHOS e FIM_PEDIDO para libertação de recursos usados e fim efectivo do pedido. Se durante a execução do pedido ele for afectado por uma falha, passará para o estado S_TABELA_F (que executa a etapa responsável pela escolha de esquemas na falha). Daqui o pedido pode terminar por falha através do estado S_TERMINOU_EM_FALHA. Pode passar em vez disto por (alguns) estados como T_DETERMINA_RP (escolha de um novo CR), T_PASSA_A_USAR_RP (utiliza na prática o CR) ou T_PREPARA_NOVAS_RESERVAS (efectua as reservas para o novo CR). Existe ainda a possibilidade de passar pelo estado T_REMOVREG_CAMINHOS_NA_FALHA se foi decidida a escolha de um novo CA. Após o término da falha o pedido pode passar aos estados T_PEDIDO_EM_REVERSAO e T_PASSA_A_USAR_AP usados no tratamento de esquemas reversivos. Um outro estado, S_TABELA_O, foi considerado para suportar optimização, mas não é usado na implementação actual. Existem ainda algumas transições entre estados não referidas, por exemplo para suportar a terminação de pedidos que não foi conseguido estabelecer.

A Figura 5.3 ilustra o percurso de vários pedidos, desde o início do pedido - criação, até ao fim do pedido - conclusão, passando por vários estados intermédios. São também apresentadas nessa figura várias mensagens geradoras de pedidos e uma mensagem geradora de uma falha (não são apresentadas as mensagens pedido nem as mensagens falha). Apesar de não ser uma representação à escala, é visível na figura que existe um grande período de passagem de tempo que corresponde ao intervalo em que o pedido está em progresso. Na realidade na implementação actual, como será referido à frente, o estado S_EM_PROGRESSO é mesmo o único em que existe passagem do tempo simulado. Na figura, apenas o Pedido 2 foi afectado pela falha. Este pedido começa por ter um período em progresso, após o que é afectado por uma falha (instante t_4), recupera desta e passa ao segundo período em progresso. Posteriormente, a falha é eliminada (sinalizada pelo aparecimento de uma mensagem (de fim de) falha, em t_6 , não

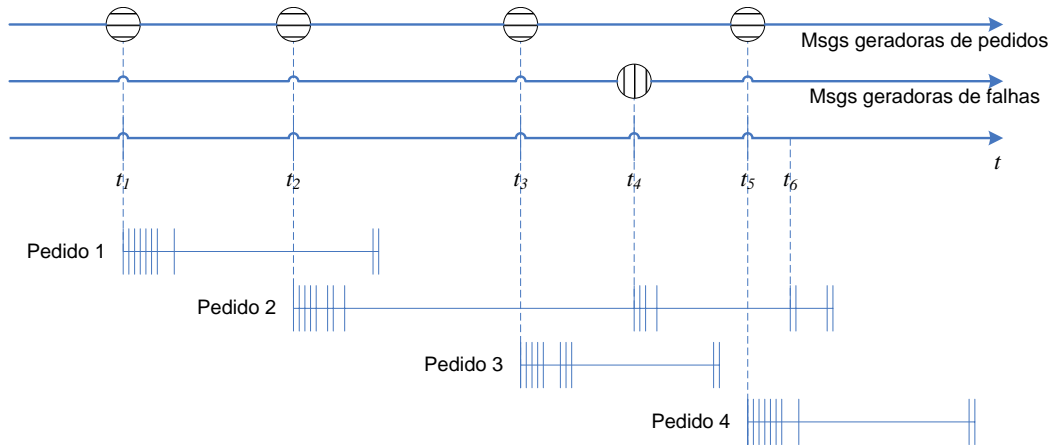


Figura 5.3: O percurso de quatro pedidos.

representada na figura) e o pedido regressa ao caminho original, terceiro período em progresso.

Foi referida nesta secção a utilização da distribuição uniforme discreta na escolha da origem e destino do pedido, da LB do pedido, do ramo afectado por uma falha e a utilização do método de Monte Carlo na determinação do CT. Para utilizar outras distribuições ou outros parâmetros das distribuições, que se considerem mais adequados ao sistema que se pretenda simular, a alteração do código não seria difícil, mas a determinação das distribuições e parâmetros mais adequados à rede em causa envolveria a necessidade de um estudo adicional.

5.4 Projecto do programa de simulação

Nesta secção é descrita a estrutura do programa de simulação. A Figura 5.4 mostra as classes principais que foram desenvolvidas para este programa e algumas das suas relações de herança de propriedades. De notar que a figura não contém a totalidade das classes e muito menos as ligações entre elas. Com a figura pretende-se dar uma visão global da implementação do programa. As classes `cSIMPLEMODULE`, `cMESSAGE`, `cFSM`, `cOUTVECTOR` e `cSTATISTIC` fazem parte da hierarquia de objectos do ambiente de simulação utilizado (OMNeT++), todas as restantes foram desenvolvidas de raiz.

As classes apresentadas na Figura 5.4 podem agrupar-se de acordo com a sua função em seis grupos principais, distinguidos na figura através da diferença de cores. Os seis grupos identificados são:

- Componentes nucleares – As classes que implementam os componentes



Figura 5.4: Diagrama das classes mais significativas definidas na implementação.

principais usados na definição da estrutura do programa de simulação, ilustradas na Figura 5.5.

- Componentes associados aos esquemas de recuperação – As classes que implementam as especificidades de cada esquema de recuperação, descritas na Figura 5.6.
- Componentes específicos do novo sistema de recuperação – As classes necessárias à definição e tratamento das estruturas de informação necessárias para o sistema de recuperação proposto, nomeadamente relativas ao tratamento das tabelas, descritas na Figura 5.7.
- Estruturas para armazenamento de dados – As classes que definem estruturas para armazenamento e funções de manipulação de informação, apresentadas nas Figura 5.8.
- Componentes de recolha de dados – Fornecem as funções de recolha de informação e algumas funções de consolidação de informação, agrupadas na Figura 5.9.

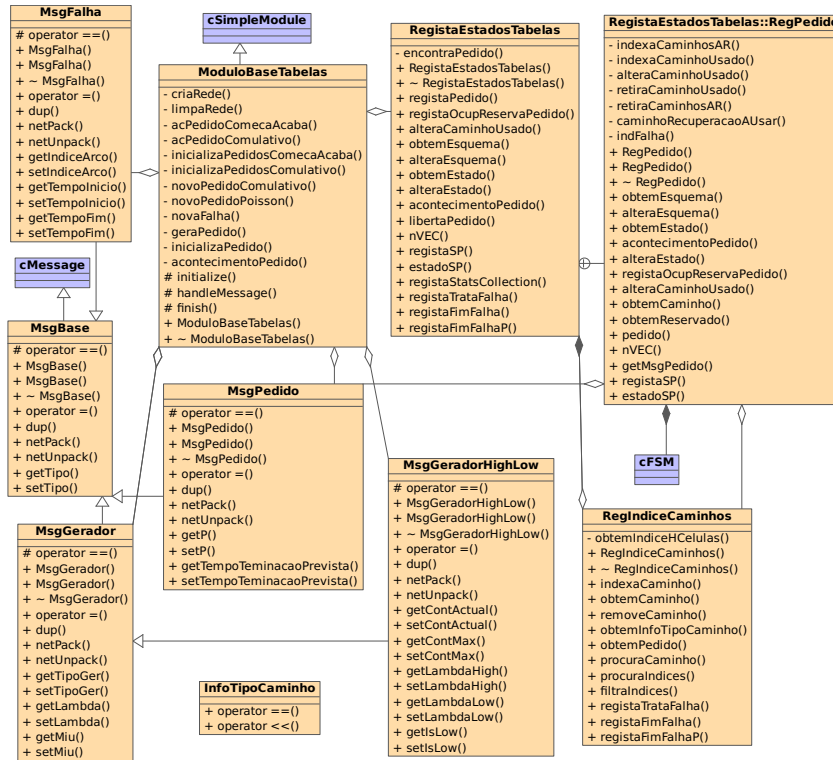


Figura 5.5: Componentes nucleares.

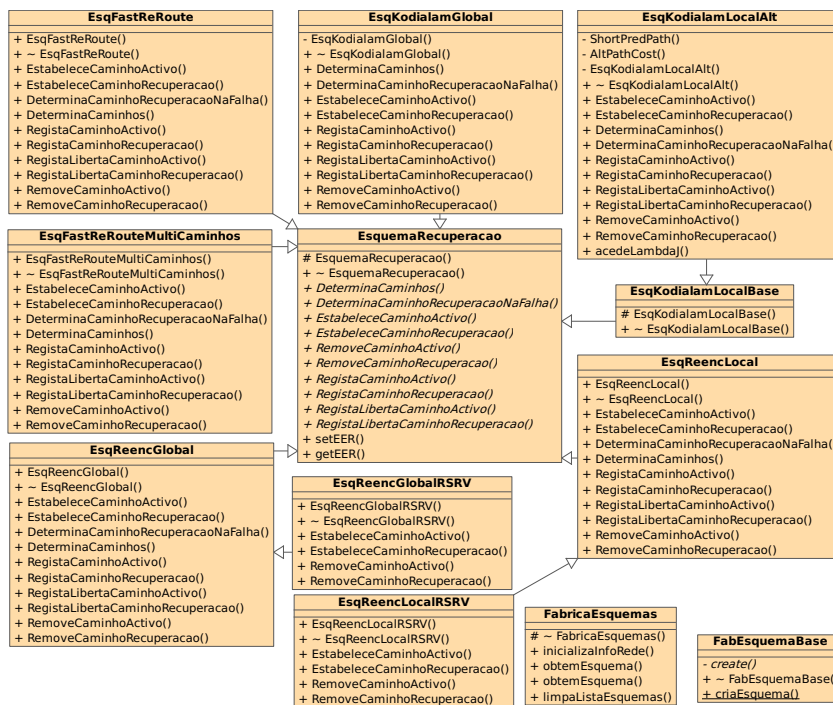


Figura 5.6: Componentes associados aos esquemas de recuperação.

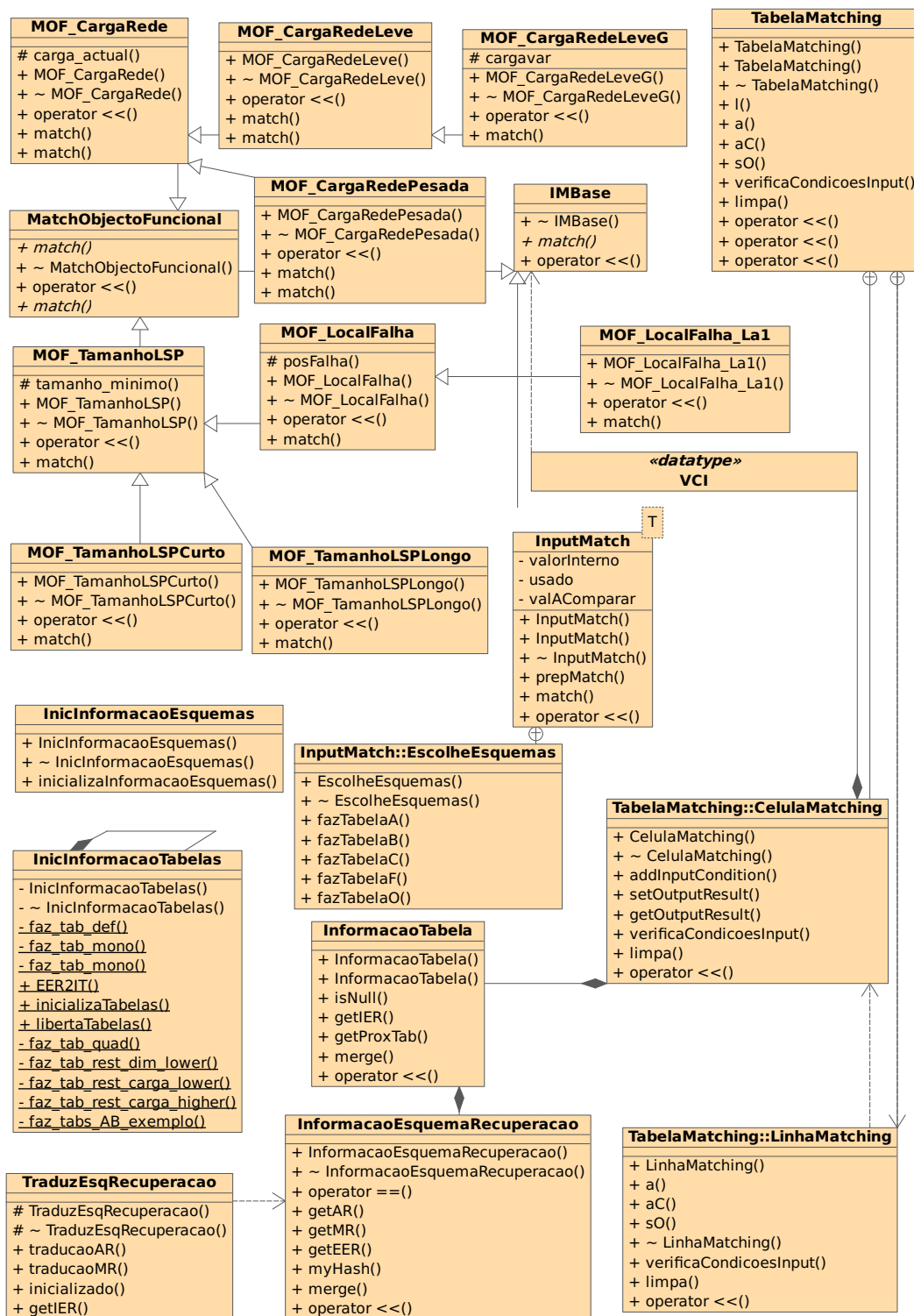


Figura 5.7: Componentes específicos do novo sistema.

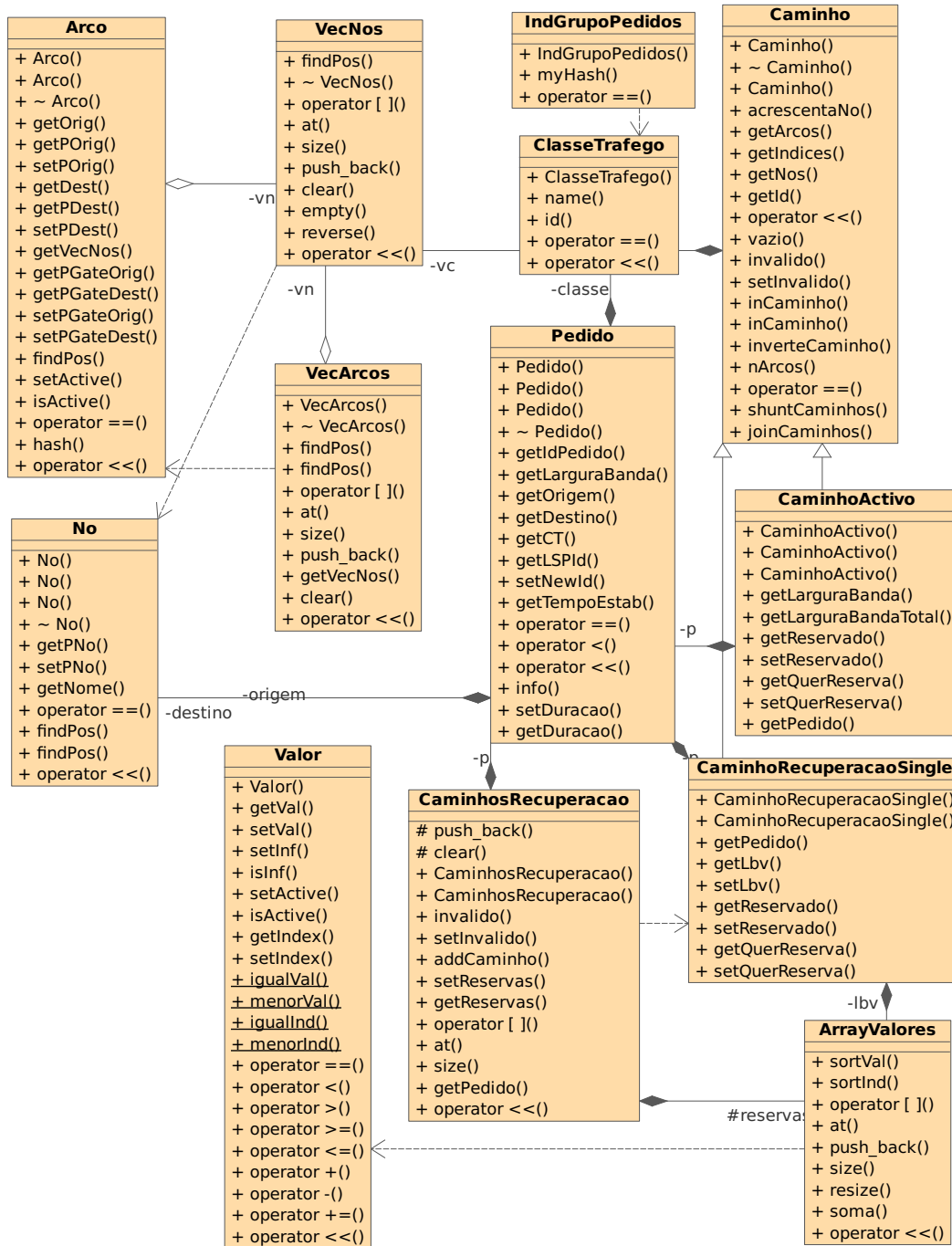


Figura 5.8: Estruturas para armazenamento de dados.

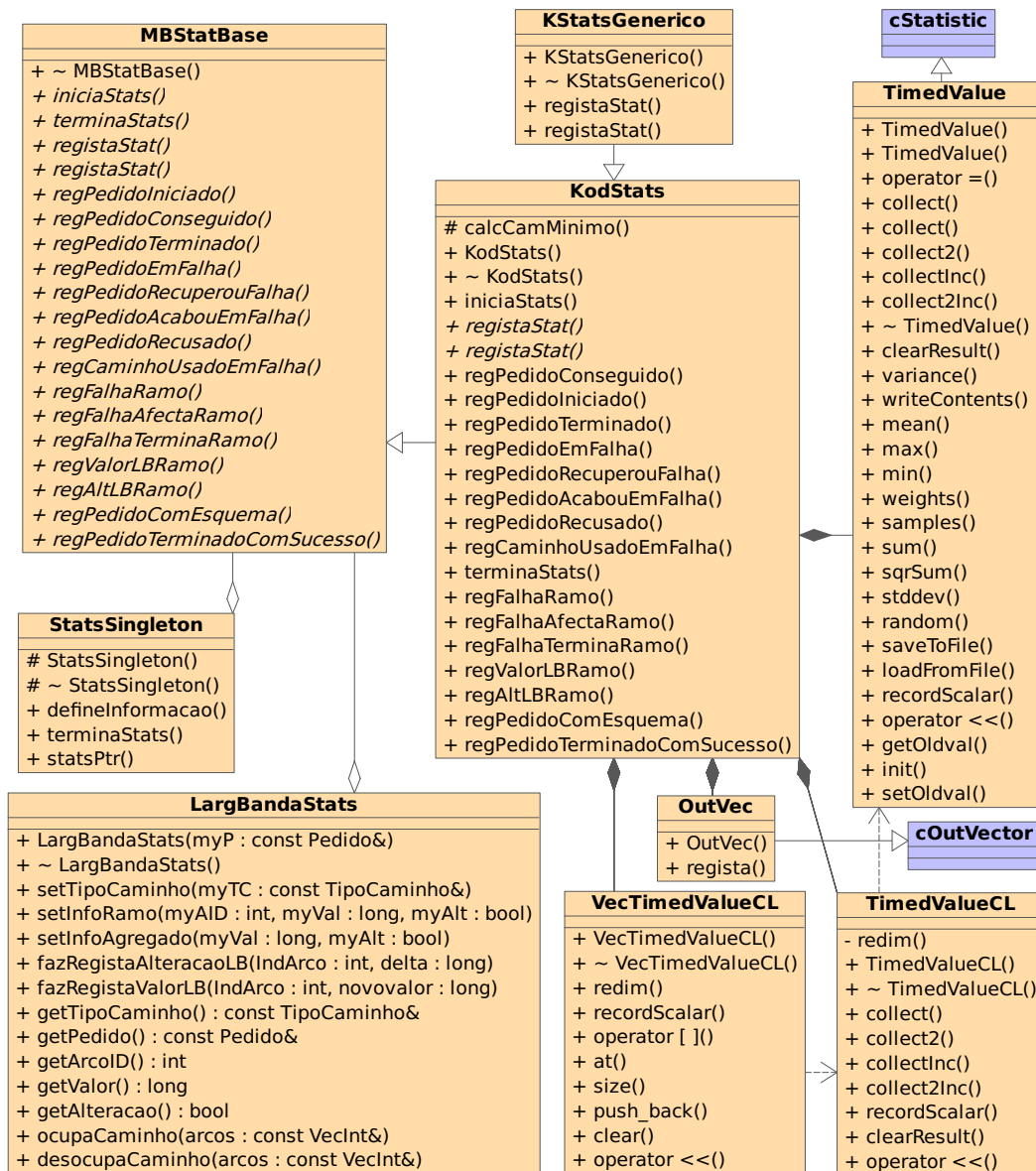


Figura 5.9: Componentes de recolha de dados.

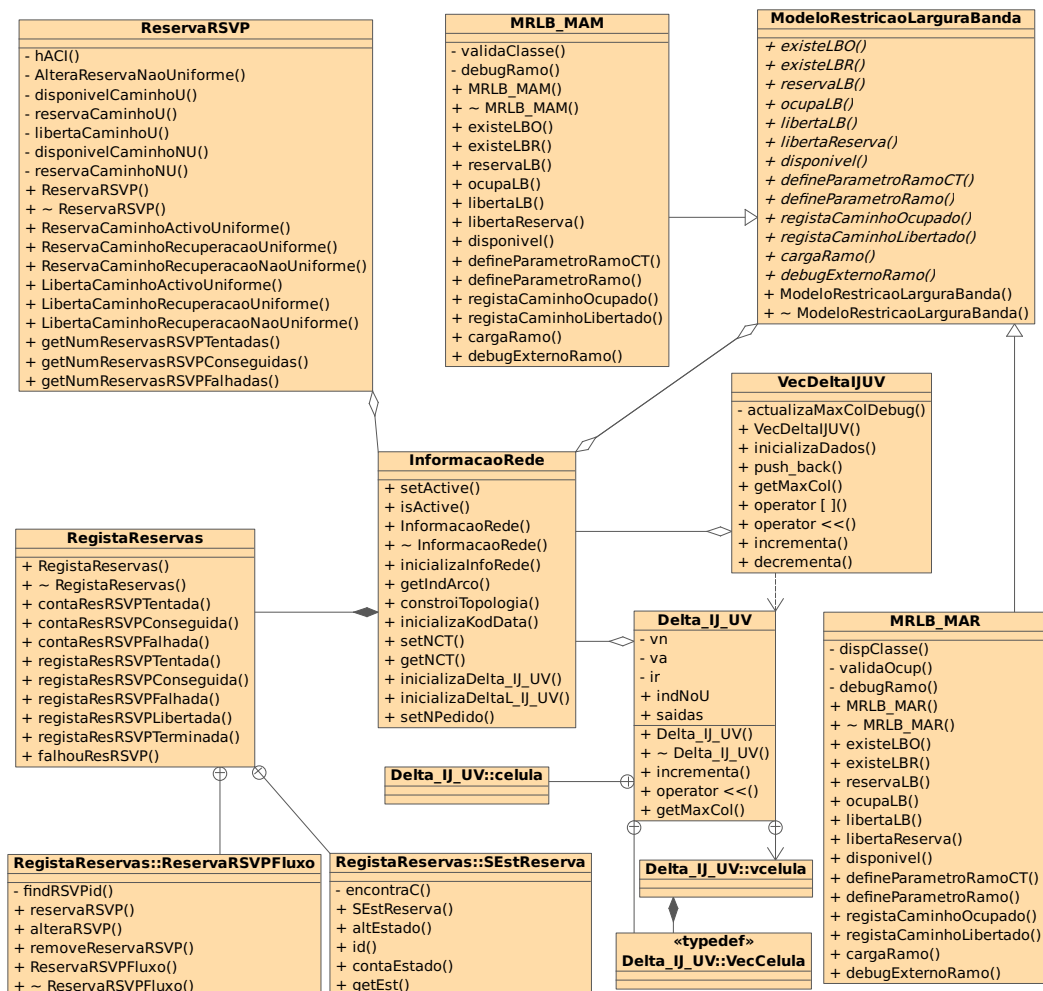


Figura 5.10: Componentes auxiliares.

- Componentes para armazenamento da topologia, reservas e ocupações da rede – As classes que definem estruturas para armazenamento de informação da rede e para controlo de reservas e ocupações dos ramos, agrupadas na Figura 5.10.

Nas figuras cada uma das classes é representada indicando os seus métodos (os campos foram omitidos). A visibilidade das componentes é assinaladas por - para visibilidade privada, por # para visibilidade protegida e por + para visibilidade pública.

Seguidamente ir-se-á fazer uma descrição das várias componentes. Informação pormenorizada acerca dos vários objectos referidos, encontra-se no Apêndice A.

5.4.1 Componentes nucleares (módulo base)

O suporte principal da simulação é implementado pela classe `MODULOBASETABELAS`, que implementa o código C++ correspondente ao módulo `OMNeT++` principal da simulação. Este código tem mecanismos genéricos (independente da rede que tenha sido definida) para ir buscar aos outros módulos a informação que precisa para construir uma representação interna da topologia da rede (na classe `INFORMACAOREDE`).

Na simulação ao fluxo isto é feito no início e a partir daí esse conjunto de módulos nunca mais é usado nessa corrida, usando-se a representação interna construída.

O `MODULOBASETABELAS` tem no princípio da simulação uma função de inicialização (carrega a informação da topologia, e a partir dela cria a informação da rede; escolhe as tabelas a usar e inicializa a recolha de dados estatísticos). Também cria e escalona uma mensagem geradora de pedidos (associada à geração de pedidos com uma determinada origem e um determinado destino e ao escalonamento desse próximo pedido) e uma mensagem geradora de falhas (o mesmo para as falhas). As mensagens geradoras implicam a criação de mensagens pedido e falhas. Qualquer mensagem recebida é tratada no `MODULOBASETABELAS` pela função `HANDLEMESSAGE`, que depois, em função do tipo de mensagem recebida a passa à função `ACONTECIMENTOPEPIDO` que a trata directamente (para as mensagens geradoras e para as mensagens correspondentes a uma falha) ou a passa a `ACPEDIDOCOMECAACABA` (para as mensagens correspondentes a um pedido)

O método `ACPEDIDOCOMECAACABA` é responsável pela evolução dos estados do pedido. Para tal vai usar a classe `REGISTAESTADOSTABELAS`, classe que associa a cada pedido em execução uma máquina de estados e permite a evolução dos mesmos através dessa máquina de estados. Tal é implementado através de uma classe auxiliar `REGISTAESTADOSTABELAS::REGPEDIDO`. Quando um pedido é escalonado para ser tratado, a classe `REGISTAESTADOSTABELAS` identifica esse pedido numa tabela de `HASH` de `REGISTAESTADOSTABELAS::REGPEDIDO`. Esta tabela inclui todos os pedidos em execução (`REGISTAESTADOSTABELAS` acrescenta o pedido nessa tabela se ainda lá não estiver). Após encontrar o pedido, `REGISTAESTADOSTABELAS` executa o método `REGISTAESTADOSTABELAS::REGPEDIDO::ACONTECIMENTOPEPIDO` associado ao pedido a tratar. Esta função contém a definição total da máquina de estados, e vai permitir a evolução do pedido ao longo da máquina de estados. Essa evolução implica o passar de estado transitório em estado transitório, fazendo as operações associadas a esses estados transitórios (que podem incluir por

exemplo efectuar a escolha de esquema de recuperação usando tabelas, o efectuar ou libertar reservas ou a recolha de informação estatística). Esta evolução vai ocorrer até que o pedido chegue a um estado estacionário. Quando tal ocorre, e após efectuar as operações associadas a esse estado, a função termina, e o método `MODULOBASETABELAS::ACPEDIDOCOMECAACABA` continua o tratamento, que geralmente implica a recolha de algumas medidas estatísticas e o escalonamento da mensagem pedido. Este ciclo de tratamento/escalonamento, associado a uma mensagem pedido vai ocorrer até a mensagem chegar ao estado `FIM_PEDIDO`, em que é finalmente eliminada (e removida dos registos de `REGISTAESTADOSTABELAS`).

Cada uma das mensagens usadas no sistema tem uma definição (indicando tipo de mensagem e outra informação associada à mensagem) num ficheiro `.MSG`. Esta definição é transformada pelo OMNeT++ numa classe apropriada (no simulador usa-se `MSGGERADOR`, `MSGGERADORHIGHLOW`, `MSGFALHA` e `MSGPEDIDO`).

O código das mensagens é transformado em código C++ usando o comando `OPP_MSGC` (do ambiente de desenvolvimento OMNeT++). Este cria as classes C++ para as mensagens definidas nos ficheiros `.MSG` (no caso em estudo no ficheiro `MODULOBASE.MSG`), criando funções `GET` e `SET` para cada um dos campos definidos para a mensagem, bem como o construtor, destrutor e operadores de comparação e atribuição.

As mensagens são construídas numa hierarquia cuja base é `MSGBASE` (que define para todas as mensagens um campo que indica o seu tipo). As mensagens `MSGPEDIDO`, `MSGFALHA` e `MSGGERADOR` são especializações da `MSGBASE`, com informação adicional associada à sua função (por exemplo, uma `MSGPEDIDO` guarda informação sobre um `PEDIDO`, enquanto uma `MSGFALHA` tem campos que indicam o instante de início e de fim da falha). As `MSGGERADOR` são usadas para guardar informação dos geradores de pedidos e de falhas (e mais um campo para permitir distinguir entre mensagens geradoras de pedidos e de falhas). Para simular carga variável, foi definido um tipo extra `MSGGERADORHIGHLOW`, como especialização de `MSGGERADOR`, que permite guardar a informação adicional necessária para permitir duas taxas de geração de pedidos e alternar entre estas taxas.

5.4.2 Suporte para múltiplos esquemas de recuperação

A classe `ESQUEMARECUPERACAO` é usada no suporte a múltiplos esquemas de recuperação. Esta classe define um conjunto de métodos virtuais puros, cuja implementação nos seus filhos (correspondentes a cada esquema) irão definir

a determinação de caminhos quando surge um pedido ou uma falha. As funções principais são: `DETERMINACAMINHOS`, que determina o CA e o/os CR (este último é inválido no caso de esquemas de reencaminhamento); e `DETERMINACAMINHORECUPERACAONAFALHA`, que determina o CR na falha (só definida nos esquemas de reencaminhamento), recebendo o CA e informação sobre a localização da falha. As restantes são funções auxiliares para actualizar estruturas internas de dados dos esquemas.

A sequência de chamada das funções é `DETERMINACAMINHOS`, `ESTABELECECAMINHOACTIVO`, `ESTABELECECAMINHORECUPERACAO` (que tratam das reservas RSVP) e `REGISTACAMINHOACTIVO` e `REGISTACAMINHORECUPERACAO` (que efectuam a alteração das estruturas de dados internas do esquema). Quando o pedido termina a ordem é a inversa, ou seja primeiro chama-se `REGISTALIBERTACAMINHOACTIVO` e `REGISTALIBERTACAMINHORECUPERACAO` e só depois o `REMOVECAMINHOACTIVO` e `REMOVECAMINHORECUPERACAO` (primeiro alteram-se as estruturas de dados internas e depois removem-se as reservas RSVP).

A classe `ESQUEMARECUPERACAO` possui dois ponteiros que lhe permitem aceder à informação comum da rede (topologia – arcos, nós, estruturas que tem que ser partilhadas por várias instâncias de um esquema de recuperação, etc) e das reservas RSVP, classes `INFORMACAOREDE` e `RESERVARSVP` respectivamente (descritas na Secção 5.4.5). Na simulação ao fluxo não é utilizado o RSVP como definido nos RFC, mas sim um RSVP simplificado (a diferença principal entre estas implementações tem a ver com a altura em que são efectuadas as reservas em cada um dos ramos dos caminhos, na implementação efectuada são todos reservados simultaneamente ou seja não existe passagem de tempo entre a reserva dos vários ramos). Os diferentes esquemas invocam os métodos de `RESERVARSVP` com parâmetros diferentes. O método `REMOVECAMINHOACTIVO` remove as reservas que tiverem sido feitas no `ESTABELECECAMINHOACTIVO` e `REMOVECAMINHORECUPERACAO` remove as reservas que tiverem sido feitas no `ESTABELECECAMINHORECUPERACAO`.

Os métodos `REGISTACAMINHOACTIVO` e `REGISTACAMINHORECUPERACAO` e os métodos `REGISTALIBERTACAMINHOACTIVO` e `REGISTALIBERTACAMINHORECUPERACAO` nalguns esquemas são complexos, principalmente naqueles que fazem partilha de LB de protecção, enquanto que noutros não precisam de fazer nada (caso dos de reencaminhamento).

As funções da classe `ESQUEMARECUPERACAO` começam por ser chamadas quando o pedido se encontra no estado `T_DETERMINA_CAMINHOS` da máquina de estados, onde é chamada a função `DETERMINACAMINHOS`. No estado `T_ESTABREG_CAMINHOS` são invocados os métodos `ESTABELECECAMINHOACTIVO`, `Es-`

TABELECECAMINHORECUPERACAO, REGISTACAMINHOACTIVO e REGISTACAMINHO-RECUPERACAO, e o estado T_DETERMINA_RP é onde é chamada o método DETERMINACAMINHORECUPERACAO NA FALHA (se o caminho já foi determinado anteriormente não se passa neste estado, passa-se logo para o estado T_PASSA_A_USAR_RP). O estado T_REMOVREG_CAMINHOS que será chamado depois do estado S_EM_PROGRESSO (ou no fim se houve alguma falha na reserva⁸) é onde são chamados os métodos REGISTALIBERTACAMINHOACTIVO e REGISTALIBERTACAMINHORECUPERACAO e depois o REMOVECAMINHOACTIVO e REMOVECAMINHORECUPERACAO. No estado S_TABELA_F pode determinar-se um novo CA (novo esquema), nesse caso será necessário retirar os caminhos anteriores e voltar ao estado T_DETERMINA_CAMINHOS.

A classe ESQUEMARECUPERACAO apresenta várias especializações, implementadas nas classes ESQREENCGLOBAL e ESQREENCGLOBALRSRV (reencaminhamento global, sem e com reserva, respectivamente), ESQREENCLOCAL e ESQREENCLOCALRSRV (reencaminhamento local, de novo sem e com reserva respectivamente), bem como ESQFASTREROUTE e ESQFASTREROUTEMULTICAMINHOS (implementação do esquema de protecção Fast Reroute e de uma variante do mesmo) e ESQKODIALAMGLOBAL, ESQKODIALAMLOCALBASE e ESQKODIALAMLOCALALT (esquemas de protecção com partilha da LB reservada para tal; foram desenvolvidos esquemas de âmbito global e local, bem como uma variante local melhorada).

5.4.3 Suporte para a escolha dinâmica do esquema de recuperação

Uma tabela do sistema que se pretenda simular será uma instância de TABELAMATCHING. Cada instância de TABELAMATCHING possui uma ou várias instâncias de TABELAMATCHING::LINHAMATCHING que por sua vez possui uma ou várias instâncias de TABELAMATCHING::CELULAMATCHING. Cada instância de TABELAMATCHING::CELULAMATCHING possui um vector de ponteiros para IMBase (correspondendo às condições que têm de ser verificadas para essa célula ser considerada um *match*). Para além disso cada célula possui também o resultado a aplicar no caso de um *match*. Este resultado é do tipo INFORMACAOTABELA

A classe IMBASE possui as especializações INPUTMATCH<T> e MATCHOBJECTOFUNCIONAL. A classe IMBASE define o interface a seguir pelos objectos INPUTMATCH<T>, MATCHOBJECTOFUNCIONAL e suas especializações. Este interface obriga a que cada classe derivada implemente uma função denominada MATCH que será verdadeira se a condição a testar se verificar. A existência de um vec-

⁸Na actual implementação, falhas na reserva apenas são possíveis se os esquemas calcularem incorrectamente a LB disponível e tentarem reservar mais do que o possível.

tor de ponteiros para estes objectos, nas células, permite que o teste de *matching* de células possa ser feito quaisquer que sejam as especializações efectivas das condições a associar a cada célula. Assim o template `INPUTMATCH<T>` permite testar a condição de uma instância de um objecto possuir um valor pré-definido (por exemplo o CT ser CT1). A classe `MATCHOBJECTOFUNCIONAL` especializa o interface para suportar testes mais complexos, definidos pelas classes suas especializações. Assim: as classes `MOF_CARGAREDE`, `MOF_CARGAREDELEVE`, `MOF_CARGAREDELEVEG` e `MOF_CARGAREDEPESADA` são especializações de `MATCHOBJECTOFUNCIONAL` associados ao teste de carga na rede; as classes `MOF_LOCALFALHA`, `MOF_LOCALFALHA_LA1`, `MOF_LOCALFALHA_LA2` e `MOF_LOCALFALHA_LA3` são especializações de `MATCHOBJECTOFUNCIONAL` associados ao teste do local da falha; finalmente as classes `MOF_TAMANHO LSP`, `MOF_TAMANHO LSPCURTO` e `MOF_TAMANHO LSPLONGO` são especializações de `MATCHOBJECTOFUNCIONAL` associados ao teste de uma estimativa do comprimento do CA.

A classe `INFORMACAO TABELA` possui a indicação de que ocorreu um *null match* ou no caso de ter existido efectivamente um *match* possui informação sobre as características do esquema de recuperação escolhido (`INFORMACAO ESQUEMA RECUPERACAO`) e a indicação de qual a próxima tabela a testar. Perante um *match* as características do esquema de recuperação escolhido são adicionadas às características escolhidas anteriormente para o esquema de recuperação a usar. Se foi escolhido completamente um determinado esquema, as características desse esquema sobrepõem-se às escolhas anteriores. Para obter as características dos esquemas de forma padronizada foi desenvolvida a classe `TRADUZESQ RECUPERACAO`.

As classes anteriores definem uma infraestrutura que pode ser usada para qualquer instanciação de esquemas e tabelas. Para aplicar na prática essa infraestrutura aos esquemas usados e tabelas testadas, foi necessário definir um conjunto de classes auxiliares, incluindo `INICINFORMACAO TABELAS` que define quais os tipos de tabelas efectivamente suportados na implementação efectuada (por exemplo, tabelas “mono-esquema” ou tabelas que usem esquemas diferentes para carga leve e não leve) e que necessitará de ser expandida se for necessário criar tabelas para outras condições ou combinações de condições. A classe `INICINFORMACAO ESQUEMAS` define quais são as traduções dos nomes, utilizados nas tabelas, nas classes que suportam efectivamente os esquemas, que são especializações de `ESQUEMA RECUPERACAO` (por exemplo, `EER_PG1` corresponde à classe `ESQKODIALAMGLOBAL`). Finalmente a classe `INICINFORMACAO ETIQUETAS` permite guardar informação auxiliar que associa a cada esquema de recuperação o seu âmbito (local ou global) e o seu modelo (protecção ou reencaminhamento).

Na classe `INICINFORMACAO TABELAS` são implementadas várias funções cujo objectivo é servir de suporte à função `INICIALIZA TABELAS`, que é na prática a função que inicializa todos os tipos de tabelas que poderão ser utilizadas nessa simulação.

5.4.4 Estruturas para armazenamento de Informação

Foram definidos diversas estruturas para armazenamento de informação usada (nós, arcos e caminhos da rede, pedidos e CT associados a estes). Agruparam-se esses objectos no que é designado por estruturas de armazenamento de dados, que permitem também manipulação adequada a cada um destes.

As classes `NO` e `ARCO` (e os seus agrupamentos `VECNOs` e `VECARCOS`) foram definidas para guardar informação sobre os nós e os arcos da rede, respectivamente. Um `ARCO` pode ser construído directamente a partir de dois `NO` ou através das posições destes num `VECNOs` (é geralmente usado neste caso o vector de todos os nós da rede).

Foram ainda necessárias as classes `PEDIDO` (para guardar informação sobre um pedido) e `CLASSE TRAFEGO` (informação sobre CT). Pedidos podem ser associados em `GRUPO PEDIDOS`. Para criar um `PEDIDO` é necessário conhecer os `NO` origem e destino e a `CLASSE TRAFEGO` do mesmo, bem como informação adicional como a `LB` necessária ao mesmo.

Para guardar informação de um `CA` e de um `CR` global ou local foram desenvolvidas as classes `CAMINHO`, `GRUPO CAMINHOS`, `CAMINHO ACTIVO`, `CAMINHO RECUPERACAO SINGLE` e `CAMINHOS RECUPERACAO`. Um `CAMINHO` é definido como uma sequência de `NO`. As restantes especializações de `CAMINHO` permitem guardar informação adicional necessária (como o `PEDIDO` associado) e permitem associar vários `CAMINHO` (necessário por exemplo para guardar os `CR` para esquemas de protecção local).

Finalmente, para tratar de valores que possam ser Infinitos ou Indefinidos, foi construída a classe `VALOR` que oferece estas funcionalidades, e que pode ser agrupada em `ARRAY VALORES`.

5.4.5 Estruturas para recolha de dados e estruturas auxiliares

As três estruturas principais implementadas para a obtenção e registo das estatísticas são `MBSTATBASE`, a sua derivada `KODSTATS` e a classe `KSTATSGENERICO` derivada desta última. A classe `MBSTATBASE` define apenas funções virtuais puras, tendo apenas função de interface para as classes derivadas. A classe `KODSTATS`

implementa quase todas as funções de `MBSTATBASE`, excepto duas, que são implementadas pela classe `KSTATSGENERICO`. O objecto `MODULOBASETABELAS` possui um ponteiro para `MBSTATBASE`, mas como a esse ponteiro é atribuída uma instância de `KSTATSGENERICO`, quando `MODULOBASETABELAS` invoca uma função a que é executada é a de `KSTATSGENERICO`, ou a de `KODSTATS` no caso de não ter sido redefinida em `KSTATSGENERICO` (polimorfismo).

Para obtenção das estatísticas escalares foram necessários os objectos auxiliares `TIMEDVALUE`, `TIMEDVALUECL` e `VECTIMEVALUECL`. `TIMEDVALUE` é um objecto estatístico que permite calcular valores correspondentes à média de uma medida, ponderada pelos intervalos de tempo que decorrem entre as recolhas de cada valor dessa medida. `TIMEDVALUECL` é um vector de objectos `TIMEDVALUE` em que é guardado um valor da medida por CT e um valor correspondente ao agregado dos CT. Desta forma o número de elementos do vector `TIMEDVALUECL` é igual ao número de CT mais um. Finalmente, `VECTIMEVALUECL` é um vector de objectos `TIMEDVALUECL`, geralmente com tantos elementos quantos os ramos da rede. Na prática este vector é utilizado para armazenar medidas por ramo da rede ou por esquema de recuperação.

O registo das estatísticas vectoriais implicou a definição do objecto auxiliar `OUTVEC`, derivado do objecto `cOUTVECTOR` da biblioteca de classes do OMNeT++, e que permite o registo de valores através da função membro `RECORD` de `cOUTVECTOR`.

Vários métodos da classe `KODSTATS` são responsáveis por registar no ficheiro de saídas vectoriais atributos do tipo `OUTVEC`, recorrendo à função `REGISTA`, ou responsáveis por coligir valores dos atributos escalares, dos tipos `TIMEDVALUE`, `TIMEDVALUECL` ou `VECTIMEVALUECL`, que no final da simulação serão registado no ficheiro de saídas escalares. A classe auxiliar `LARGBANDASTATS` foi também usada para registar valores estatísticos relativos à alteração da ocupação de LB.

Suporte para reservas, ocupações e múltiplos BCM

A informação da rede que deverá ser partilhada por todos os esquemas de recuperação e restante código é armazenada na classe `INFORMACAOREDE` ou em instâncias de outras classes que podem ser acedidas através desta. Esta informação inclui a topologia (carregada aquando do início da simulação) e as ocupações efectivas de cada ramo em cada instante. Estas informações são armazenadas directamente em especializações do `MODELORESTRICAO LARGURABANDA`, mas também são agregadas em estruturas de dados específicas de alguns esquemas, como `DELTA_IJ_UV`.

As ocupações dos ramos da rede são mediadas por um BCM. Estes foram implementados através de especializações da classe `MODELORESTRICAO LARGURABANDA`, designadas `MRLB_MAM` e `MRLB_MAR`. Estes tratam de armazenar e gerir informação sobre as ocupações de cada ramo considerando as restrições associadas a cada CT. Permitem também receber informação sobre a disponibilidade actual de LB em cada CT de acordo com essas restrições (e de acordo com as reservas RSVP activas).

A classe `RESERVARSPV` corresponde a uma implementação simplificada do RSVP. Implementa as funções associadas a reservas de caminhos simples (`RESERVACAMINHOACTIVO UNIFORME`, `RESERVACAMINHO RECUPERACAO UNIFORME`, `LIBERTACAMINHOACTIVO UNIFORME` e `LIBERTACAMINHO RECUPERACAO UNIFORME`) e as associadas às reservas de conjuntos de ramos não uniformes – necessários para suportar partilha de LB de protecção (`RESERVACAMINHO RECUPERACAO NAO UNIFORME` e `LIBERTACAMINHO RECUPERACAO NAO UNIFORME`). Além destas funções existem também funções auxiliares para obter o número de reservas RSVP tentadas, conseguidas e falhadas. Na implementação actual em geral todas as reservas RSVP tentadas são conseguidas, uma vez que os esquemas verificam a ocupação/reserva antes de tentar fazer novas reservas, e que não há passagem de tempo entre as reservas dos vários ramos de um caminho.

As reservas possuem uma identificação, através da qual é possível saber em que ramos foi feita reserva e de que valor. A função `RESERVACAMINHOACTIVO UNIFORME` permite fazer a reserva para o CA. Neste caso os ramos a reservar formam um caminho (o nó em que um ramo termina é o início do ramo seguinte do caminho) e a LB necessária é a mesma em todos os ramos (tratando-se de uma reserva bastante simples). A função `RESERVACAMINHOACTIVO UNIFORME` chama a função `RESERVACAMINHO U` que por sua vez irá utilizar o BCM para fazer a reserva efectiva da LB (ocorre alteração das estruturas de informação do BCM, no respeitante ao CT em causa).

As reservas de LB associadas a CR que partilham LBP são geridas usando reservas fictícias, uma vez que o valor a reservar por ramo pode ser diferente entre eles e os ramos a reservar podem não constituir um caminho. Estas questões são tratadas pelas reservas não uniformes. Para o conseguir, altera-se na prática um conjunto de reservas “reais” que já existem (uma por ramo) em função das modificações necessárias. É criada também uma nova reserva “fictícia” na qual são indicados todos os ramos efectivamente usados pelo caminho com um valor de LB zero (simplesmente para manter o registo de recursos usados). Para além de muitas reservas “fictícias” podem existir vários conjuntos de reservas “reais” simultaneamente, a cada um dos quais corresponde uma assinatura. Esta situa-

ção ocorre quando tivermos simultaneamente vários esquemas que precisem de fazer reservas não uniformes que são manipuladas independentemente. Estas reservas complexas são criadas pelo método `RESERVACAMINHORECUPERACAONAOUNIFORME`.

As funções de libertação de recursos podem simplesmente libertar a LB em caminhos uniformes anteriormente reservados (`LIBERTACAMINHOACTIVOUNIFORME`, `LIBERTACAMINHORECUPERACAOUNIFORME`). Pode também ser necessário libertar reservas não uniformes (é o caso de `LIBERTACAMINHORECUPERACAONAOUNIFORME`) e nesse caso as variações a fazer nas reservas devem ser calculadas previamente, fora da função, uma vez que não podem geralmente ser inferidas das reservas anteriores.

5.5 Opções tomadas no novo sistema de recuperação

No desenvolvimento do sistema foi necessário tomar algumas decisões de implementação que em certos casos influenciaram o comportamento final do sistema. Nesta secção são descritas as principais decisões e o seu efeito no sistema desenvolvido.

No sistema são permitidas várias classes de serviço, todas com determinadas garantias de QoS (excepto uma, de tráfego BE) e por essa razão um serviço só é fornecido se for possível oferecer largura de banda garantida (excepto para a referida classe de serviço BE).

Na implementação corrente, cada pedido gerado, de uma dada classe de tráfego, corresponde simplesmente ao pedido de um LSP para transportar o tráfego de uma origem para um determinado destino (os LSP são dirigidos por definição). Não está portanto também associado a um pedido o LSP para o tráfego de retorno.

Nos casos em estudo consideram-se redes em que é implementado DS-TE, e pretende-se que na ocorrência de uma falha as exigências de DS-TE continuem a ser cumpridas. Os caminhos de protecção e os caminhos protegidos por estes pertencem todos ao mesmo CT DS-TE.

Para adequar o sistema, apresentado no capítulo 4, a DS-TE as tabelas tiveram de ser adaptadas. Nessa adaptação, nas condições correspondentes às classes de serviço passou a usar-se CT. Os CT são agregados de tráfego que terão tratamento semelhante, como definido pelo DS-TE (Faucheur e Lai, 2003).

Por questões de simplicidade, foi assumida a inexistência de preempção relativamente a reservas ou ocupações. Porém o tratamento destas situações poderia ser incorporado no sistema através da criação de uma etapa auxiliar constituída

por tabelas que seriam avaliadas pelos pedidos preemptidos (no seu caminho activo ou no seu caminho de recuperação) de forma a escolher a acção a tomar neste caso.

Foram implementados vários esquemas de recuperação, por reencaminhamento e por protecção. Nos esquemas de protecção todos os LSP são criados com reserva. Nos esquemas de reencaminhamento os LSP activos podem ser estabelecidos com ou sem reserva (os LSP de recuperação são, por definição, apenas calculados quando a falha ocorre). No simulador desenvolvido são suportados vários CT, sendo os LSP estabelecidos com reserva em todos eles, excepto num (CT0, correspondente a tráfego BE). A utilização de reservas por CT permite um controlo mais apertado da utilização da LB (Faucheur e Lai, 2003).

O CT correspondente a tráfego BE será sempre estabelecido sem reserva por duas razões: para não gastar recursos com este CT e porque não será implementada preempção.

A implementação dos esquemas por reencaminhamento com e sem reserva permite aplicar esse tipo de recuperação tanto aos CT com reserva como ao CT sem reserva. Da combinação das explicações anteriores depreende-se que ao CT correspondente a tráfego BE será sempre aplicado um esquema por reencaminhamento sem reserva. Com a implementação apenas ao nível do fluxo os reencaminhamentos com e sem reserva serão funcionalmente equivalentes porque no reencaminhamento com reserva a informação de reserva coincide com a informação de ocupação (nas decisões de encaminhamento, nos esquemas de recuperação com reserva é usada a informação acerca das reservas enquanto que nos esquemas de recuperação sem reserva é usada a informação acerca das ocupações). No entanto são feitas ambas as implementações (com e sem reserva) por questões de coerência com restrições associadas aos CT e porque essas restrições poderão ser adequadas na análise da simulação ao nível do pacote.

Todos os esquemas de recuperação usam o modo reversivo com excepção do reencaminhamento sem reserva (apenas os esquemas de recuperação por reencaminhamento é podem ser aplicados sem reserva). Desta forma, em geral, após a recuperação da falha o tráfego retorna ao CA. Se esta opção não fosse tomada, continuar a usar o CR depois da falha ser recuperada poderia conduzir a problemas nos esquemas de recuperação que fazem partilha inter pedidos. No esquema de reencaminhamento sem reserva optou-se por continuar a usar o CR mesmo depois da falha ter sido recuperada, uma vez que os recursos do CA não foram reservados e podem já não estar disponíveis.

Se surgir uma segunda falha num pedido que utiliza um esquema de recuperação por reencaminhamento sem reserva esse pedido é terminado como con-

sequência dessa falha. Apesar de não cumprir as indicações do RFC 3469 (Sharma *et al.*, 2003) esta opção foi tomada por questões de simplicidade de implementação e por ter sido constatado que se tratava de uma situação muito pouco frequente.

No caso do reencaminhamento local, o pedido pode ser terminado na ocorrência de uma falha no CA, mesmo tendo sido possível encontrar um CR que contorne a falha. Esta situação ocorre se em algum dos ramos do CA que continua a ser usado após a recuperação a LB disponível para esse pedido segundo o BCM, no momento da falha, não for suficiente (note-se que a LB necessária nesse ramo já estava a ser usada pelo pedido, mas o acesso a essa LB tinha sido concedido quando a rede se encontrava noutro estado de disponibilidade). Muito provavelmente este pedido teria sido já terminado por preempção, se esta tivesse sido implementada.

Apesar da máquina de estados do programa de simulação estar preparado para a existência de falhas simultâneas, tal não pode ocorrer na prática. Quando um recurso falhou, nenhuma nova falha afectará o sistema enquanto a anterior não for reparada (não há a possibilidade de falhas simultâneas). Os únicos recursos que podem ser afectados por falhas são as ligações entre os LSR.

Na versão já desenvolvida o simulador permite fazer simulações apenas ao nível do fluxo. Contudo este programa de simulação foi desenvolvido de forma a poder vir a ser reutilizado na implementação ao nível do pacote, e no Apêndice B são apresentados alguns desenvolvimentos já efectuados, específicos da simulação ao nível do pacote. Para este tipo de simulação poder ser realizado falta principalmente a implementação do protocolo RSVP⁹ com as extensões necessárias para DS-TE.

Optimizações possíveis

O código desenvolvido poderá tornar-se mais eficiente. Algumas optimizações que poderão ser feitas são:

- Quando são usados dois esquemas no mesmo CT que fazem partilha inter pedido estão a ser mantidos registos independentes da LB partilhada, poder-se-á tornar a partilha mais eficiente se os dois esquemas fizerem uso comum da estrutura de registo de partilha de LB.
- Ao determinar o CR no reencaminhamento, considerar-se também como disponível a LB do CA efectivamente acessível, o que não acontece para todos os ramos do CA na implementação actual.

⁹O INET, enquadramento simulacional descrito no Apêndice B, possui já uma implementação simples do RSVP.

6 Análise Experimental

Neste capítulo são apresentados os resultados obtidos por uma análise simulacional usando a implementação descrita no Capítulo 5, de forma a validar o sistema construído. Este capítulo começa, na Secção 6.1, por indicar a topologia das redes utilizadas neste estudo experimental; a Secção 6.2 descreve as características gerais da análise simulacional; a Secção 6.3 apresenta as medidas de desempenho utilizadas; a Secção 6.4 descreve um conjunto de experiências que pretendem testar hipóteses subjacentes à abordagem proposta no Capítulo 4. Estas hipóteses (que verificam características parcelares do sistema) são apresentadas e testadas sucessivamente na Secção 6.5, terminando este capítulo com um teste a um sistema que apresenta todas as características simultaneamente, numa aplicação simplificada do modelo descrito no Capítulo 4. Na Secção 6.6 foi feita uma análise final desta experimentação.

6.1 Topologias das redes

Uma vez que os resultados experimentais podem ser fortemente influenciados pela estrutura topológica utilizada, na análise simulacional são consideradas mais do que uma topologia de rede (na prática para os resultados apresentados foram consideradas duas no total, mas o sistema permite facilmente suportar qualquer outra rede). Para obter essas topologias foi feita uma pesquisa na literatura porque era pretendido obter redes que fossem amplamente utilizadas em trabalhos de investigação da área (sendo este um aspecto importante para permitir uma comparação entre os vários esquemas de recuperação considerados e o novo esquema, ou sistema, de recuperação proposto no Capítulo 4).

A pesquisa efectuada permitiu identificar várias topologias de rede que são usadas extensivamente. Uma dessas topologias é a apresentada na Figura 6.1 (a mesma foi apresentada anteriormente no Capítulo 3), a qual é usada em vários trabalhos incluindo Calle *et al.* (2003); Kodialam e Lakshman (2000); Qiao e Xu (2002); Suri *et al.* (2001); Urra *et al.* (2005b); Xiong *et al.* (2003); Xu *et al.* (2002). Trata-se de uma rede com 15 nós e 56 ramos dirigidos (cada ramo não dirigido na figura representa dois ramos dirigidos), que neste trabalho será designada por

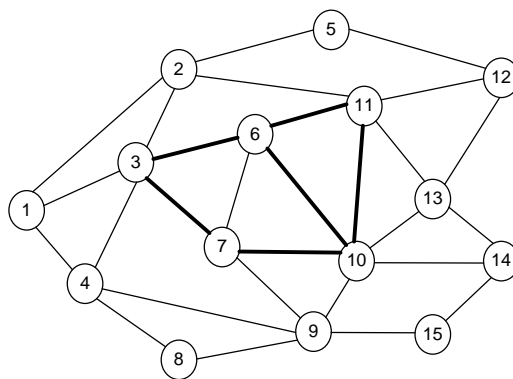


Figura 6.1: Topologia da rede KL-15 (adaptada de Kodialam e Lakshman, 2000).

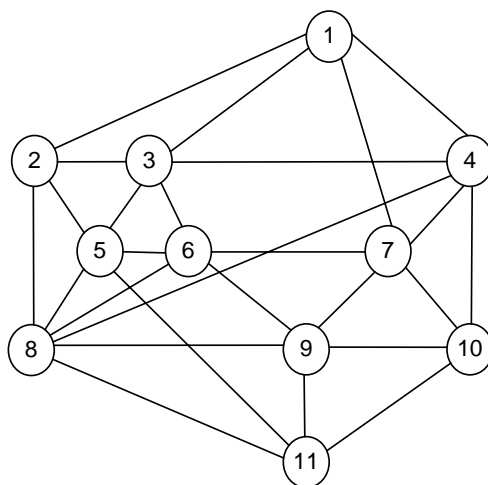


Figura 6.2: Topologia da rede europeia COST 239 (adaptada de Batchelor *et al.*, 2000).

KL-15. A capacidade de cada ramo dirigido fino é de 60 unidades e de cada ramo dirigido grosso é de 240 unidades, em ambos os casos em cada direcção.

Uma segunda topologia de rede também usada frequentemente é a rede COST 239 (Batchelor *et al.*, 2000). Exemplos de trabalhos que usam esta rede são Autenrieth e Kirstadter (2002); Iselt *et al.* (2004); Menth e Martin (2005); Pasqualini *et al.* (2004); Prinz *et al.* (2005); Schwabe e Gruber (2005). A estrutura topológica é apresentada na Figura 6.2, trata-se de uma rede com 11 nós e 52 ramos dirigidos (apenas 50 ramos em Autenrieth e Kirstadter, 2002), cada ramo não dirigido na figura representa dois ramos dirigidos.

Por último, uma topologia de rede que representa uma rede típica de um ISP é usada por exemplo em Apostolopoulos *et al.* (1998); Kar *et al.* (2003); Urra *et al.* (2005a); (Calle, 2004, pág. 106); e (Pinto, 2005, pág. 92). Esta topologia corresponde a uma rede realista bem conhecida e é constituída por 18 nós e 30 liga-

ções simétricas. Também têm sido usadas extensivamente outras topologias de rede baseadas no mapa dos EUA. Uma versão com apenas 14 nós, aparece por exemplo nos trabalhos Koubàa *et al.* (2005); Li e Wang (2005); Liu e Ruan (2005); Losego *et al.* (2005). E uma versão com 46 nós e 76 ramos aparece em Xiong *et al.* (2003); Xu *et al.* (2002). No entanto, estas redes não foram analisadas porque foi considerado que as duas iniciais (KL-15 e COST 239) tinham características suficientemente distintas.

6.2 Características gerais às várias experiências

Uma vez seleccionadas as topologias das redes a utilizar (redes KL-15 e COST 239), colocou-se o problema do seu dimensionamento. Não é possível dimensionar as redes de forma a obter uma configuração das capacidades que seja óptima para mais do que um dos cenários considerados, em particular a utilização de esquemas de recuperação com diferentes necessidades de recursos e a utilização de vários CT com requisitos diferentes. Foi escolhido um dimensionamento presente na literatura para a rede KL-15 (referido anteriormente) e para a rede COST 239 foi escolhida a capacidade de 100 unidades para cada um dos ramos dirigidos, capacidade semelhante à capacidade média na rede KL-15. Foram escolhidos factores de carga com grande variação na tentativa de produzir situações de carga adequada aos casos em estudo.

Os pedidos chegam à rede (um de cada vez) com LB uniformemente distribuídos no intervalo [1,6]. A origem e o destino dos LSP são escolhidos aleatoriamente de entre todos os nós. Também é escolhida aleatoriamente o CT associado ao pedido, a partir de uma distribuição uniforme usando o Método de Monte Carlo (proporções de tráfego indicadas no parâmetro MAM_LIM_CLASSES). As chegadas seguem um processo de Poisson de intensidades 0.1 e 0.2 (parâmetro LAMBDA_POISSON) pedidos de ligação por unidade de tempo, respectivamente para a rede KL-15 e COST 239. As durações dos pedidos são exponencialmente distribuídas com valor médio igual a $1/0.001$ ($1/\text{MIU_POISSON}$) unidades de tempo. Para gerar outras situações de carga multiplicou-se a intensidade de chegada dos pedidos por sucessivos valores inteiros (no intervalo [2,6]).

Cada falha afecta apenas um ramo dirigido que é escolhido aleatoriamente de entre todos os ramos dirigidos na rede. Estas são também geradas como processos de Poisson. Ocorrem 0.0005 (LAMBDA_FALHAS) por unidade de tempo e são reparadas 0.001 (MIU_FALHAS) por unidade de tempo, independentemente das condições de carga. O número de falhas que afectam o sistema é ligeiramente menor que o referido anteriormente porque uma falha que é gerada quando a

anterior ainda não foi recuperada é simplesmente eliminada. Esta restrição foi colocada devido à limitação dos esquemas de recuperação que não estão preparados para lidar com falhas simultâneas.

Em cada uma das corridas efectuadas foram gerados 2 800 000 pedidos e um tempo de transição correspondente a 100 000 pedidos. Este período de transição foi aplicado tanto no início como no final da simulação, no primeiro caso para evitar efeitos transitórios enquanto a rede não atingiu uma carga média estável e no segundo caso para evitar o efeito nas medidas (por exemplo no rácio de pedidos terminados / pedidos tentados) de considerar pedidos que não chegam a terminar durante o período simulado por falta de tempo de simulação. Estes valores foram escolhidos relativamente grandes para poderem ser usados tanto para a situação de carga base como para a situação de carga mais elevada (quanto maior for a intensidade de geração de pedidos menor vai ser o tempo simulado). Para as diversas experiências, os resultados apresentados referem-se a medidas durante o período útil, após o período de transição (logo, o número de pedidos analisados foi 2 600 000). As várias condições de carga (parâmetro `FACTOR_TRAFEGO`) geradas no estudo efectuado foram além da carga base, acréscimos desta de 100%, 200%, 300%, 400% e 500%, tanto na rede KL-15 como na COST 239. Verificou-se, para um número significativo de casos, que passado o tempo de transição, os resultados não apresentavam flutuações significativas. Esses resultados são relativos a uma situação com factor de carga elevado e por conseguinte uma situação em que o tempo de aquecimento usado foi menor.

Para cada um dos casos foram executadas 10 replicações independentes (com exactamente os mesmos parâmetros diferindo apenas nos números aleatórios usados) e foram obtidas as médias em todos os casos, os resultados da simulação mostram que são pequenas as variações entre as várias replicações. Os intervalos de confiança apresentados para os diferentes resultados foram calculados pelo método estatístico das replicações independentes, conforme descrito no Apêndice F.

O programa foi executado em vários microcomputadores com características distintas (com processadores Pentium III, Pentium IV, Pentium Core, e Pentium Core2 tendo sido estes dois últimos tipos os mais comuns) em máquinas virtuais com 512 MB de memória principal. O sistema operativo utilizado foi o *Linux (Ubuntu)*. O tempo de simulação para cada corrida variou de 5 minutos a 20 horas, sendo esta diferença resultante principalmente de dois factores, a grande diferença de complexidade na implementação dos vários esquemas de recuperação e a diferença de capacidade dos processadores. Para obter todos os resultados foi necessário um tempo de computação de cerca de 500 dias de CPU.

| Nome | Valor |
|--------------------|------------------------------|
| NUMMAXPEDIDOS | 2 800 000 |
| NPEDIDOINICIALSTAT | 100 000 |
| N_CLASSES_TRAF | 4 |
| MAM_LIM_CLASSES | 0,5, 0,2, 0,2 e 0,1 |
| MAR_FACTOR_CLASSES | 0, 1,5, 1,5 e 3 |
| MAR_LIMIAR | 0,1 |
| LAMBDA_POISSON | 0,1 (KL-15) e 0,2 (COST 239) |
| MIU_POISSON | 0,001 |
| LAMBDA_FALHAS | 0,0005 |
| MIU_FALHAS | 0,001 |
| BW | entre 1 e 6 |
| FACTOR_TRAFEGO | 1, 2, 3, 4, 5 ou 6 |

Tabela 6.1: Valores dos parâmetros de entrada comuns a todas as corridas.

Todos os ficheiros dos resultados obtidos nas várias corridas executadas foram consolidados em bases de dados usando *Microsoft Access*. A análise dos resultados foi feita usando o *Microsoft Excel* tendo este como entrada os dados das base de dados referidas anteriormente.

A Tabela 6.1 sintetiza os valores dos parâmetros de entrada comuns às diversas simulações e que foram utilizados na obtenção dos resultados apresentados na Secção 6.5. Na Tabela 6.1 são apresentados para o parâmetro `FACTOR_TRAFEGO` todos os valores que são utilizados (só é utilizado um valor de `FACTOR_TRAFEGO` para cada corrida). Para chegar aos valores dos parâmetros `NUMMAXPEDIDOS`, `MAR_FACTOR_CLASSES`, `MAR_LIMIAR` e `FACTOR_TRAFEGO` foram realizadas várias simulações prévias. Por exemplo, para o parâmetro `MAR_FACTOR_CLASSES` foram utilizados inicialmente os valores 0, 1, 1 e 2, verificou-se que estes valores funcionavam relativamente bem quando todos os CT utilizavam esquemas de reencaminhamento, mas quando alguns CT utilizavam esquemas de protecção existiam situações em que o bloqueio no estabelecimento do CT0 era inferior a outros CT. Com os valores actuais embora tenham sido obtidas melhorias significativas, ainda ocorrem algumas dessas situações na rede KL-15.

As proporções de tráfego (parâmetro `MAM_LIM_CLASSES`) utilizadas foram 10%, 20%, 20% e 50%, respectivamente para CT3, CT2, CT1 e CT0. CT3 corresponde a tráfego de prioridade elevada, CT2 e CT1 correspondem a tráfego de prioridade normal e CT0 corresponde a tráfego *Best Effort*. Foi atribuído um valor de proporção pequeno ao CT3 para obedecer às recomendações no RFC 4126 (Ash, 2005) e no RFC 3564 (Faucheur e Lai, 2003). Atribuíram-se proporções iguais ao

| Descrição | Cálculo |
|---|------------------------------|
| Bloqueio no Estabelecimento por CT | Expressão (5.2) |
| Desconexão na Falha por CT | Expressão (5.4) |
| Nº de Ramos (Comprimento) do CA por CT | Expressão (5.10) |
| Nº de CR por Pedido Estabelecido por CT | Expressão (5.14) |
| Nº de pedidos tentados (por cada esquema) por CT | ESQTENTADO (Tabela 5.3) |
| Nº de pedidos estabelecidos (por cada esquema) por CT | ESQESTABELECIDO (Tabela 5.3) |
| LB gasta em todos os ramos (para todos os CT) | Expressão (5.15) |
| LB gasta em todos os ramos por CT | Expressão (5.16) |

Tabela 6.2: Medidas usadas nas experiências.

CT2 e ao CT1 para facilitar a tarefa de comparação da aplicação de diferentes esquemas de recuperação ao tráfego de prioridade normal.

O modelo MAR foi utilizado em todas as corridas. Seguindo as especificações para esse modelo, RFC 4126 (Ash, 2005), a LB garantida para o CT0 (tráfego *Best Effort*) é zero, isto significa que este tráfego só será admitido se a soma de MAR_LIMIAR com a LB) solicitada pelo pedido em causa, for superior à LB) disponível. As restrições de LB (proporções de LB) para os outros CT, são 30%, 30% e 30% da LB dos ramos, respectivamente para CT1, CT2 e CT3 (estes valores são o resultado da multiplicação do parâmetro MAM_LIM_CLASSES por MAR_FACTOR_CLASSES).

6.3 Medidas de desempenho e Intervalos de Confiança

De entre as medidas simples e compostas descritas na Secção 5.2, as experiências efectuadas usam em particular as enunciadas na Tabela 6.2. A forma de cálculo de cada destas encontra-se descrita na expressão ou medida respectiva.

O cálculo da precisão dos resultados obtidos é um factor muito importante para a análise de resultados de qualquer estudo simulacional. Para este efeito, fez-se uma análise estatística, utilizando intervalos de confiança de acordo com um grau de precisão pré-definido, para cada uma das medidas analisadas do modelo. A forma de determinação dos intervalos de confiança é descrita no Apêndice F.

Os intervalos de confiança apresentados foram determinados de acordo com a Fórmula (F.10) apresentada no Apêndice F. O grau de confiança utilizado foi de 95% ao qual corresponde um alfa de 5%. Para cada valor, é apresentado o mínimo, máximo e ponto médio correspondente ao intervalo de confiança. De acordo com os resultados apresentados nas diversas tabelas, pode-se verificar que

a dimensão do intervalo de confiança é em geral aceitável.

Uma vez que a análise dos resultados é um problema multicritério, o objectivo das experiências será a apresentação de resultados que nos permitam tirar conclusões relativamente a hipóteses que irão sendo formuladas e não a escolha de um conjunto óptimo de tabelas a utilizar nas diferentes condições.

6.4 Pressupostos das experiências

6.4.1 Estratégia para a construção de tabelas

A construção do conjunto de tabelas que melhor se adequa ao tráfego a transportar, de acordo com as necessidades desse tráfego, com a capacidade de recolha de informação (pelos LSR) para a selecção e utilização dos esquemas de recuperação e dado o conjunto de esquemas de recuperação disponível, será em geral uma actividade bastante complexa, e merecedora de investigação.

Sem pretensões de ser a solução ideal (até porque não é garantido que exista uma solução ideal), é apresentada de seguida a estratégia que será usada para construir e validar um conjunto de tabelas, num ambiente simulacional. Pretende-se partir de uma construção inicial simples, e numa sucessão de etapas acrescentar/refinar as tabelas de forma a conseguir resultados mais adequados e validar a estratégia seguida. Assim, são propostas as seguinte etapas:

1. construção de uma tabela “mono-esquema” que permita obter *valores base* para cada um dos CT;
2. construção de uma tabela “multi-esquema” que escolha para cada CT um esquema de recuperação específico e adequado às suas características;
3. construção de uma tabela para adequar o esquema de recuperação a usar ao comprimento esperado do CA;
4. construção de uma tabela para adequar o esquema de recuperação a usar em função da carga actual da rede.

No fim de cada uma destas etapas serão efectuados testes para determinar as vantagens/desvantagens que esta traz ao processo e a cada um dos tipos de classe.

Se se pretender garantir SLA diferentes para diferentes pares Origem-Destino (O-D) as condições correspondentes a esses SLA deverão ser colocada em tabelas que os implementem (sob a forma de condições e tratamentos distintos para

SLA/O-D diferentes ou sob a forma de tabelas diferentes que partilham parcialmente as mesmas condições, ou seja uma tabela para o SLA_x com origem-destino O-D, outra para o SLA_y ..., etc). Desta forma torna-se mais fácil diferenciar a gestão do tráfego correspondente a cada SLA.

6.4.2 Hipóteses a testar

À sequência de tabelas, introduzida na Secção 6.4.1, está associada uma sequência de hipóteses, a confirmar:

1. Esquemas de recuperação diferentes apresentam características diferentes, e não existe um único esquema de recuperação que seja o melhor para todos os critérios;
2. A escolha de esquemas diferentes por CT permite seleccionar, para cada CT, o esquema que deverá garantir desempenho adequado da rede para tráfego desse CT;
3. Caminhos activos de comprimento diferente devem ser protegidos de forma diferente:
 - Caminhos mais curtos devem ser protegidos por mecanismos globais;
 - Em caminhos mais longos devem ser favorecidos mecanismos locais;
4. A carga da rede deve influenciar o esquema de recuperação a usar:
 - Perante cargas mais leves faz sentido usar esquemas de recuperação mais consumidores de LB mas menos sujeitos a desconexão na falha;
 - Perante cargas mais elevadas faz sentido apostar em mecanismos de partilha da largura de banda;
 - Perante cargas mais elevadas faz sentido não utilizar esquemas de recuperação de âmbito local, ou seja deve nessa condição usar-se apenas esquemas de recuperação de âmbito global;

6.4.3 Objectivos

De seguida são apresentados alguns objectivos que poderão ser considerados usando o simulador desenvolvido. Junto com cada um dos objectivos são indicadas as medidas que foram consideradas para verificar até que ponto é que os objectivos foram alcançados.

- Estabelecer o conjunto de pedidos que minimizem o bloqueio de determinado CT.

Medidas:

- Contar quantos pedidos de cada CT foram submetidos ao sistema.
- Contar quantos pedidos de cada CT foram estabelecidos.
- Determinar a partir das medidas anteriores o bloqueio no estabelecimento por CT

- Criar caminhos que forneçam a menor latência possível para os CT que o requeiram.

Medidas:

- Obter o número de nós médio dos CA.
- Considerar que o aumento de número de nós corresponde a uma maior latência.

- Oferecer sobrevivência a falhas a CT que o requeiram.

Medida:

- Obter a percentagem de pedidos que sobrevive a falhas por CT¹.

- Minimizar a sobrecarga associada ao estado.

Medida:

- Contabilizar o número de CR estabelecidos por CT².

Cada objectivo é associado a um CT (ou ocasionalmente à rede como um todo) em função das características do tráfego pressupostas para a análise. Assim, por exemplo, o CT0 é considerado em todos as hipóteses testadas como correspondendo a tráfego de baixa prioridade, pelo que alguns destes objectivos (como usar a menor LB para recuperação) se lhe aplicam, enquanto outros (por exemplo ter garantias de sobrevivência) nem sequer foram testados.

De notar que a simulação ao fluxo não nos permite facilmente verificar até que ponto são atingidos objectivos associados aos tempos de recuperação. Por esta razão, os valores obtidos devem ser interpretados com o conhecimento que o simulador pode favorecer soluções que conduzem a tempos mais longos do que os desejados.

¹Devido à implementação considerada, nomeadamente a ausência de falhas simultâneas, a sobrevivência a falhas é garantida a 100% se a recuperação usar um esquema de protecção.

²O número de CR estabelecidos é uma medida simplista da sobrecarga associada ao estado.

6.5 Organização das experiências

A Tabela 6.3 apresenta a lista dos esquemas de recuperação implementados de entre os quais serão feitas as selecções. PG1 corresponde à implementação do esquema de recuperação proposto em Kodialam e Lakshman (2003). Trata-se de um esquema de protecção global no qual o CA e o CR são determinados simultaneamente e a determinação é feita usando partilha de reserva inter pedido (*interdemand*). PG3 é também um esquema de protecção global, sendo os CA e os CR determinados entre a origem e o destino utilizando o algoritmo de *Dijkstra*. Os esquemas PL1 e PL3 são implementações simplificadas do *Fast Reroute*, variante *One-to-one Backup*, (Pan *et al.*, 2005), com partilha intra pedido (*intrademand*) e sem partilha inter pedido (*interdemand*). A diferença entre eles está na determinação dos caminhos, em PL1 é utilizado o algoritmo de *Dijkstra* para determinar o único CA e os vários CR, enquanto que em PL3 são dadas mais oportunidades ao pedido determinando-se mais do que um CA (considerando CA com até, no máximo, mais dois ramos do que o mínimo) se não for possível encontrar algum CR para o CA anteriormente determinado. PL2 corresponde à implementação de protecção local proposto em Kodialam e Lakshman (2002b) com os melhoramentos descritos na Secção 3.1 do Capítulo 3. Trata-se em esquema de protecção local no qual o CA e os CR são determinados simultaneamente e a determinação é feita usando partilha de reserva inter pedido (*interdemand*) e intra pedido (*intrademand*). Para uma descrição mais detalhada dos esquemas PG1, PL1 (PL3) e PL2 consultar a Secção 2.4 do Capítulo 2 e para PL2 consultar também a Secção 3.1 do Capítulo 3.

Os esquemas RG2, RL5, RG1 e RL4 correspondem a esquemas de recuperação por reencaminhamento, nos dois primeiros é efectuada reserva de recursos antes da sua ocupação enquanto que nos dois últimos isso já não acontece. Em todos estes quatro esquemas os CA são determinados entre a origem e o destino utilizando o algoritmo de *Dijkstra*. Este algoritmo também é utilizado na determinação dos CR para os pedidos que são afectados por falhas. Em RG2 e RG1 os CR são determinados entre a origem e o destino do pedido, enquanto que em RL5 e RL4 são determinados entre o nó antes da falha e o nó destino (ou qualquer outro nó entre o nó depois da falha e o nó destino). Para detalhes de implementação consultar a Secção 5.4.2 do Capítulo 5 e o Apêndice A.

Em termos de objectivos para cada CT, podemos distinguir entre CT0 (considerado de tráfego *Best Effort*) e CT1 a CT3 (considerados de tráfego mais valioso, merecedor de protecção). Nas hipóteses seguintes, e na ausência de clarificação adicional, e unicamente por razões de simplificação da análise, será considerado

| Nome | Descrição |
|------|--|
| PG1 | Com partilha de reserva (Kodialam e Lakshman, 2003) |
| PG3 | Sem partilha de reserva |
| PL1 | Reserva, sem partilha <i>interdemand</i> (Pan <i>et al.</i> , 2005) |
| PL2 | Reserva, com partilha <i>interdemand</i> (Kodialam e Lakshman, 2002b, versão melhorada) |
| PL3 | Reserva, sem partilha <i>interdemand</i> (Pan <i>et al.</i> , 2005, versão multi-caminhos) |
| RG1 | Sem reserva |
| RG2 | Com reserva |
| RL4 | Sem reserva |
| RL5 | Com reserva |

Tabela 6.3: Esquemas de recuperação implementados.

que CT1 a CT3 tem objectivos semelhantes em termos de recuperação (ou seja, não serão consideradas diferenças entre eles). Por esta razão, a avaliação da maior parte das hipóteses seguintes utiliza na prática o caso “mono-esquema”, ou seja, escolha do mesmo esquema ou do mesmo subconjunto de esquemas para CT1 a CT3. A excepção ao referido ocorre nas Secções 6.5.2 e 6.5.5, onde as características de cada CT são explicitadas e os seus objectivos indicados. Em todos os exemplos não será prestada atenção aos resultados do CT0, porque este não será sujeito a nenhum tipo de optimização e não apresenta quaisquer objectivos a atingir, fornecendo apenas tráfego “de fundo” para as experiências.

Embora o simulador construído permitisse testar muitas mais situações e modelos mais complexos, o estudo foi em certos pontos simplificado para viabilizar a análise efectuada. Por exemplo, definir objectivos diferentes para os CT tornaria mais difícil a comparação entre os casos por falta de um referencial comum. A necessidade de simplificar a análise é também visível na comparação das hipóteses da Secção 6.5.1 e 6.5.2. Se fossem seguidas as mesmas orientações na análise da segunda hipótese que foram usadas na primeira, ter-se-ia de analisar no total cerca de 700 tabelas, e a comparação entre estas seria ainda complicada pelo desconhecimento do efeito das interacções recíprocas.

Nas próximas experiências quando se apresenta apenas uma tabela, esta é implementada como a tabela por omissão da etapa B (sendo as restantes tabelas omissão vazias).

6.5.1 Construção de uma tabela “mono-esquema”

Na primeira experiência, iremos investigar o comportamento de diversas tabelas “mono-esquema”, ou seja, tabelas que propõem o mesmo esquema de recupera-

ção para todos os CT (com excepção do CT0). Esta abordagem pretende investigar e possivelmente confirmar o grupo de Hipóteses H1, e ao mesmo tempo obter *valores base* para as medidas para cada um dos CT por esquema de recuperação.

Hipótese H1a – Esquemas de recuperação diferentes apresentam características diferentes;

Hipótese H1b – Não existe um único esquema de recuperação que seja o melhor para todos os critérios.

A hipótese H1a postula que o desempenho de cada um dos CT (avaliado de acordo com as medidas descritas anteriormente) é dependente do esquema de recuperação usado. Ou seja, os valores das medidas (considerando todas as restantes condições, excepto o esquema de recuperação, iguais) serão dependentes do esquema de recuperação.

A hipótese H1b postula que não existe um esquema de recuperação que seja melhor em todos os critérios para todos os CT. Tal ocorre desde logo porque alguns critérios são conflitantes entre si. Para além disso, se os CT tiverem objectivos distintos, tal pode também ocorrer porque as diferenças de comportamento devidas aos esquemas de recuperação não se traduzem necessariamente nos objectivos pretendidos para os diferentes CT. Isto quer dizer que certos esquemas podem tornar mais fácil atingir os objectivos pretendidos por um determinado CT, ao mesmo tempo que obtêm resultados piores para outro CT.

A estratégia para o teste destas hipóteses passará por construir tabelas que pretendem avaliar o impacto da utilização do mesmo esquema para todos os CT, com excepção do CT0. Por questões básicas de adequação dos esquemas de recuperação às características do tráfego a transportar, as tabelas ditas “mono-esquema” não indicarão o mesmo esquema de recuperação para todos os CT uma vez que não existem variantes dos esquemas (o mesmo esquema com e sem reserva) para aplicar a todos os CT. Assim, a presença de reserva não será considerada para o CT0 (tráfego *Best Effort*), pelo que para este CT apenas serão considerados como admissíveis esquemas de reencaminhamento sem reserva. Os restantes CT no entanto usarão todos o mesmo esquema de recuperação, escolhido de entre os esquemas com reserva.

| <i>Condição</i> | CT0 | CT1 | CT2 | CT3 |
|-----------------|-----|-----|-----|-----|
| <i>Sempre</i> | RL4 | PL1 | PL1 | PL1 |

Tabela 6.4: Tabela “mono-esquema”.

O número total de tabelas usadas nesta secção é 14 o que corresponde às possíveis combinação dos dois esquemas de recuperação sem reserva com os 7 com reserva. Essas tabelas encontram-se no Apêndice C, Tabela C.1 a Tabela C.14, a Tabela 6.4 apresenta uma delas. A Tabela 6.5 mostra um resumo das 14 tabelas. Neste capítulo sempre que os tipos de tabela são referenciados é usada a notação TTxx para indicar o tipo de tabela xx (excepto nos gráficos onde aparece simplesmente o número do tipo de tabela). Note-se que o esquema de recuperação aplicado ao CT1, ao CT2 e ao CT3 é sempre o mesmo, pelo que poderia então pensar utilizar-se apenas dois CT. A razão que levou a não fazê-lo foi permitir averiguar o funcionamento do BCM sobre os CT.

Verificou-se que mesmo depois de se ter tentado ajustar os parâmetros do MAR, na rede KL-15 para factores de carga pequenos, CT0 obtém frequentemente probabilidades de bloqueio no estabelecimento inferiores aos restantes CT. Isto verificou-se apenas quando os restantes CT usam esquemas de protecção. Considera-se que esta situação ocorre devido à conjugação de os outros CT usarem protecção (e para estabelecer um pedido poderem precisar de mais do dobro da LB que é necessária para o CA do pedido) e de nessa rede existirem alguns ramos sobredimensionados (o que não acontece em COST 239).

| <i>Tipo de tabela</i> | CT0 | CT1 - CT3 |
|-----------------------|-----|-----------|
| <i>TT1</i> | RL4 | PL1 |
| <i>TT2</i> | RL4 | PL2 |
| <i>TT3</i> | RL4 | PG1 |
| <i>TT4</i> | RL4 | RG2 |
| <i>TT5</i> | RL4 | RL5 |
| <i>TT6</i> | RG1 | PL1 |
| <i>TT7</i> | RG1 | PL2 |
| <i>TT8</i> | RG1 | PG1 |
| <i>TT9</i> | RG1 | RG2 |
| <i>TT10</i> | RG1 | RL5 |
| <i>TT11</i> | RL4 | PL3 |
| <i>TT12</i> | RG1 | PL3 |
| <i>TT13</i> | RL4 | PG3 |
| <i>TT14</i> | RG1 | PG3 |

Tabela 6.5: Resumo das características das tabelas “mono-esquema”.

As Tabelas 6.6 e 6.7 apresentam a probabilidade de rejeição de pedidos no estabelecimento, conforme definido na expressão 5.1 (valor médio \pm metade do intervalo de confiança) de acordo com o esquema de recuperação utilizado, tanto por CT1 como CT3, quando CT0 utiliza o esquema RL4. Tanto na Tabela 6.6 como na Tabela 6.7 foram apresentados resultados para vários factores de carga, valores

| | | PL2 | PG1 | PG3 | PL3 | PL1 |
|---|-----|----------------|----------------|----------------|----------------|----------------|
| 1 | CT1 | 0,63% ± 0,02% | 0,68% ± 0,01% | 0,85% ± 0,02% | 1,25% ± 0,03% | 1,50% ± 0,03% |
| | CT3 | 0,53% ± 0,01% | 0,55% ± 0,01% | 0,61% ± 0,01% | 0,73% ± 0,01% | 0,83% ± 0,02% |
| 2 | CT1 | 6,68% ± 0,04% | 9,69% ± 0,04% | 12,33% ± 0,02% | 19,00% ± 0,03% | 19,16% ± 0,04% |
| | CT3 | 2,92% ± 0,02% | 3,58% ± 0,03% | 5,38% ± 0,04% | 8,57% ± 0,05% | 8,84% ± 0,05% |
| 3 | CT1 | 18,59% ± 0,05% | 26,35% ± 0,07% | 27,24% ± 0,05% | 33,38% ± 0,07% | 33,44% ± 0,05% |
| | CT3 | 8,24% ± 0,05% | 11,20% ± 0,07% | 14,24% ± 0,06% | 18,88% ± 0,09% | 19,08% ± 0,09% |
| 4 | CT1 | 27,51% ± 0,07% | 37,48% ± 0,08% | 36,67% ± 0,06% | 41,91% ± 0,07% | 41,93% ± 0,06% |
| | CT3 | 13,71% ± 0,08% | 19,11% ± 0,09% | 21,86% ± 0,08% | 26,79% ± 0,10% | 26,97% ± 0,11% |

Tabela 6.6: Bloqueio no estabelecimento para CT1 e CT3, rede KL-15 e factores de carga entre 1 e 4 - Tabelas “mono-esquema” - Protecção.

| | | RL5 | RG2 |
|---|-----|----------------|----------------|
| 2 | CT1 | 0,11% ± 0,00% | 0,11% ± 0,00% |
| | CT3 | 0,05% ± 0,00% | 0,04% ± 0,00% |
| 3 | CT1 | 1,52% ± 0,02% | 1,55% ± 0,02% |
| | CT3 | 0,46% ± 0,01% | 0,46% ± 0,01% |
| 4 | CT1 | 6,43% ± 0,04% | 6,50% ± 0,04% |
| | CT3 | 1,73% ± 0,03% | 1,76% ± 0,02% |
| 5 | CT1 | 12,75% ± 0,06% | 12,82% ± 0,05% |
| | CT3 | 3,68% ± 0,03% | 3,71% ± 0,02% |
| 6 | CT1 | 18,66% ± 0,07% | 18,74% ± 0,06% |
| | CT3 | 6,16% ± 0,05% | 6,20% ± 0,04% |

Tabela 6.7: Bloqueio no estabelecimento para CT1 e CT3, rede KL-15 e factores de carga entre 2 e 6 - Tabelas “mono-esquema” - Reencaminhamento.

entre 1 e 4 na Tabela 6.6 e entre 2 e 6 na Tabela 6.7. Apenas foram apresentados resultados quando CT0 utiliza o esquema RL4 uma vez que constatou-se que os resultados apenas sofrem alterações insignificantes se em vez de RL4 for utilizado por CT0 o esquema RG1, o mesmo acontecerá nas restantes tabelas desta secção. As Tabelas D.10 e D.11 no Apêndice D, ilustram exemplos dessas situações, apresentam resultados para os tipos de tabela TT2 (RL4/PL2) e TT7 (RG1/PL2) que apenas diferem no esquema utilizado por CT0. As Tabelas 6.6 e 6.7 encontram-se ordenadas, por coluna, por ordem crescente de bloqueio (excepto num caso na Tabela 6.7). Embora exista o pressuposto que o bloqueio de RL5 seja inferior ao de RG2 deve ser notado que existem sobreposições dos intervalos de confiança em quase todos os factores de carga. Não são apresentados valores para CT2 uma vez que, como era esperado, são praticamente iguais aos do CT1. Esta afirmação pode ser confirmada observando as Tabelas D.13 e D.14 no Apêndice D, que apresentam resultados para os casos particulares dos tipos de tabela TT2 e TT4.

| | | RG2 | RL5 |
|---|-----|----------------|----------------|
| 2 | CT1 | 2,64% ± 0,16% | 3,63% ± 0,19% |
| | CT3 | 1,45% ± 0,15% | 2,28% ± 0,24% |
| 3 | CT1 | 14,88% ± 0,35% | 17,75% ± 0,38% |
| | CT3 | 9,26% ± 0,41% | 12,20% ± 0,47% |
| 4 | CT1 | 28,11% ± 0,59% | 30,92% ± 0,79% |
| | CT3 | 17,19% ± 0,51% | 20,75% ± 0,53% |
| 5 | CT1 | 36,69% ± 0,62% | 38,87% ± 0,69% |
| | CT3 | 23,26% ± 0,37% | 26,59% ± 0,71% |

Tabela 6.8: Desconexão na falha para CT1 e CT3, rede KL-15 e factores de carga entre 2 e 5 - Tabelas "mono-esquema".

A Tabela 6.8 apresenta a probabilidade de desconexão na falha, conforme definido na expressão 5.3 (valor médio ± metade do intervalo de confiança) quando é utilizado o esquema de reencaminhamento RG2 e RL5 (é apenas nos esquemas de reencaminhamento que pode ocorrer desconexão na falha), tanto por CT1 como CT3 (CT2 aproximadamente igual a CT1, como pode ser observado no exemplo apresentado na Tabela D.15 do Apêndice D), apenas quando CT0 utiliza o esquema RL4, pelas razões já apresentadas anteriormente (exemplo na Tabela D.12 do Apêndice D). São apresentados resultados para os valores de factor de carga entre 2 e 5, verificando-se consistentemente um melhor desempenho de RG2 face a RL5.

| | | PG3 | PL1 | PL3 | PL2 | PG1 |
|---|-----|---------------|---------------|---------------|---------------|---------------|
| 1 | CT1 | 2,143 ± 0,001 | 2,150 ± 0,001 | 2,152 ± 0,001 | 2,467 ± 0,001 | 2,774 ± 0,001 |
| | CT3 | 2,138 ± 0,001 | 2,142 ± 0,001 | 2,142 ± 0,001 | 2,436 ± 0,001 | 2,653 ± 0,001 |
| 2 | CT1 | 2,177 ± 0,001 | 2,172 ± 0,001 | 2,183 ± 0,001 | 2,522 ± 0,001 | 2,925 ± 0,002 |
| | CT3 | 2,163 ± 0,001 | 2,169 ± 0,001 | 2,176 ± 0,000 | 2,495 ± 0,000 | 2,791 ± 0,002 |
| 3 | CT1 | 2,170 ± 0,001 | 2,149 ± 0,001 | 2,162 ± 0,001 | 2,576 ± 0,001 | 2,968 ± 0,001 |
| | CT3 | 2,175 ± 0,001 | 2,168 ± 0,001 | 2,178 ± 0,001 | 2,543 ± 0,001 | 2,875 ± 0,002 |
| 4 | CT1 | 2,154 ± 0,001 | 2,128 ± 0,001 | 2,140 ± 0,001 | 2,581 ± 0,002 | 2,967 ± 0,003 |
| | CT3 | 2,174 ± 0,001 | 2,159 ± 0,001 | 2,170 ± 0,001 | 2,570 ± 0,001 | 2,921 ± 0,002 |

Tabela 6.9: Número de ramos dos CA para CT1 e CT3, rede KL-15 e factores de carga entre 1 e 4 - Tabelas "mono-esquema" - Protecção.

A Tabela 6.9 apresenta-se ordenada por colunas tendo em consideração apenas a carga nominal porque os esquemas PG3, PL3 e PL1, embora apresentem comprimentos de CA muito próximos, não mantêm a ordenação relativa entre eles com o aumento dos factores de carga. O esquema PL1 é aquele que apresenta comprimentos de CA menores para mais factores de carga. Os esquemas PL2 e PG1 apresentam comprimentos de CA significativamente superiores. Entre os esquemas RL5 e RG2 não há praticamente distinção em termos de comprimento

do CA, como se pode verificar na Tabela 6.10.

| | | RL5 | RG2 |
|---|-----|---------------|---------------|
| 1 | CT1 | 2,142 ± 0,001 | 2,142 ± 0,001 |
| | CT3 | 2,141 ± 0,000 | 2,141 ± 0,000 |
| 2 | CT1 | 2,173 ± 0,001 | 2,173 ± 0,001 |
| | CT3 | 2,155 ± 0,001 | 2,156 ± 0,001 |
| 3 | CT1 | 2,282 ± 0,001 | 2,283 ± 0,001 |
| | CT3 | 2,213 ± 0,001 | 2,214 ± 0,001 |
| 4 | CT1 | 2,410 ± 0,001 | 2,411 ± 0,001 |
| | CT3 | 2,295 ± 0,001 | 2,297 ± 0,001 |
| 5 | CT1 | 2,480 ± 0,001 | 2,481 ± 0,001 |
| | CT3 | 2,362 ± 0,001 | 2,363 ± 0,002 |

Tabela 6.10: Número de ramos dos CA para CT1 e CT3, rede KL-15 e factores de carga entre 1 e 5 - Tabelas “mono-esquema” - Reencaminhamento.

A observação dos resultados das Tabelas 6.6, 6.7, 6.8 e 6.9 permite concluir que não existe um esquema que seja superior em todas as medidas analisadas. No entanto observa-se que PL2 é superior a PG1, para todos os factores de carga, e que PG3 é superior a PL3 e PL1. Note-se que, como foi referido, a superioridade de PG3 em termos de comprimento do CA só se verifica para a carga nominal.

Verifica-se que nos esquemas de protecção, a partir do factor de carga 2, em geral a LB usada por CA decresce (Tabela 6.11), o que aponta para uma situação de congestão da rede. Já nos esquemas de reencaminhamento (Tabela 6.12) tal não acontece. É significativo o aumento de LB por CA dos esquemas que fazem partilha (PL2 e PG1) relativamente aos que não fazem partilha (PG3, PL1 e PL3). Entre os esquemas com partilha, também a LB por CA para PG1 é significativamente superior a PL2.

| | | PG3 | PL3 | PL1 | PL2 | PG1 |
|---|-----|---------------|---------------|---------------|---------------|---------------|
| 1 | CT1 | 5,494 ± 0,003 | 5,504 ± 0,003 | 5,509 ± 0,003 | 6,335 ± 0,004 | 7,115 ± 0,005 |
| | CT3 | 5,495 ± 0,005 | 5,497 ± 0,005 | 5,498 ± 0,004 | 6,420 ± 0,007 | 6,932 ± 0,005 |
| 2 | CT1 | 5,781 ± 0,003 | 5,639 ± 0,003 | 5,589 ± 0,004 | 6,376 ± 0,004 | 7,684 ± 0,006 |
| | CT3 | 5,622 ± 0,005 | 5,693 ± 0,004 | 5,682 ± 0,005 | 6,352 ± 0,004 | 7,137 ± 0,007 |
| 3 | CT1 | 5,413 ± 0,003 | 5,016 ± 0,003 | 4,979 ± 0,004 | 6,875 ± 0,004 | 7,521 ± 0,006 |
| | CT3 | 5,818 ± 0,006 | 5,724 ± 0,005 | 5,681 ± 0,005 | 6,550 ± 0,008 | 7,592 ± 0,006 |
| 4 | CT1 | 4,932 ± 0,005 | 4,527 ± 0,005 | 4,499 ± 0,005 | 6,573 ± 0,005 | 7,085 ± 0,009 |
| | CT3 | 5,751 ± 0,005 | 5,471 ± 0,005 | 5,423 ± 0,004 | 6,753 ± 0,007 | 7,741 ± 0,007 |

Tabela 6.11: LB média dos CA para CT1 e CT3, rede COST 239 e factores de carga entre 1 e 4 - Tabelas “mono-esquema” - Protecção.

Da Tabela 6.11, relativa à LB dos CA, verifica-se que PL3 e PL2 apresentam sobreposição dos intervalos de confiança na carga nominal. Para os restantes

factores de carga o consumo de LB pelos CA é superior em PL3.

| | | RL5 | RG2 |
|---|-----|---------------|---------------|
| 1 | CT1 | 5,490 ± 0,003 | 5,490 ± 0,003 |
| | CT3 | 5,494 ± 0,005 | 5,494 ± 0,005 |
| 2 | CT1 | 5,514 ± 0,003 | 5,514 ± 0,003 |
| | CT3 | 5,502 ± 0,004 | 5,502 ± 0,004 |
| 3 | CT1 | 5,713 ± 0,004 | 5,714 ± 0,004 |
| | CT3 | 5,578 ± 0,005 | 5,577 ± 0,005 |
| 4 | CT1 | 6,343 ± 0,008 | 6,354 ± 0,006 |
| | CT3 | 5,789 ± 0,005 | 5,790 ± 0,006 |

Tabela 6.12: LB média dos CA para CT1 e CT3, rede COST 239 e factores de carga entre 1 e 4 - Tabelas "mono-esquema" - Reencaminhamento.

Observa-se, na Tabela 6.13 que existe uma grande diferença entre a LB média para os CR de um pedido nos esquemas que fazem partilha (PG1 e PL2) relativamente à dos que não fazem partilha (PG3, PL3 e PL1). É também visível que nos esquemas com partilha, a LB média para os CR para PG1 é inferior a PL2.

| | | PG1 | PL2 | PG3 | PL3 | PL1 |
|---|-----|---------------|---------------|---------------|----------------|----------------|
| 1 | CT1 | 1,744 ± 0,003 | 2,131 ± 0,003 | 7,734 ± 0,005 | 10,969 ± 0,006 | 11,024 ± 0,006 |
| | CT3 | 2,116 ± 0,007 | 2,280 ± 0,007 | 7,735 ± 0,005 | 10,932 ± 0,009 | 10,965 ± 0,008 |
| 2 | CT1 | 1,702 ± 0,005 | 1,978 ± 0,003 | 8,436 ± 0,007 | 12,213 ± 0,007 | 12,113 ± 0,008 |
| | CT3 | 1,748 ± 0,004 | 2,137 ± 0,004 | 7,964 ± 0,004 | 11,702 ± 0,010 | 11,708 ± 0,011 |
| 3 | CT1 | 1,518 ± 0,003 | 1,888 ± 0,004 | 8,523 ± 0,009 | 11,311 ± 0,008 | 11,238 ± 0,010 |
| | CT3 | 1,650 ± 0,004 | 2,032 ± 0,004 | 8,495 ± 0,009 | 12,293 ± 0,009 | 12,215 ± 0,013 |
| 4 | CT1 | 1,306 ± 0,008 | 1,695 ± 0,004 | 8,007 ± 0,005 | 10,345 ± 0,012 | 10,300 ± 0,013 |
| | CT3 | 1,565 ± 0,006 | 2,003 ± 0,004 | 8,716 ± 0,007 | 12,064 ± 0,012 | 11,975 ± 0,010 |

Tabela 6.13: LB média para os CR de um CA para CT1 e CT3, rede COST 239 e factores de carga entre 1 e 4 - Tabelas "mono-esquema" - Protecção.

Nos esquemas de reencaminhamento verifica-se (Tabela 6.14) um maior consumo de LB para os CR de um pedido em RG2 que em RL5 (excepto para o factor de carga 4, onde existe sobreposição dos intervalos de confiança).

Quando se utiliza a rede COST 239 o comprimento do CA (Tabela D.4, no Anexo D) apresenta valores inferiores aos apresentados com a rede KL-15, mas em geral as relações entre os diversos esquemas mantêm-se (ou seja quando existem esquemas com medidas significativamente maiores que outros essas relações mantêm-se; quando os esquemas apresentam medidas próximas entre si numa rede essas medidas na outra rede também estarão próximas entre si). O mesmo se verifica, na maior parte dos casos, para as outras medidas. Estas observações podem ser confirmadas através das comparações das tabelas nesta secção com as tabelas respectivas da Secção D.1 do Apêndice D.

| | | RL5 | RG2 |
|---|-----|---------------|---------------|
| 1 | CT1 | 7,320 ± 0,035 | 7,920 ± 0,037 |
| | CT3 | 7,385 ± 0,054 | 7,988 ± 0,050 |
| 2 | CT1 | 7,510 ± 0,037 | 7,981 ± 0,031 |
| | CT3 | 7,476 ± 0,070 | 7,986 ± 0,073 |
| 3 | CT1 | 8,380 ± 0,047 | 8,498 ± 0,049 |
| | CT3 | 8,211 ± 0,054 | 8,375 ± 0,033 |
| 4 | CT1 | 9,031 ± 0,074 | 9,014 ± 0,052 |
| | CT3 | 8,957 ± 0,080 | 8,936 ± 0,064 |

Tabela 6.14: LB média para os CR de um CA para CT1 e CT3, rede COST 239 e factores de carga entre 1 e 4 - Tabelas "mono-esquema" - Reencaminhamento.

Da Tabela 6.6, relativa à probabilidade de rejeição de pedidos no estabelecimento de acordo com o esquema de recuperação utilizado, podemos observar que se confirma que os esquemas com partilha de LB reservada para protecção têm probabilidades de rejeição menores que os que não fazem partilha. Da mesma tabela podemos também confirmar que os esquemas de recuperação de âmbito local apresentam bloqueios superiores aos esquemas de âmbito global (comparar PG3 com PL1 e PL3). No entanto esta afirmação só é válida quando não há partilha inter pedido. Nos esquemas em que há partilha inter pedido a eficiência na partilha existente pode fazer com que o local tenha melhor desempenho que o global em termos de bloqueio, é o que acontece com PL2 e PG1. Como foi referido no Capítulo 3, PL2 foi sujeito a vários melhoramentos, e pode ser essa a razão para os resultados obtidos, mas não o podemos afirmar uma vez que não obtivemos resultados com a implementação inicial de PL2 nas mesmas condições em que obtivemos os resultados para PG1, nomeadamente no respeitante à existência de falhas. Outro factor que contribui para a maior probabilidade de bloqueio no estabelecimento de PG1 é este usar CA com número de ramos significativamente superior ao usado pelo PL2 (ver Tabela 6.9).

A análise dos resultados na Tabela 6.6 mostra que PL2 é melhor que PG1 em termos de bloqueio no estabelecimento, o que pode justificar-se devido a PL2 usar CA significativamente mais curtos que PG1, e tal não ser compensado por PG1 usar CR que consomem menos LB (dos resultados nas Tabelas 6.9 e 6.13). Isto parece mostrar que PG1 tenta otimizar demasiado a partilha à custa de criar CA mais longos, o que é pior do que partilhar ligeiramente menos mas usar menos recursos nos CA. Isto faz com que PL2 seja, geralmente, preferido a PG1.

O bloqueio no estabelecimento é menor nos esquemas com partilha de reserva, PL2 e PG1, do que nos que não fazem partilha (PG3, PL3 e PL1). No entanto para o comprimento dos CA verifica-se o inverso ou seja os esquemas com partilha são

os que obtêm comprimentos maiores (e por conseguinte maiores LB para os CA). Por outro lado relativamente ao consumo de LB para os CR os esquemas PL2 e PG1 são os que gastam menos, como era de esperar.

Ainda relativamente à probabilidade de rejeição de pedidos no estabelecimento, mas agora para os esquemas de reencaminhamento, observa-se que é pouco sensível à utilização de recuperação de âmbito local ou global (ver Tabela 6.7), o mesmo não acontece relativamente à desconexão na falha (ver Tabela 6.8).

6.5.2 Construção de uma tabela de esquema por CT

Construção de uma *tabela de esquema por CT* que apresente para cada CT um esquema próprio.

Hipótese H2 – A escolha de esquemas diferentes por CT permite seleccionar para cada CT o esquema que deverá garantir desempenho adequado da rede para tráfego desse CT.

Para testar esta hipótese construiremos tabelas que utilizem esquemas de recuperação diferentes por CT. Pretende-se com isto verificar as consequências que traz utilizar esquemas diferentes por CT.

Uma vez que não é aceitável construir tabelas para todas as combinações de esquema/CT, que forneçam *tabelas de esquema por CT*, optou-se por limitar a análise a CT com requisitos pré-definidos que nos apontam para alguns esquemas com características apropriadas a estes CT (a cada um dos CT associou-se apenas uma Classe de Serviço). A Tabela 6.15 apresenta os requisitos que foram considerados na determinação do(s) esquema(s) mais apropriados a cada um do CT. Foram utilizados como requisitos o atraso no estabelecimento de pedidos e as receitas geradas pelos pedidos. Relativamente à sensibilidade ao atraso no estabelecimento considerou-se CT3 e CT2 muito sensíveis, CT1 pouco sensível e CT0 insensível. Relativamente à receita gerada considerou-se que CT3 e CT1 produzem muita, CT2 alguma e CT0 muito pouca. Como medidas indicadoras dos requisitos utilizou-se o número médio de ramos do CA e o bloqueio médio dos pedidos (considerando o bloqueio nos estabelecimentos e desconexão nas falhas), respectivamente para o atraso no estabelecimento dos pedidos e para as receitas geradas (assumindo que comprimentos de CA maiores correspondem a tempos de cálculo maiores e conduzem a tempos de estabelecimento maiores, e que maiores bloqueios fornecem menos receitas). Repare-se que o objectivo de construir a Tabela 6.15 foi obter, de uma forma grosseira, um conjunto de tabelas

para usar nesta experiência, utilizando simplesmente o conhecimento, acerca dos esquemas de recuperação, obtido pela experiência da Secção 6.5.1.

| <i>Requisito</i> | CT0 (BE) | CT1 (Vídeo a pedido) | CT2 (VoIP) | CT3 (Vídeo em tempo real) |
|------------------|--------------------|--------------------------------|----------------------|-------------------------------------|
| <i>Atraso</i> | -- | - | ++ | ++ |
| <i>Receita</i> | - | ++ | + | ++ |

Tabela 6.15: Classificação dos CT.

| <i>CT</i> | Esquema |
|------------|----------------|
| <i>CT3</i> | PL2 e PL3 |
| <i>CT2</i> | PL3 e PG3 |
| <i>CT1</i> | RG2, PL2 e PG1 |
| <i>CT0</i> | RL4 |

Tabela 6.16: Esquemas a usar por cada CT.

Como resultado do estudo anterior foram seleccionados para cada CT os esquemas mais adequados, a Tabela 6.16 apresenta esse esquemas. As tabelas a usar nesta secção correspondem a todas as combinações possíveis desses esquemas. Para o CT3, poderia ter sido mais apropriado, segundo os requisitos da Tabela 6.15, indicar o esquema PG3 em vez de PL3. Relativamente a CT1 foi também incluído PG1, apesar das observações feitas na secção anterior, uma vez que, dos esquemas de protecção, é o segundo a apresentar melhores resultados em termos de bloqueio no estabelecimento. Foi também incluído o esquema RG2 para esse CT, pois foi considerado que poderia ser aceitável alguma desconexão na falha para este tráfego. A Tabela 6.17 é uma das tabelas obtidas por combinação das opções da Tabela 6.16.

| <i>Condição</i> | CT0 | CT1 | CT2 | CT3 |
|-----------------|------------|------------|------------|------------|
| <i>Sempre</i> | RL4 | PG1 | PL3 | PL2 |

Tabela 6.17: Tabela “multi-esquema”.

No Apêndice C é apresentada a totalidade das tabelas analisadas nas experiências desta secção, sendo 12 no total (C.15, C.16, C.17, C.18, C.19, C.20, C.21, C.22, C.23, C.24, C.25 e C.26, as quais correspondem respectivamente aos tipos de tabela TT61 a TT72). A Tabela 6.18 apresenta um resumo dessas tabelas.

Ir-se-á tentar mostrar com a análise dos resultados obtidos com estas tabelas que a utilização de esquemas adequados a cada CT permitem mais facilmente

| Tipo de tabela | CT0 | CT1 | CT2 | CT3 |
|----------------|-----|-----|-----|-----|
| TT61 | RL4 | RG2 | PL3 | PL2 |
| TT62 | RL4 | RG2 | PL3 | PL3 |
| TT63 | RL4 | RG2 | PG3 | PL2 |
| TT64 | RL4 | RG2 | PG3 | PL3 |
| TT65 | RL4 | PL2 | PL3 | PL2 |
| TT66 | RL4 | PL2 | PL3 | PL3 |
| TT67 | RL4 | PL2 | PG3 | PL2 |
| TT68 | RL4 | PL2 | PG3 | PL3 |
| TT69 | RL4 | PG1 | PL3 | PL2 |
| TT70 | RL4 | PG1 | PL3 | PL3 |
| TT71 | RL4 | PG1 | PG3 | PL2 |
| TT72 | RL4 | PG1 | PG3 | PL3 |

Tabela 6.18: Resumo das características das tabelas de esquema por CT.

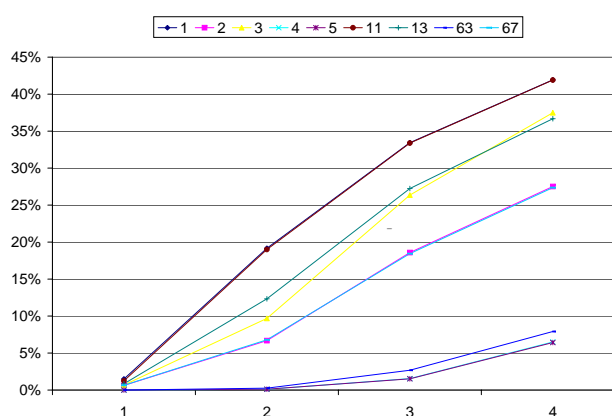


Figura 6.3: Bloqueio no estabelecimento para CT1, em função do factor de carga (rede KL-15) - “mono-esquema” versus “multi-esquema”.

(sem actuar sobre o BCM) obter as características desejadas para esse CT do que usando simples tabelas “mono-esquema”. Note-se que os resultados obtidos com TT69 a TT72 não são apresentados uma vez que pela sua análise se verificou um comportamento semelhante mas com valores menos desejáveis que os obtidos com TT65 a TT68.

Na Figura 6.3 mostra-se o bloqueio no estabelecimento para CT1, para as tabelas TT1 (PL1), TT2 (PL2), TT3 (PG1), TT4 (RG2), TT5 (RL5), TT11 (PL3), TT13 (PG3), TT63 e TT67, para o factor de carga 1, 2, 3 e 4, para a rede KL-15. Nesta figura as chaves das séries são o tipo de tabela. O mesmo acontece em geral nas restantes figuras desta e das próximas secções. Na Figura 6.3 apresenta-se apenas o resultado da tabela TT63, uma vez que este é representativo do comportamento das tabelas TT61 a TT64. Os resultados de TT67 são igualmente representativos de TT65 a TT68.

| <i>Tipo de tabela</i> | Carga 1 | Carga 2 | Carga 3 | Carga 4 |
|-----------------------|----------------|----------------|----------------|----------------|
| <i>TT5</i> | 0,00% ± 0,00% | 0,11% ± 0,00% | 1,52% ± 0,02% | 6,43% ± 0,04% |
| <i>TT4</i> | 0,00% ± 0,00% | 0,11% ± 0,00% | 1,55% ± 0,02% | 6,50% ± 0,04% |
| <i>TT63</i> | 0,00% ± 0,00% | 0,26% ± 0,01% | 2,68% ± 0,02% | 7,92% ± 0,03% |
| <i>TT67</i> | 0,64% ± 0,01% | 6,79% ± 0,03% | 18,44% ± 0,05% | 27,36% ± 0,05% |
| <i>TT2</i> | 0,63% ± 0,02% | 6,68% ± 0,04% | 18,59% ± 0,05% | 27,51% ± 0,07% |
| <i>TT13</i> | 0,85% ± 0,02% | 12,33% ± 0,02% | 27,24% ± 0,05% | 36,67% ± 0,06% |
| <i>TT3</i> | 0,68% ± 0,01% | 9,69% ± 0,04% | 26,35% ± 0,07% | 37,48% ± 0,08% |
| <i>TT11</i> | 1,25% ± 0,03% | 19,00% ± 0,03% | 33,38% ± 0,07% | 41,91% ± 0,07% |
| <i>TT1</i> | 1,50% ± 0,03% | 19,16% ± 0,04% | 33,44% ± 0,05% | 41,93% ± 0,06% |

Tabela 6.19: Bloqueio no estabelecimento para CT1, rede KL-15 e factores de carga entre 1 e 4 - “mono-esquema” versus “multi-esquema”.

Os esquemas que apresentaram melhor desempenho no “mono-esquema” para o CT1, menor bloqueio no estabelecimento, foram RL5 e RG2 (de notar que a escolha de um destes esquemas para CT1 faz com que fique sujeito a desconexão na falha, ver Figura D.1 no Apêndice D). No entanto, estes esquemas não poderiam ser usados em CT2 e CT3 uma vez que se assume que estes CT requerem protecção. Da Figura 6.3 (intervalos de confiança na Tabela 6.19) verifica-se que as tabelas TT61 a TT64 (representadas por TT63) apresentam bloqueios ligeiramente superiores aos dos esquemas RL5 e RG2 mas melhores que quaisquer outros esquemas usados em tabelas “mono-esquema”. As tabelas TT65 a TT68 (representadas na Figura 6.3 pelo resultado da tabela TT67) apresentam resultados semelhantes aos do esquema PL2 (“mono-esquema”) para o CT1 (PL2 apresenta o melhor bloqueio no estabelecimento para os esquemas de protecção). Do anterior retira-se que as Tabelas TT61 a TT68 são inferiores aos “mono-esquema” RL5 e RG2 e semelhantes ao PL2, mas melhores que os restantes “mono-esquema”. Deve ser notado que os “mono-esquema” RL5 e RG2 não são realistas, pela ausência de protecção na falha, para CT2 e CT3. Por esta razão basta demonstrar que essas tabelas (TT61–TT68) são melhores que o “mono-esquema” PL2 em algum dos restantes CT, para validar a aproximação “multi-esquema”.

A Figura 6.4 mostra o comprimento do CA para CT2, para as tabelas TT2 (PL2), TT4 (RG2), TT5 (RL5) e TT63, para o factor de carga 1, 2, 3 e 4, para a rede KL-15.

No respeitante ao CT2 o requisito mais importante é o atraso que vamos estimar a partir do comprimento do CA. Para esta medida é fácil verificar, da observação da Figura 6.4 (intervalos de confiança na Tabela 6.20), que as tabelas TT61 a TT68 apresentam melhores valores para CT2 do que o “mono-esquema” PL2 (na Figura 6.4 apresenta-se apenas o resultado da tabela TT63 uma vez que este é representativo do das tabelas TT61 a TT68). Embora não seja necessário considerar

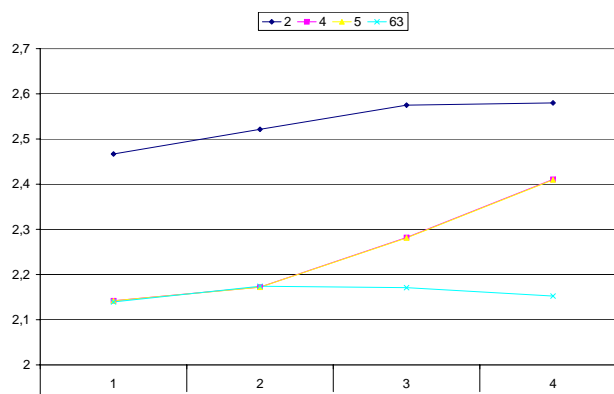


Figura 6.4: Número de ramos do CA para CT2, em função do factor de carga (rede KL-15) - “mono-esquema” versus “multi-esquema”.

| <i>Tipo de tabela</i> | Carga 1 | Carga 2 | Carga 3 | Carga 4 |
|-----------------------|----------------|----------------|----------------|----------------|
| TT63 | 2,139 ± 0,001 | 2,174 ± 0,001 | 2,171 ± 0,001 | 2,152 ± 0,001 |
| TT5 | 2,142 ± 0,001 | 2,172 ± 0,001 | 2,281 ± 0,002 | 2,410 ± 0,002 |
| TT4 | 2,142 ± 0,001 | 2,172 ± 0,001 | 2,282 ± 0,002 | 2,411 ± 0,001 |
| TT2 | 2,467 ± 0,002 | 2,521 ± 0,002 | 2,575 ± 0,001 | 2,580 ± 0,002 |

Tabela 6.20: Número de ramos do CA para CT2, rede KL-15 e factores de carga entre 1 e 4 - “mono-esquema” versus “multi-esquema”.

a comparação para as tabelas (TT61–TT64) por ausência de protecção, é possível verificar que as tabelas TT61 e TT68 também apresentam comprimentos de CA menores que os “mono-esquema” usando RL5 e RG2, visível particularmente em cargas mais elevadas.

As Figuras 6.5 e 6.6 mostram, respectivamente, o bloqueio no estabelecimento e o comprimento do CA, para CT3, para as tabelas TT1 (PL1), TT2 (PL2), TT3 (PG1), TT11 (PL3), TT13 (PG3), TT61, TT62, TT63 e TT64, para o factor de carga 1, 2, 3 e 4, para a rede KL-15. Na Figura 6.5 (Tabela 6.21) são apresentados as tabelas TT61 a TT64 pois são as que apresentam melhor desempenho para CT3.

Finalmente, no respeitante ao CT3, de novo verifica-se que algumas tabelas de TT61 a TT68, particularmente TT61 e TT63, apresentam melhores desempenhos que todas as tabelas “mono-esquema” de protecção no respeitante ao bloqueio no estabelecimento (ver Tabela 6.21) e apresentam valores competitivos no respeitante ao comprimento do CA (ver Figura 6.5). Outras tabelas na mesma gama (como 62 e 64) apresentam ainda melhores valores para o comprimento do CA, mas pior desempenho no respeitante ao bloqueio (ver Figuras 6.5 e 6.6, e intervalos de confiança respectivos nas Tabelas 6.21 e 6.22). Isto que demonstra que existe um compromisso entre estes dois valores, devendo o operador da rede

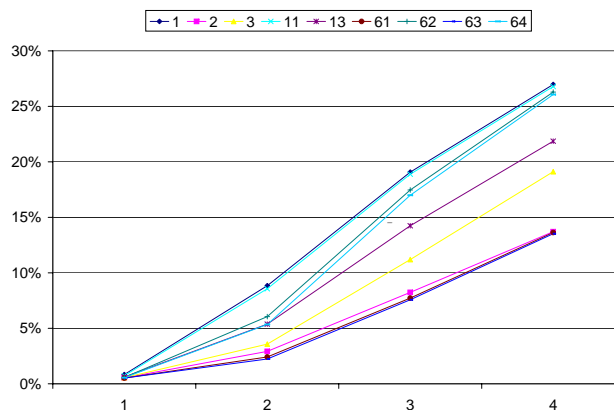


Figura 6.5: Bloqueio no estabelecimento para CT3, em função do factor de carga (rede KL-15) - "mono-esquema" versus "multi-esquema".

| Tipo de tabela | Carga 1 | Carga 2 | Carga 3 | Carga 4 |
|----------------|---------------|---------------|----------------|----------------|
| TT63 | 0,51% ± 0,01% | 2,24% ± 0,02% | 7,57% ± 0,07% | 13,52% ± 0,08% |
| TT61 | 0,52% ± 0,01% | 2,42% ± 0,03% | 7,72% ± 0,06% | 13,63% ± 0,05% |
| TT2 | 0,53% ± 0,01% | 2,92% ± 0,02% | 8,24% ± 0,05% | 13,71% ± 0,08% |
| TT3 | 0,55% ± 0,01% | 3,58% ± 0,03% | 11,20% ± 0,07% | 19,11% ± 0,09% |
| TT13 | 0,61% ± 0,01% | 5,38% ± 0,04% | 14,24% ± 0,06% | 21,86% ± 0,08% |
| TT64 | 0,56% ± 0,01% | 5,35% ± 0,05% | 16,99% ± 0,11% | 26,07% ± 0,08% |
| TT62 | 0,57% ± 0,01% | 6,05% ± 0,04% | 17,46% ± 0,09% | 26,29% ± 0,08% |
| TT11 | 0,73% ± 0,01% | 8,57% ± 0,05% | 18,88% ± 0,09% | 26,79% ± 0,10% |
| TT1 | 0,83% ± 0,02% | 8,84% ± 0,05% | 19,08% ± 0,09% | 26,97% ± 0,11% |

Tabela 6.21: Bloqueio no estabelecimento para CT3, rede KL-15 e factores de carga entre 1 e 4 - "mono-esquema" versus "multi-esquema".

escolher qual pretende beneficiar. No Apêndice D, as Figuras D.2, D.3, D.4 e D.5 (com intervalos de confiança respectivamente nas Tabelas D.16, D.17, D.18 e D.19) apresentam resultados para as mesmas medidas que as Figuras 6.19, 6.20, 6.21 e 6.22, mas para a rede COST 239.

Se no CT3 o objectivo for a minimização do bloqueio, pode-se afirmar que a tabela TT63 apresenta em geral melhores resultados que todas as tabelas "mono-esquema" de protecção para todos os CT (nos critérios mais relevantes para cada CT) sendo assim a melhor opção. De notar que a tabela TT63 corresponde à tabela construída usando o esquema que apresenta melhor bloqueio no estabelecimento para CT1 (RG2), menor comprimento de CA para CT2 (PG3) e o menor bloqueio no estabelecimento usando um esquema de protecção para CT3 (PL2). Isto parece confirmar que escolher os "melhores" esquemas de acordo com os requisitos do CT melhora o desempenho do CT nesses requisitos (o que pode parecer uma conclusão trivial, mas que poderia não ocorrer devido à interacção

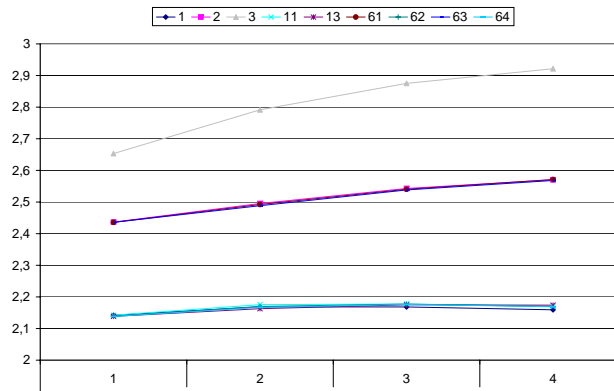


Figura 6.6: Número de ramos do CA para CT3, em função do factor de carga (rede KL-15) - “mono-esquema” versus “multi-esquema”.

| Tipo de tabela | Carga 1 | Carga 2 | Carga 3 | Carga 4 |
|----------------|---------------|---------------|---------------|---------------|
| TT1 | 2,142 ± 0,001 | 2,169 ± 0,001 | 2,168 ± 0,001 | 2,159 ± 0,001 |
| TT64 | 2,138 ± 0,001 | 2,167 ± 0,001 | 2,176 ± 0,001 | 2,169 ± 0,001 |
| TT62 | 2,139 ± 0,001 | 2,170 ± 0,001 | 2,177 ± 0,001 | 2,169 ± 0,001 |
| TT11 | 2,142 ± 0,001 | 2,176 ± 0,000 | 2,178 ± 0,001 | 2,170 ± 0,001 |
| TT13 | 2,138 ± 0,001 | 2,163 ± 0,001 | 2,175 ± 0,001 | 2,174 ± 0,001 |
| TT63 | 2,436 ± 0,001 | 2,488 ± 0,001 | 2,538 ± 0,001 | 2,568 ± 0,001 |
| TT2 | 2,436 ± 0,001 | 2,495 ± 0,000 | 2,543 ± 0,001 | 2,570 ± 0,001 |
| TT61 | 2,436 ± 0,001 | 2,492 ± 0,001 | 2,540 ± 0,002 | 2,571 ± 0,001 |
| TT3 | 2,653 ± 0,001 | 2,791 ± 0,002 | 2,875 ± 0,002 | 2,921 ± 0,002 |

Tabela 6.22: Número de ramos do CA para CT3, rede KL-15 e factores de carga entre 1 e 4 - “mono-esquema” versus “multi-esquema”.

entre CT).

6.5.3 Construção de uma tabela por comprimento esperado de CA

Cada tabela irá apresentar, para cada CT dois esquemas, um a ser usado quando o caminho esperado for curto e outro quando essa condição não se verificar. Para determinar o comprimento esperado de CA uma possibilidade seria ter um histórico de comprimentos por par origem-destino e tomar a decisão com base nesse histórico. Uma questão que aqui se colocaria seria qual a dimensão deste histórico, desde o início da simulação ou com a dimensão de uma janela pré-definida. Outra possibilidade seria estimá-lo pelo comprimento do CA mínimo possível entre a origem e o destino do pedido, em função do estado de ocupação da rede, no momento da chegada do pedido. Optou-se por esta última.

Hipótese H3 – Caminhos activos de comprimento diferente devem ser protegidos de forma diferente:

- Caminhos mais curtos devem ser protegidos por mecanismos globais.
- Em caminhos mais longos devem ser favorecidos mecanismos locais.

A Tabela 6.23 corresponde a uma das tabelas usadas nesta experiência. Cada CT utiliza um esquema de âmbito global a menos que o comprimento esperado do CA seja superior a dois, e neste caso utilizará um esquema de âmbito local.

| <i>Condição Comp. do CA</i> | CT0 | CT1 | CT2 | CT3 |
|---------------------------------|-----|-----|-----|-----|
| ≤ 2 | RG1 | PG1 | PG1 | PG1 |
| > 2 | RL4 | PL1 | PL1 | PL1 |

Tabela 6.23: Tabela por comprimento esperado de CA.

No Apêndice C encontram-se as quatro tabelas usadas na experiência desta secção, Tabela C.27, C.28, C.29 e C.30 (a Tabela 6.24 mostra um resumo delas).

| <i>Tipo de tabela</i> | CT0 | | CT1 - CT3 | |
|-----------------------|--------------|-----------|--------------|-----------|
| | CAP ≤ 2 | CAP > 2 | CAP ≤ 2 | CAP > 2 |
| TT81 | RG1 | RL4 | PG1 | PL1 |
| TT82 | RG1 | RL4 | PG1 | PL3 |
| TT83 | RG1 | RL4 | PG3 | PL1 |
| TT84 | RG1 | RL4 | PG3 | PL3 |

Tabela 6.24: Resumo das características das tabelas por comprimento esperado de CA.

Ir-se-á verificar se se consegue oferecer melhores tempos de recuperação a pedidos para os quais foram estimados CA longos, sem contudo aumentar significativamente o bloqueio. Para isto irão ser comparados os resultados da experiência mono-esquema em termos de bloqueios com os valores nesta experiência. Mais especificamente ir-se-á comparar os resultados obtidos para cada tabela nesta experiência com os resultados das duas tabelas “mono-esquema”, obtidos na Secção 6.5.1 (correspondentes aos dois esquemas usados em cada uma das tabelas desta secção). De notar que os resultados de TT81 e TT82 são semelhantes, assim como os de TT83 e TT84. Por esta razão, nas figuras e tabelas que se seguem será sempre usado apenas um de cada par (encontrando-se os resultados obtidos com o outro no Anexo D).

Os gráficos na Figura 6.7 ilustram o bloqueio no estabelecimento para CT1 e CT3, para as tabelas TT11 (PL3), TT14 (PG3) e TT84, para o factor de carga 1, 2, 3 e 4, para a rede COST 239 e KL-15 respectivamente. Nesta figura as chaves das

séries são dadas por um par (CT-Tipo de tabela). O mesmo acontece em todas as figuras em que cada chave está associada a um par de números.

Na Figura 6.7 (intervalos de confiança nas Tabelas 6.25 e 6.26) verifica-se que existe um aumento do bloqueio causado pela utilização de protecção local (o mesmo verifica-se na Figura D.7, no Apêndice D, correspondente à utilização do tipo de tabela TT81). Este aumento é tanto maior quanto mais frequente for a necessidade de utilizar protecção local, que é o caso da rede KL-15.

| | | TT11 | TT14 | TT84 |
|---|-----|----------------|----------------|----------------|
| 1 | CT1 | 0,02% ± 0,00% | 0,00% ± 0,00% | 0,00% ± 0,00% |
| | CT3 | 0,00% ± 0,00% | 0,00% ± 0,00% | 0,00% ± 0,00% |
| 2 | CT1 | 13,86% ± 0,08% | 5,06% ± 0,06% | 7,54% ± 0,07% |
| | CT3 | 1,39% ± 0,02% | 0,33% ± 0,01% | 0,51% ± 0,01% |
| 3 | CT1 | 32,85% ± 0,05% | 24,23% ± 0,07% | 27,63% ± 0,05% |
| | CT3 | 9,40% ± 0,05% | 3,96% ± 0,04% | 5,51% ± 0,04% |
| 4 | CT1 | 43,40% ± 0,06% | 36,39% ± 0,09% | 39,20% ± 0,09% |
| | CT3 | 19,71% ± 0,10% | 11,79% ± 0,07% | 14,71% ± 0,07% |

Tabela 6.25: Bloqueio no estabelecimento para CT1 e CT3, rede COST 239, factores de carga entre 1 e 4 e tipos de tabela TT11, TT14 e TT84 (PG3-PL3).

Os gráficos na Figura 6.8 mostram a LB média gasta por CT considerando todos os ramos da rede (obtida através da média aritmética dos resultados da expressão 5.16 para todos os ramos), para CT1 e CT3, para as tabelas TT8 (PG1), TT11 (PL3) e TT82, para o factor de carga 1, 2, 3 e 4, para a rede COST 239 e KL-15 respectivamente (o mesmo ilustra a Figura D.8, no Apêndice D, correspondente à utilização do tipo de tabela TT83).

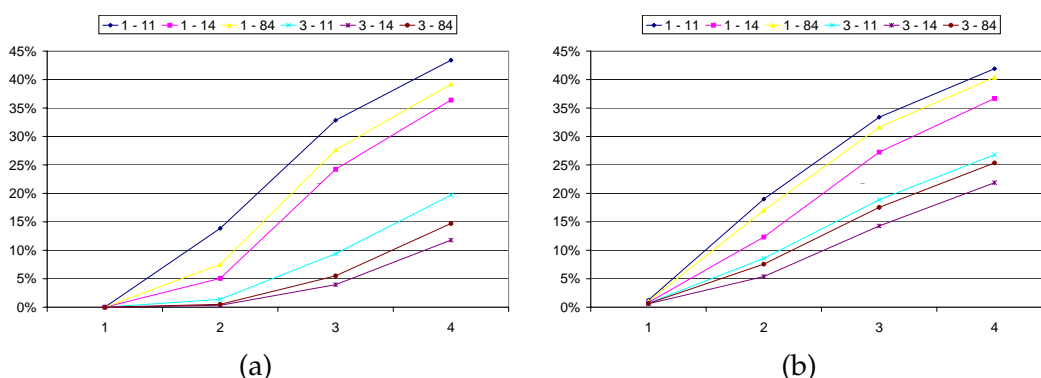


Figura 6.7: Bloqueio no estabelecimento para CT1 e CT3, em função do factor de carga - “mono-esquema” versus misto por comprimento de CA (PG3-PL3): (a) Rede COST 239. (b) Rede KL-15.

| | | TT11 | TT14 | TT84 |
|---|-----|----------------|----------------|----------------|
| 1 | CT1 | 1,25% ± 0,03% | 0,84% ± 0,02% | 1,07% ± 0,02% |
| | CT3 | 0,73% ± 0,01% | 0,61% ± 0,01% | 0,68% ± 0,01% |
| 2 | CT1 | 19,00% ± 0,03% | 12,33% ± 0,04% | 17,04% ± 0,04% |
| | CT3 | 8,57% ± 0,05% | 5,39% ± 0,04% | 7,57% ± 0,05% |
| 3 | CT1 | 33,38% ± 0,07% | 27,25% ± 0,06% | 31,64% ± 0,06% |
| | CT3 | 18,88% ± 0,09% | 14,27% ± 0,08% | 17,53% ± 0,09% |
| 4 | CT1 | 41,91% ± 0,07% | 36,68% ± 0,06% | 40,38% ± 0,07% |
| | CT3 | 26,79% ± 0,10% | 21,87% ± 0,07% | 25,35% ± 0,08% |

Tabela 6.26: Bloqueio no estabelecimento para CT1 e CT3, rede KL-15, factores de carga entre 1 e 4 e tipos de tabela TT11, TT14 e TT84 (PG3-PL3).

Na Figura 6.9 (intervalos de confiança na Tabela 6.27) mostra-se a média da LB gasta por todos os CT considerando todos os ramos da rede (obtida através da média aritmética dos resultados da expressão 5.15 para todos os ramos), para as tabelas TT8 (PG1), TT11 (PL3) e TT82, para o factor de carga 1, 2, 3 e 4, para a rede COST 239 e KL-15 respectivamente (o mesmo ilustra a Figura D.9, no Apêndice D, correspondente à utilização do tipo de tabela TT83).

Os gráficos na Figura 6.10 (intervalos de confiança nas Tabelas 6.28 e 6.29) mostram a média do número de ramos do CA, para CT1 e CT3, para as tabelas TT1 (PL1), TT8 (PG1) e TT81, para o factor de carga 1, 2, 3 e 4, para a rede COST 239 e KL-15 respectivamente. Da Figura 6.10 vemos que a rede utilizada tem grande influência no número de ramos do CA. Deve ser notado que na rede COST 239 existem valores para esta medida que são, para a tabela TT81, muito próximos dos do comprimento obtidos com a tabela TT8, chegando mesmo a ser ligeiramente maiores que estes em alguns factores de carga. Embora a aproximação aos resultados da tabela TT8 seja consistente com o observado com as outras medi-

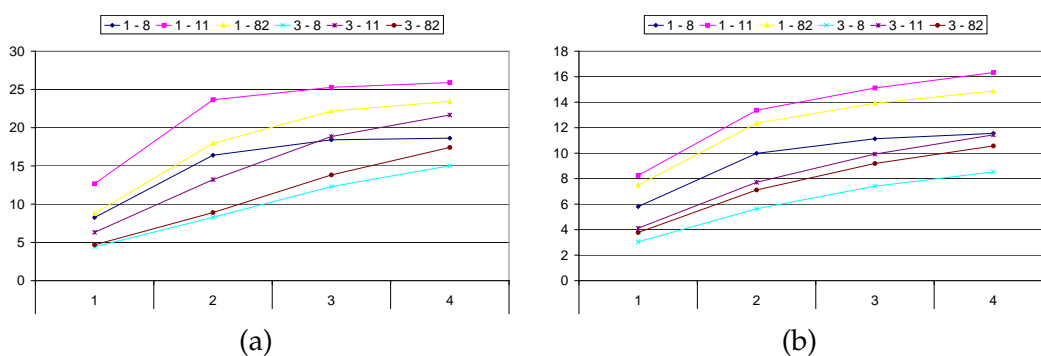


Figura 6.8: Média para todos os ramos da LB gasta, para CT1 e CT3, em função do factor de carga - "mono-esquema" versus misto por comprimento de CA (PG1-PL3): (a) Rede COST 239. (b) Rede KL-15.

| | Tipo tabela | Carga 1 | Carga 2 | Carga 3 | Carga 4 |
|----------|-------------|--------------|--------------|--------------|--------------|
| COST 239 | TT8 | 31,49 ± 0,02 | 63,14 ± 0,06 | 83,02 ± 0,02 | 86,10 ± 0,02 |
| | TT11 | 42,30 ± 0,03 | 81,32 ± 0,03 | 87,13 ± 0,01 | 88,77 ± 0,01 |
| | TT82 | 32,86 ± 0,03 | 67,48 ± 0,06 | 85,07 ± 0,02 | 87,62 ± 0,01 |
| KL-15 | TT8 | 21,40 ± 0,01 | 39,67 ± 0,02 | 48,48 ± 0,01 | 52,40 ± 0,01 |
| | TT11 | 27,47 ± 0,02 | 47,72 ± 0,02 | 55,17 ± 0,02 | 59,50 ± 0,03 |
| | TT82 | 25,55 ± 0,02 | 45,37 ± 0,03 | 52,78 ± 0,02 | 56,65 ± 0,03 |

Tabela 6.27: Média para todos os ramos da LB gasta, por todos os CT, para as redes COST 239 e KL-15, factores de carga entre 1 e 4 e tipos de tabela TT8, TT11 e TT82 (PG1-PL3).

das, não foi possível obter uma justificação inequívoca para o motivo de valores maiores (uma possibilidade poderá ser a interacção com o outro esquema).

Na Figura 6.11 apresenta-se a média do número de CR por pedido, para CT3, para as tabelas TT8 (PG1), TT11 (PL3) e TT82, para o factor de carga 1, 2, 3 e 4, para a rede COST 239 e KL-15.

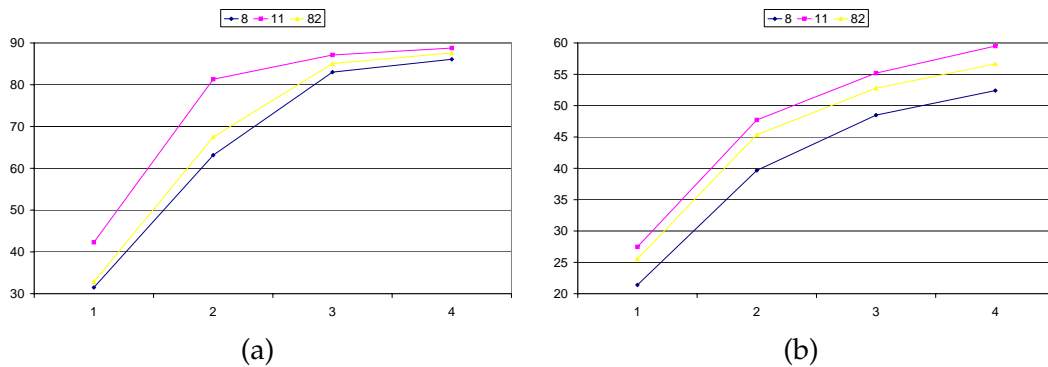


Figura 6.9: Média para todos os ramos da LB gasta, por todos os CT, em função do factor de carga - “mono-esquema” versus misto por comprimento de CA (PG1-PL3): (a) Rede COST 239. (b) Rede KL-15.

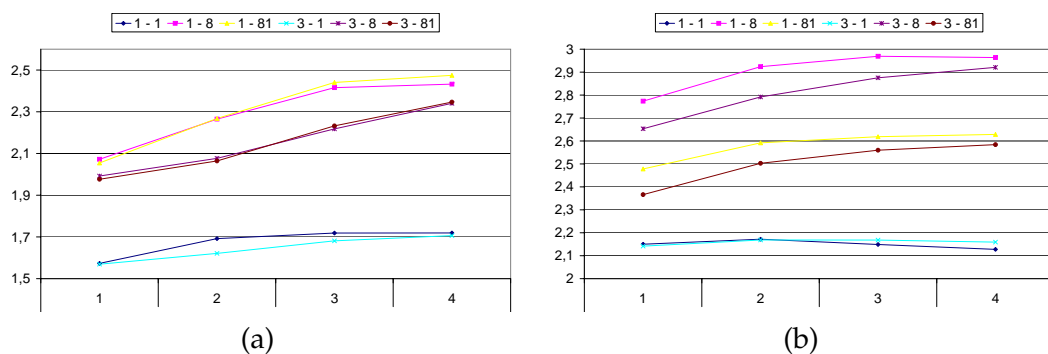


Figura 6.10: Número de ramos do CA, para CT1 e CT3, em função do factor de carga - “mono-esquema” versus misto por comprimento de CA (PG1-PL1): (a) Rede COST 239. (b) Rede KL-15.

| | | TT1 | TT8 | TT81 |
|---|-----|-------------------|-------------------|-------------------|
| 1 | CT1 | $1,573 \pm 0,001$ | $2,072 \pm 0,001$ | $2,055 \pm 0,001$ |
| | CT3 | $1,569 \pm 0,001$ | $1,992 \pm 0,001$ | $1,977 \pm 0,001$ |
| 2 | CT1 | $1,692 \pm 0,001$ | $2,265 \pm 0,002$ | $2,268 \pm 0,002$ |
| | CT3 | $1,621 \pm 0,001$ | $2,077 \pm 0,002$ | $2,064 \pm 0,002$ |
| 3 | CT1 | $1,719 \pm 0,001$ | $2,416 \pm 0,003$ | $2,441 \pm 0,001$ |
| | CT3 | $1,681 \pm 0,001$ | $2,218 \pm 0,002$ | $2,232 \pm 0,003$ |
| 4 | CT1 | $1,719 \pm 0,001$ | $2,433 \pm 0,002$ | $2,475 \pm 0,003$ |
| | CT3 | $1,707 \pm 0,001$ | $2,340 \pm 0,003$ | $2,347 \pm 0,004$ |

Tabela 6.28: Número de ramos do CA, para CT1 e CT3, rede COST 239, factores de carga entre 1 e 4 e tipos de tabela TT1, TT8 e TT81 (PG1-PL1).

| | | TT1 | TT8 | TT81 |
|---|-----|-------------------|-------------------|-------------------|
| 1 | CT1 | $2,150 \pm 0,001$ | $2,774 \pm 0,001$ | $2,478 \pm 0,001$ |
| | CT3 | $2,142 \pm 0,001$ | $2,653 \pm 0,001$ | $2,366 \pm 0,001$ |
| 2 | CT1 | $2,172 \pm 0,001$ | $2,924 \pm 0,002$ | $2,592 \pm 0,001$ |
| | CT3 | $2,169 \pm 0,001$ | $2,792 \pm 0,001$ | $2,503 \pm 0,001$ |
| 3 | CT1 | $2,149 \pm 0,001$ | $2,969 \pm 0,001$ | $2,619 \pm 0,002$ |
| | CT3 | $2,168 \pm 0,001$ | $2,875 \pm 0,002$ | $2,560 \pm 0,001$ |
| 4 | CT1 | $2,128 \pm 0,001$ | $2,964 \pm 0,002$ | $2,629 \pm 0,002$ |
| | CT3 | $2,159 \pm 0,001$ | $2,921 \pm 0,002$ | $2,584 \pm 0,002$ |

Tabela 6.29: Número de ramos do CA, para CT1 e CT3, rede KL-15, factores de carga entre 1 e 4 e tipos de tabela TT1, TT8 e TT81 (PG1-PL1).

Os resultados da combinação dos dois esquemas aproximam-se mais dos do esquema de âmbito global na rede COST 239 e dos do local na KL-15. Tal pode ocorrer porque a primeira é mais emalhada que a segunda, e por conseguinte os caminhos serão em geral mais curtos.

Observou-se que a utilização de esquemas mistos cria resultados consistentes com a conjugação das características de dois esquemas de protecção. Assim,

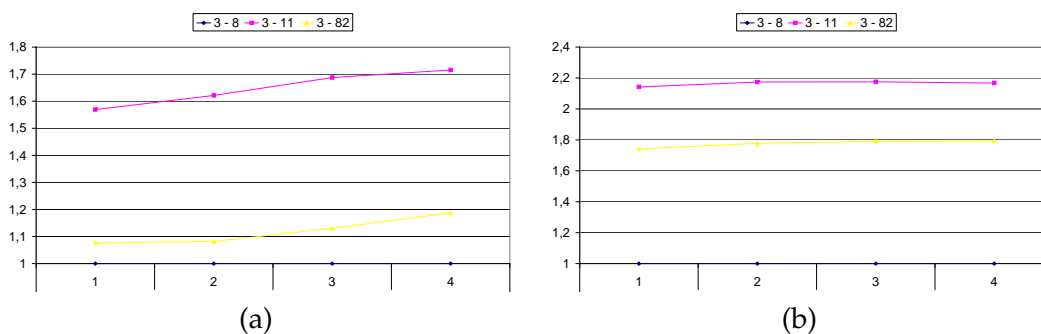


Figura 6.11: Número de CR por pedido, para CT3, em função do factor de carga - “mono-esquema” versus misto por comprimento de CA (PG1-PL3): (a) Rede COST 239. (b) Rede KL-15.

tanto quanto foi possível verificar, para as medidas apresentadas (bloqueio no estabelecimento, média para todos os ramos da LB gasta - por todos os CT ou por CT, e número de CR por pedido) qualquer dos tipos de tabelas TT81 a TT84 apresenta resultados que se localizam entre os resultados do dois esquemas que o originou. No respeitante ao comprimento dos CA embora essa situação também ocorra frequentemente existe um número relevante de excepções.

6.5.4 Construção de uma tabela por carga da rede

Construção de uma tabela que indique, para cada CT dois esquemas, um a ser usado quando a carga da rede for leve e outro quando essa condição não se verificar (excepto para o CT0, que manterá sempre o mesmo esquema).

Hipótese H4 – A carga da rede deve influenciar o esquema de recuperação a usar:

- i Perante cargas mais leves faz sentido usar esquemas de recuperação mais consumidores de LB mas menos sujeitos a desconexão na falha?
- ii Perante cargas mais elevadas faz sentido apostar em mecanismos de partilha da largura de banda?
- iii Perante cargas mais elevadas faz sentido não utilizar esquemas de recuperação de âmbito local, ou seja deve nessa condição usar-se apenas esquemas de recuperação de âmbito global?

Ao contrário do que acontece nas experiências anteriores, para tornar esta experiência mais realista ire-se-á variar periodicamente a carga oferecida à rede. Nas experiências anteriores foi usado um factor de carga, a que correspondia um sistema de geração de pedidos de Poisson com uma determinada intensidade, ou frequência, (λ), calculada como o produto do valor do factor de carga pela frequência correspondente à carga nominal (λ_{base}). Para simular períodos de carga mais leve e de carga mais pesada foi decidido criar um gerador que alterna entre duas intensidades, ou frequências, de geração: λ_{high} e λ_{low} , o que será designado por carga flutuante. Na experiência desta secção, em cada corrida metade dos pedidos são gerados com frequência de λ_{high} e a outra metade com a frequência de λ_{low} . λ_{high} e λ_{low} são calculadas de forma a que a sua média harmónica (a frequência equivalente à intensidade média de ocorrência dos pedidos) seja igual a λ . Note-se que isto implica que o sistema estará durações diferentes em “carga leve” e “carga pesada”. Desta forma, para calcular a intensidade de pedidos equivalente à intensidade efectiva de pedidos durante toda a simulação,

| Descrição | Valor |
|---|----------|
| Factor da amplitude | 1.5 |
| Número de pedidos entre leituras da carga na rede | 100 |
| Número de pedidos gerados em cada período | 10000 |
| Factor de carga | 1, 2 e 3 |

Tabela 6.30: Parâmetros de entrada adicionais (ou com valores diferentes) para a experiência desta secção.

| Factor de carga | 1 | 2 | 3 |
|------------------|-------|------|-------|
| λ_{high} | 0.215 | 0.43 | 0.645 |
| λ_{low} | 0.065 | 0.13 | 0.195 |
| Média harmónica | 0.1 | 0.2 | 0.3 |

Tabela 6.31: Valores para λ_{high} e λ_{low} .

deve ser usada a fórmula da média harmónica (para detalhes sobre o cálculo de λ_{high} e λ_{low} ver Apêndice E).

O simulador recebe um parâmetro para definir a amplitude entre λ_{high} e λ_{low} , em função do factor de carga e de λ_{base} . Essa amplitude é função do factor de carga para poder simular variações efectivas de carga mesmo quando o factor de carga é elevado.

A intensidade da carga oferecida à rede varia de 10000 em 10000 pedidos. A carga da rede será medida, periodicamente de 100 em 100 pedidos (o que corresponde a 100 leituras por período de carga), sendo a percentagem de LB média ocupada calculada considerando todos os ramos da rede (como a média aritmética das percentagens de LB ocupada em cada ramo). Foram criadas três situações de carga flutuante, correspondentes respectivamente aos factores de carga 1, 2 e 3. A Tabela 6.30 apresenta um resumo destes valores.

Para cada um dos três factores de carga os valores para λ_{high} e λ_{low} (e média harmónica respectiva) são apresentados na Tabela 6.31 (o valor da carga nominal (λ_{base}) foi mantido o das experiências anteriores).

Para averiguar a hipótese iremos construir tabelas que definem um esquema para usar se a carga for leve e outro para usar no caso contrário. Designamos por limiar de carga o valor da carga considerado a fronteira entre carga leve e não leve. São realizados testes para quatro limiares de carga distintos: 40%, 50%, 60% e 70%. Quando surge um pedido, o valor da carga que se irá usar na decisão do esquema a usar, ou seja o valor que será comparado com o limiar de carga, é o valor da carga que foi obtido para o período corrente (valor obtido

periodicamente como referido anteriormente).

Após seleccionar os esquemas que se considera que irão permitir validar o grupo de hipóteses H4 construíram-se as tabelas a usar nesta experiência. A Tabela C.31, C.32, C.33, C.34 e C.35 do Apêndice C, correspondem a essas tabelas. A cada uma destas tabelas correspondem quatro tipos de tabelas, que diferem apenas no valor do limiar de carga.

| <i>Tipo de tabela</i> | CT0 | CT1 – CT3 | |
|-----------------------------------|-----|----------------------|-------------------|
| | | $\leq \text{limiar}$ | $> \text{limiar}$ |
| <i>TT121, TT141, TT151, TT161</i> | RL4 | PL3 | RL5 |
| <i>TT122, TT142, TT152, TT162</i> | RL4 | PG3 | RL5 |
| <i>TT123, TT143, TT153, TT163</i> | RL4 | PL3 | PL2 |
| <i>TT124, TT144, TT154, TT164</i> | RL4 | PG3 | PL2 |
| <i>TT125, TT145, TT155, TT165</i> | RL4 | PL3 | PG3 |

Tabela 6.32: Resumo das características das tabelas por carga da rede.

A Tabela 6.32 apresenta um resumo das tabelas usadas nesta experiência. Os tipos de tabela 12X, 15X, 16X e 14X correspondem respectivamente a tabelas com limiares de carga 40%, 50%, 60% e 70%. De notar que os resultados de TT122, TT142, TT152 e TT162 apresentam comportamentos semelhantes aos de TT121, TT141, TT151, TT161, assim como os resultados de TT124, TT144, TT154 e TT164 apresentam comportamentos semelhantes aos de TT123, TT143, TT153 e TT163. Por esta razão apenas são apresentados os resultados dos TT ímpares neste trabalho.

As Tabelas C.31 e C.32 irão ser usadas para o teste da hipótese H4i, correspondem a utilizar PL3 ou PG3 em cargas leves e RL5 em carga não leves. As Tabelas C.33 e C.34 irão ser usadas para a verificação da hipótese H4ii, correspondem a utilizar PL3 ou PG3 em cargas leves e PL2 em carga não leves. Para o teste da hipótese H4iii irá usar-se a Tabela C.35 que correspondem a utilizar PL3 em cargas leves e PG3 em carga não leves.

Os resultados obtidos com cada um dos tipos de tabela resumidos na Tabela 6.32 serão comparados com os resultados obtidos com as duas tabelas “mono-esquema” correspondentes, sujeitas às mesmas condições de carga. Por esta razão, nesta experiência recorreremos novamente às tabelas “mono-esquema” correspondentes a PL3 (tipo de tabela TT11), PG3 (tipo de tabela TT13), RL5 (tipo de tabela TT5) e PL2 (tipo de tabela TT2). São novamente obtidos resultados com elas mas aqui sujeitas a carga flutuante. Embora as tabelas “mono-esquema” correspondentes a PL3, PG3, RL5 e PL2 sejam efectivamente as mesmas usadas

na experiência da Secção 6.5.1 decidiu-se identificá-las, respectivamente, com os tipos de tabela TT131 (PL3), TT132 (PG3), TT133 (RL5) e TT134 (PL2), para distinguir a sua utilização da situação em que não foi utilizada carga flutuante.

Os gráficos na Figura 6.12 ilustram o bloqueio no estabelecimento para CT3, para as tabelas TT121, TT131 (PL3), TT133 (RL5), TT141, TT151 e TT161, para o factor de carga 1, 2 e 3, para a rede COST 239 e KL-15 respectivamente (intervalos de confiança nas Tabelas 6.33 e 6.34).

| Tipo de tabela | Carga 1 | Carga 2 | Carga 3 |
|----------------|---------------|----------------|----------------|
| TT121 | 0,00% ± 0,00% | 0,05% ± 0,00% | 0,49% ± 0,02% |
| TT131 | 1,00% ± 0,02% | 10,66% ± 0,09% | 18,74% ± 0,06% |
| TT133 | 0,00% ± 0,00% | 0,06% ± 0,01% | 0,49% ± 0,02% |
| TT141 | 0,06% ± 0,00% | 0,08% ± 0,00% | 0,54% ± 0,02% |
| TT151 | 0,00% ± 0,00% | 0,06% ± 0,00% | 0,48% ± 0,01% |
| TT161 | 0,01% ± 0,00% | 0,06% ± 0,00% | 0,50% ± 0,01% |

Tabela 6.33: Bloqueio no estabelecimento para CT3, rede COST 239, factores de carga entre 1 e 3 e tipos de tabela TT121, TT131, TT133, TT141, TT151 e TT161 (PL3-RL5).

Os gráficos na Figura 6.13 mostram o número de pedidos tentados e o número de pedidos estabelecidos, por PL3 e por RL5, para CT3, para as tabelas TT121 e TT141, para a rede KL-15 (intervalos de confiança nas Tabelas 6.35 e 6.36). Note-se que TT121 (correspondente a limiar de carga de 40%) e TT141 (correspondente a limiar de carga de 70%) são os comportamentos extremos, apresentando TT151 e TT161 (não representados nesta figura) comportamentos intermédios.

Da Figura 6.12 e 6.13 conclui-se que, em princípio, mais pedidos serão protegidos com o aumento do limiar de decisão de carga leve (ou não leve), à custa do aumento do bloqueio no estabelecimento. Por exemplo, com limiar de carga

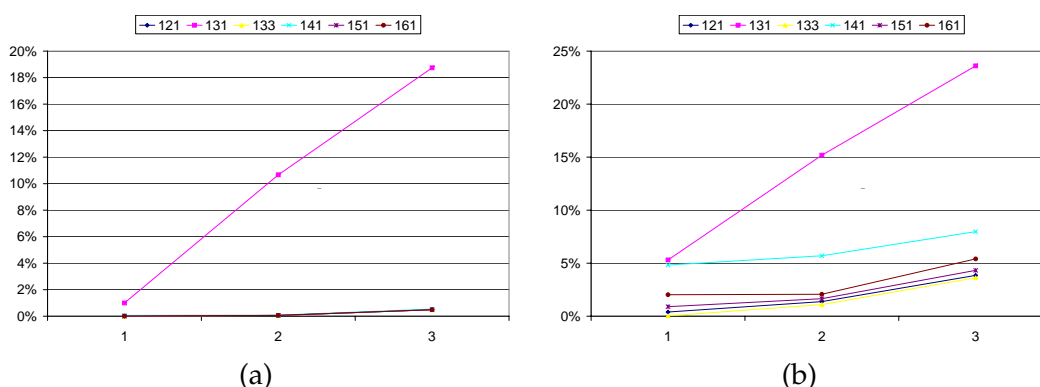


Figura 6.12: Bloqueio no estabelecimento para CT3, em função do factor de carga - “mono-esquema” versus misto por carga da rede (PL3-RL5): (a) Rede COST 239. (b) Rede KL-15.

| Tipo de tabela | Carga 1 | Carga 2 | Carga 3 |
|----------------|---------------|----------------|----------------|
| TT121 | 0,39% ± 0,01% | 1,38% ± 0,02% | 3,85% ± 0,03% |
| TT131 | 5,30% ± 0,05% | 15,19% ± 0,08% | 23,61% ± 0,09% |
| TT133 | 0,04% ± 0,00% | 1,10% ± 0,02% | 3,63% ± 0,04% |
| TT141 | 4,83% ± 0,04% | 5,70% ± 0,04% | 7,98% ± 0,05% |
| TT151 | 0,90% ± 0,01% | 1,65% ± 0,03% | 4,32% ± 0,04% |
| TT161 | 2,03% ± 0,02% | 2,07% ± 0,03% | 5,40% ± 0,03% |

Tabela 6.34: Bloqueio no estabelecimento para CT3, rede KL-15, factores de carga entre 1 e 3 e tipos de tabela TT121, TT131, TT133, TT141, TT151 e TT161 (PL3-RL5).

| | Tentado | | Estabelecido | |
|---|--------------|--------------|--------------|--------------|
| | PL3 | RL5 | PL3 | RL5 |
| 1 | 142334 ± 239 | 117865 ± 285 | 141424 ± 239 | 117748 ± 285 |
| 2 | 72641 ± 142 | 187454 ± 387 | 71970 ± 135 | 184545 ± 383 |
| 3 | 23413 ± 241 | 236716 ± 415 | 22941 ± 236 | 227167 ± 373 |

Tabela 6.35: Número de pedidos tentados versus estabelecidos, por esquema, para CT3, rede KL-15, factores de carga entre 1 e 3 e tabela TT121.

de 40% o bloqueio adicional é pequeno, mas à medida que o factor de carga aumenta o número de pedidos protegidos torna-se muito reduzido, como se pode verificar na Figura 6.12 (b). Em compensação com limiar de carga de 70% a percentagem de pedidos protegidos é sempre significativa mas o bloqueio adicional sofrido é também significativo.

Os gráficos na Figura 6.14 ilustram o bloqueio no estabelecimento para CT3, para as tabelas TT123, TT131 (PL3), TT134 (PL2), TT143, TT153 e TT163, para o factor de carga 1, 2 e 3, para a rede COST 239 e KL-15 respectivamente (intervalos de confiança nas Tabelas 6.37 e 6.38).

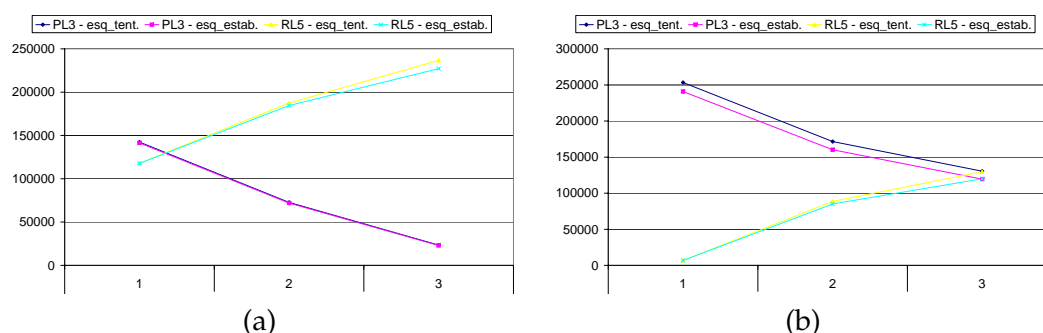


Figura 6.13: Número de pedidos tentados versus estabelecidos, por esquema, para CT3, em função do factor de carga (rede KL-15) - misto por carga da rede (PL3-RL5): (a) Limiar de carga = 40%. (b) Limiar de carga = 70%.

| | Tentado | | Estabelecido | |
|---|--------------|--------------|--------------|--------------|
| | PL3 | RL5 | PL3 | RL5 |
| 1 | 253309 ± 324 | 6890 ± 200 | 240852 ± 249 | 6789 ± 198 |
| 2 | 171509 ± 396 | 88585 ± 315 | 160156 ± 336 | 85105 ± 307 |
| 3 | 130466 ± 284 | 129663 ± 365 | 119464 ± 251 | 119914 ± 319 |

Tabela 6.36: Número de pedidos tentados versus estabelecidos, por esquema, para CT3, rede KL-15, factores de carga entre 1 e 3 e tabela TT141.

| Tipo de tabela | Carga 1 | Carga 2 | Carga 3 |
|----------------|---------------|----------------|----------------|
| TT123 | 0,00% ± 0,00% | 0,91% ± 0,03% | 5,89% ± 0,05% |
| TT131 | 1,00% ± 0,02% | 10,66% ± 0,09% | 18,74% ± 0,06% |
| TT134 | 0,00% ± 0,00% | 0,92% ± 0,03% | 5,91% ± 0,06% |
| TT143 | 0,03% ± 0,00% | 1,01% ± 0,03% | 6,01% ± 0,06% |
| TT153 | 0,00% ± 0,00% | 0,93% ± 0,02% | 5,89% ± 0,06% |
| TT163 | 0,01% ± 0,00% | 0,97% ± 0,03% | 5,93% ± 0,08% |

Tabela 6.37: Bloqueio no estabelecimento para CT3, rede COST 239, factores de carga entre 1 e 3 e tipos de tabela TT123, TT131, TT134, TT143, TT153 e TT163 (PL3-PL2).

A Figura 6.15 mostra o comprimento do CA para CT3, para as tabelas TT123, TT131 (PL3), TT134 (PL2) e TT143, para a rede KL-15 (intervalos de confiança na Tabela 6.39).

Os gráficos na Figura 6.16 mostram o número de pedidos tentados e o número de pedidos estabelecidos, por PL3 e por PL2, para CT3, para a rede KL-15, respectivamente para as tabelas TT123 e TT143 (intervalos de confiança nas Tabelas 6.40 e 6.41). De novo as TT123 (correspondente a limiar de carga de 40%) e TT143 (correspondente a limiar de carga de 70%) são os comportamentos extremos,

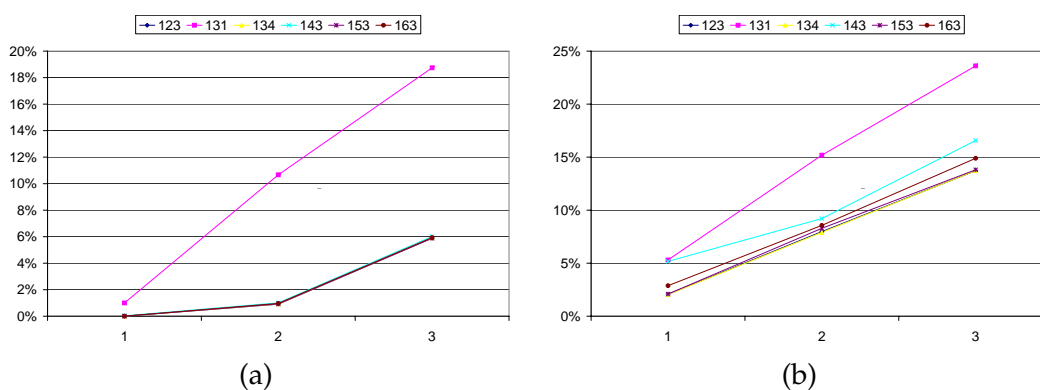


Figura 6.14: Bloqueio no estabelecimento para CT3, em função do factor de carga - “mono-esquema” versus misto por carga da rede (PL3-PL2): (a) Rede COST 239. (b) Rede KL-15.

| Tipo de tabela | Carga 1 | Carga 2 | Carga 3 |
|----------------|---------------|----------------|----------------|
| TT123 | 2,06% ± 0,03% | 7,98% ± 0,06% | 13,72% ± 0,08% |
| TT131 | 5,30% ± 0,05% | 15,19% ± 0,08% | 23,61% ± 0,09% |
| TT134 | 2,04% ± 0,03% | 7,90% ± 0,07% | 13,73% ± 0,07% |
| TT143 | 5,17% ± 0,04% | 9,21% ± 0,06% | 16,57% ± 0,08% |
| TT153 | 2,09% ± 0,03% | 8,29% ± 0,05% | 13,81% ± 0,07% |
| TT163 | 2,88% ± 0,03% | 8,56% ± 0,07% | 14,89% ± 0,07% |

Tabela 6.38: Bloqueio no estabelecimento para CT3, rede KL-15, factores de carga entre 1 e 3 e tipos de tabela TT123, TT131, TT134, TT143, TT153 e TT163 (PL3-PL2).

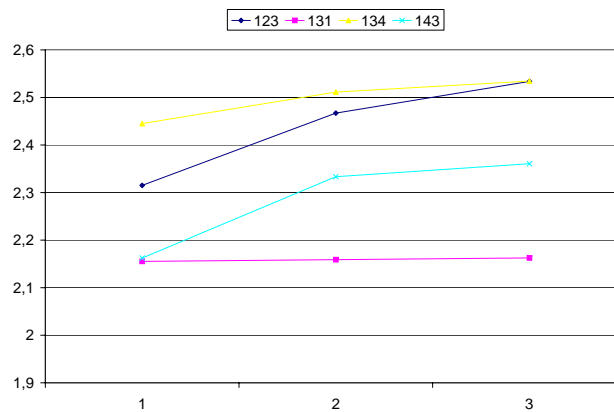


Figura 6.15: Número de ramos do CA, para CT3, em função do factor de carga (rede KL-15) - "mono-esquema" versus misto por carga da rede (PL3-PL2).

| Tipo de tabela | Carga 1 | Carga 2 | Carga 3 |
|----------------|---------------|---------------|---------------|
| TT123 | 2,315 ± 0,001 | 2,467 ± 0,001 | 2,534 ± 0,001 |
| TT131 | 2,155 ± 0,001 | 2,159 ± 0,001 | 2,163 ± 0,000 |
| TT134 | 2,445 ± 0,001 | 2,511 ± 0,001 | 2,534 ± 0,001 |
| TT143 | 2,162 ± 0,001 | 2,333 ± 0,001 | 2,361 ± 0,001 |

Tabela 6.39: Número de ramos do CA, para CT3, rede KL-15, factores de carga entre 1 e 3 e tipos de tabela TT123, TT131, TT134 e TT143 (PL3-PL2).

apresentando TT153 e TT163 (não representados nesta figura) comportamentos intermédios.

| | Tentado | | Estabelecido | |
|---|--------------|--------------|--------------|--------------|
| | PL2 | PL3 | PL2 | PL3 |
| 1 | 130136 ± 214 | 130063 ± 131 | 125405 ± 174 | 129425 ± 141 |
| 2 | 231019 ± 461 | 29076 ± 169 | 210588 ± 385 | 28761 ± 160 |
| 3 | 260125 ± 318 | 11 ± 9 | 224422 ± 282 | 11 ± 8 |

Tabela 6.40: Número de pedidos tentados versus estabelecidos, por esquema, para CT3, rede KL-15, factores de carga entre 1 e 3 e tabela TT123.

| | Tentado | | Estabelecido | |
|---|--------------|--------------|--------------|--------------|
| | PL2 | PL3 | PL2 | PL3 |
| 1 | 8069 ± 139 | 252130 ± 305 | 7271 ± 120 | 239478 ± 226 |
| 2 | 119564 ± 421 | 140531 ± 345 | 100628 ± 287 | 135504 ± 306 |
| 3 | 133070 ± 303 | 127060 ± 230 | 100391 ± 138 | 116622 ± 256 |

Tabela 6.41: Número de pedidos tentados versus estabelecidos, por esquema, para CT3, rede KL-15, factores de carga entre 1 e 3 e tabela TT143.

Da análise das Figuras 6.14, 6.15 e 6.16 é possível ver que a utilização de um esquema com partilha, de acordo com a carga, permitiu diminuir o bloqueio mas à custa do aumento do comprimento do CA. Os valores relativos de diminuição do bloqueio/aumento do comprimento do CA dependem do valor do limiar de carga escolhido, quanto maior for este último menor será o comprimento do CA mas também menor será a diminuição do bloqueio. Os resultados mostram portanto que a utilização de mecanismos de partilha para carga mais elevadas permitem obter as vantagens inerentes a estes esquemas (menor bloqueio) mas

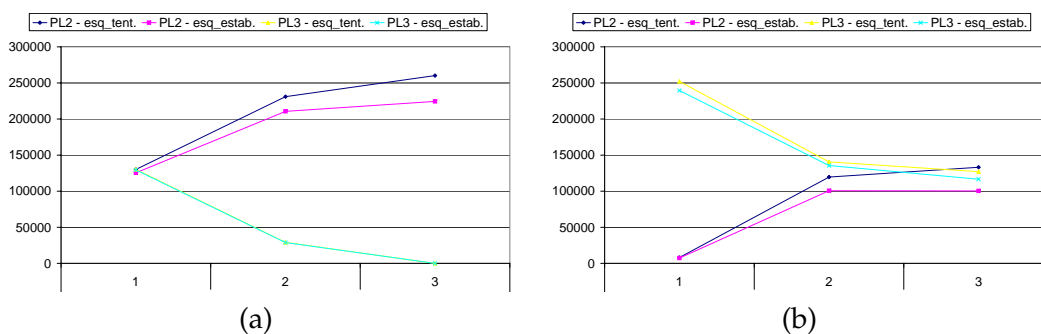


Figura 6.16: Número de pedidos tentados versus estabelecidos, por esquema, para CT3, em função do factor de carga (rede KL-15) - misto por carga da rede (PL3-PL2): (a) Limiar de carga = 40%. (b) Limiar de carga = 70%.

apenas à custa das respectivas desvantagens. Como se pode ver dos gráficos na Figura 6.16, as variações descritas anteriormente nos efeitos da utilização do esquema com partilha (em função dos limiares de carga usados), reflectem o número de pedidos que efectivamente usam esse esquema. Assim, com limiar de carga de 40% tentam estabelecer-se mais pedidos com partilha portanto os resultados em termos de bloqueio e de comprimento de CA estão mais próximos dos de PL2 “mono-esquema”. Com limiar de carga de 70% tentam-se menos pedidos com partilha e portanto os resultados estão mais próximos dos de PL3 “mono-esquema”.

Os gráficos na Figura 6.17 ilustram o bloqueio no estabelecimento para CT3, para as tabelas TT125, TT131 (PL3), TT132 (PG3), TT145, TT155 e TT165, para o factor de carga 1, 2 e 3, para a rede COST 239 e KL-15 respectivamente (intervalos de confiança nas Tabelas 6.42 e 6.43).

| Tipo de tabela | Carga 1 | Carga 2 | Carga 3 |
|----------------|---------------|----------------|----------------|
| TT125 | 0,27% ± 0,01% | 6,66% ± 0,07% | 14,45% ± 0,08% |
| TT131 | 1,00% ± 0,02% | 10,66% ± 0,09% | 18,74% ± 0,06% |
| TT132 | 0,27% ± 0,01% | 6,66% ± 0,07% | 14,44% ± 0,06% |
| TT145 | 0,29% ± 0,01% | 6,74% ± 0,08% | 14,45% ± 0,07% |
| TT155 | 0,28% ± 0,01% | 6,68% ± 0,07% | 14,44% ± 0,07% |
| TT165 | 0,28% ± 0,01% | 6,72% ± 0,06% | 14,45% ± 0,04% |

Tabela 6.42: Bloqueio no estabelecimento para CT3, rede COST 239, factores de carga entre 1 e 3 e tipos de tabela TT125, TT131, TT132, TT145, TT155 e TT165 (PL3-PG3).

Todos os gráficos apresentados nesta secção apresentam resultados apenas para CT3, no Apêndice D são apresentados os mesmos gráficos mas para CT1.

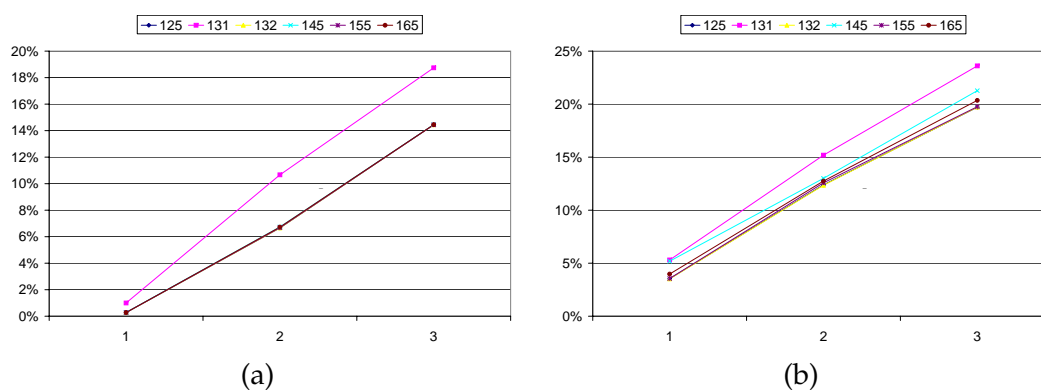


Figura 6.17: Bloqueio no estabelecimento para CT3, em função do factor de carga - “mono-esquema” versus misto por carga da rede (PL3-PG3): (a) Rede COST 239. (b) Rede KL-15.

| Tipo de tabela | Carga 1 | Carga 2 | Carga 3 |
|----------------|---------------|----------------|----------------|
| TT125 | 3,54% ± 0,04% | 12,39% ± 0,09% | 19,71% ± 0,07% |
| TT131 | 5,30% ± 0,05% | 15,19% ± 0,08% | 23,61% ± 0,09% |
| TT132 | 3,52% ± 0,04% | 12,35% ± 0,07% | 19,72% ± 0,09% |
| TT145 | 5,19% ± 0,04% | 13,00% ± 0,08% | 21,27% ± 0,07% |
| TT155 | 3,55% ± 0,03% | 12,58% ± 0,07% | 19,76% ± 0,08% |
| TT165 | 3,98% ± 0,03% | 12,75% ± 0,09% | 20,35% ± 0,09% |

Tabela 6.43: Bloqueio no estabelecimento para CT3, rede KL-15, factores de carga entre 1 e 3 e tipos de tabela TT125, TT131, TT132, TT145, TT155 e TT165 (PL3-PG3).

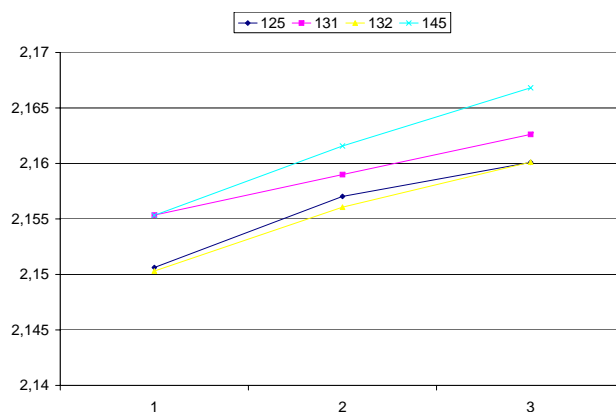


Figura 6.18: Número de ramos do CA, para CT3, em função do factor de carga (rede KL-15) - “mono-esquema” versus misto por carga da rede (PL3-PG3).

A Figura 6.18 mostra o comprimento do CA para CT3, para as tabelas TT125, TT131 (PL3), TT132 (PG3) e TT145, para a rede KL-15 (intervalos de confiança na Tabela 6.44).

| Tipo de tabela | Carga 1 | Carga 2 | Carga 3 |
|----------------|---------------|---------------|---------------|
| TT125 | 2,151 ± 0,001 | 2,157 ± 0,001 | 2,160 ± 0,001 |
| TT131 | 2,155 ± 0,001 | 2,159 ± 0,001 | 2,163 ± 0,000 |
| TT132 | 2,150 ± 0,001 | 2,156 ± 0,001 | 2,160 ± 0,001 |
| TT145 | 2,155 ± 0,001 | 2,162 ± 0,001 | 2,167 ± 0,001 |

Tabela 6.44: Número de ramos do CA, para CT3, rede KL-15, factores de carga entre 1 e 3 e tipos de tabela TT125, TT131, TT132 e TT145 (PL3-PG3).

Os gráficos na Figura 6.19 mostram o número de pedidos tentados e o número de pedidos estabelecidos, por PL3 e por PG3, para CT3, para a rede KL-15, respectivamente para as tabelas TT125 e TT145 (intervalos de confiança nas Tabelas 6.45 e 6.46). Mais uma vez as TT125 (correspondente a limiar de carga de 40%) e TT145 (correspondente a limiar de carga de 70%) são os comportamentos

extremos, apresentando TT155 e TT165 (não representados nesta figura) comportamentos intermédios.

| | Tentado | | Estabelecido | |
|---|--------------|--------------|--------------|--------------|
| | PG3 | PL3 | PG3 | PL3 |
| 1 | 130029 ± 225 | 130171 ± 114 | 121449 ± 133 | 129532 ± 126 |
| 2 | 230531 ± 510 | 29564 ± 216 | 198615 ± 431 | 29260 ± 214 |
| 3 | 260122 ± 320 | 14 ± 9 | 208847 ± 246 | 14 ± 8 |

Tabela 6.45: Número de pedidos tentados versus estabelecidos, por esquema, para CT3, rede KL-15, factores de carga entre 1 e 3 e tabela TT125.

| | Tentado | | Estabelecido | |
|---|--------------|--------------|--------------|--------------|
| | PG3 | PL3 | PG3 | PL3 |
| 1 | 7648 ± 134 | 252551 ± 345 | 6737 ± 133 | 239968 ± 276 |
| 2 | 116703 ± 338 | 143392 ± 402 | 88398 ± 236 | 137883 ± 364 |
| 3 | 132721 ± 347 | 127409 ± 247 | 87732 ± 201 | 117059 ± 248 |

Tabela 6.46: Número de pedidos tentados versus estabelecidos, por esquema, para CT3, rede KL-15, factores de carga entre 1 e 3 e tabela TT145.

Na Figura 6.18 é visível que na tabela TT125 a variação do comprimento do CA, relativamente à variação na tabela TT131 (PL3), não é significativa e é até benéfica para cargas baixas. No que concerne ao bloqueio (ver Figura 6.17) este é significativamente inferior a 131 (PL3) à custa do aumento da utilização de protecção global (mais marcante quando o limiar é 40%), como se pode ver nos gráficos da Figura 6.19. Em conclusão, nos testes efectuados, verificou-se que os esquemas mistos obtidos têm desempenhos, nas medidas estudadas, que se localizam entre os dos esquemas base, podendo os valores destes desempenhos

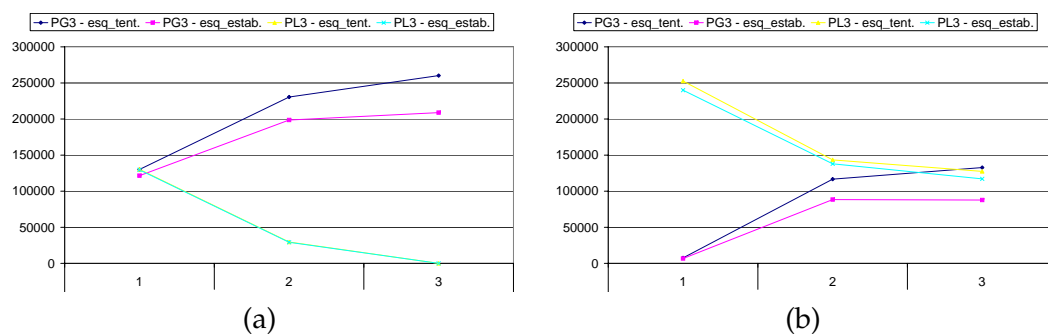


Figura 6.19: Número de pedidos tentados versus estabelecidos, por esquema, para CT3, em função do factor de carga (rede KL-15) - misto por carga da rede (PL3-PG3): (a) Limiar de carga = 40%. (b) Limiar de carga = 70%.

| <i>Condição</i> | CT0 | CT1 | CT2 | CT3 |
|-----------------|--------|------|------|------|
| <i>Sempre</i> | RL4[F] | R[] | P[] | P[] |

Tabela 6.47: #A.99 - Regras aplicadas para a etapa A.

serem aproximados a cada um dos esquemas base em função da parametrização através do limiar de carga. De notar que as diferenças observadas podem ter sido mais visíveis porque os esquemas base escolhidos apresentam diferenças significativas entre si na LB utilizada.

Notou-se que a amplitude de variação dos desempenhos em função dos limites de carga é dependente da rede usada, tendo sido mais expressiva na rede KL-15 do que na rede COST 239. Notou-se ainda que o desempenho do esquema misto parece particularmente influenciado pelo esquema usado em situações de carga mais elevada. Esta situação, visível particularmente na rede COST 239, pode ocorrer devido ao número total de pedidos bloqueados ser muito mais reduzido em situações de carga leve, o que pode contribuir também para explicar a diferença entre redes indicada anteriormente.

A ideia desta experiência foi criar diferentes situações de sazonalidade e observar se adaptar o esquema de recuperação à situação de sazonalidade trazia melhorias, o que parece verificar-se, embora por vezes de forma não muito expressiva. Os valores escolhidos para o factor de carga conduzem a situações de grande sobrecarga. Será importante repetir o estudo para outros valores do factor de carga.

6.5.5 Várias tabelas e combinação de condições

Nesta secção vai criar-se um exemplo das capacidades do sistema proposto, que ilustra algumas das suas funcionalidades.

As tabelas que constituem o sistema são as Tabela 6.47 para a etapa A e as Tabelas 6.48, 6.49 e 6.50 para a etapa B.

Em #A.99 (Tabela 6.47) indicou-se que CT2 e CT3 precisam de protecção e CT1 e CT0 não precisam. Foi escolhido RL4 para CT0, como exemplo de uma opção final, tomada na etapa A. Para tal, é indicado que uma escolha posterior para este CT apenas poderá ser efectuada na Etapa F (correspondente a uma situação de falha que afecte este pedido).

A Tabela 6.48 (#B.10) só define esquemas para CT0 e CT3. As opções para CT0 serão ignoradas porque uma opção final foi já tomada anteriormente. No respeitante ao CT3 a combinação de dois esquemas (um com pequeno bloqueio

| <i>Condição</i> <i>Comp. do CA</i> | CT0 | CT1 | CT2 | CT3 |
|---------------------------------------|--------|-----|-----|--------|
| ≤ 2 | RG1[F] | | | PG3[] |
| > 2 | RL4[F] | | | PL2[F] |

Tabela 6.48: #B.10 – De acordo com o comprimento do CA.

| <i>Condição</i> <i>R/P Carga</i> | CT0 | CT1 | CT2 | CT3 |
|-------------------------------------|-----------------|--------|--------|--------|
| <i>P</i> | <i>Leve</i> | PL2[F] | PL3[F] | PL3[F] |
| | <i>Não leve</i> | | PG3[F] | PG3[F] |
| <i>R</i> | <i>Leve</i> | RL5[F] | | |
| | <i>Não leve</i> | | | |

Tabela 6.49: #B.20 - De acordo com o modelo de recuperação e carga na rede.

| <i>Condição</i> | CT0 | CT1 | CT2 | CT3 |
|-----------------|--------|--------|--------|--------|
| <i>Sempre</i> | RL4[F] | RG2[F] | PG3[F] | PG1[F] |

Tabela 6.50: #B.99 - Regras aplicadas por omissão para a etapa B.

no estabelecimento e outro com número de ramos do CA pequeno) é conseguida através da escolha de um ou outro em função do valor estimado para o comprimento do CA. A escolha do esquema PG3 para comprimentos estimados de CA curtos foi propositadamente deixada como não final para permitir mostrar a possibilidade de redefinir um esquema numa tabela posterior da mesma etapa.

A Tabela 6.49 (#B.20) mostra a conjugação de dois tipos de restrições (em relação à carga da rede e escolha de eventuais características do esquema) para determinar esquemas para CT1, CT2 e CT3. A escolha dos esquemas para CT2 foi efectuada com base na observação que a escolha de protecção global para cargas mais elevadas consegue caminhos CA mais curtos com um custo reduzido em termos de bloqueio. O mesmo foi seguido para CT3, embora neste caso se tenha usado PL3 para cargas baixas simplesmente para demonstrar que mais que dois esquemas podem ser usados na mesma classe. De notar que os dois esquemas propostos se vão sobrepor à escolha não final de PG3 na tabela anterior. Finalmente para CT1 é apresentado a escolha do esquema PL2 para cargas leves assumindo que anteriormente tivesse sido escolhido protecção para este CT (no exemplo proposto este teste será sempre falso pelo que este esquema nunca será usado).

A Tabela 6.50 (#B.99 - tabela por omissão para a etapa B) será apenas usada na

| Factor de carga | 1 | 1.1 | 1.2 | 1.3 | 1.4 | 1.5 | 1.7 | 2 |
|------------------|-------|-------|-------|-------|-------|-------|-------|-------|
| λ_{high} | 0.215 | 0.237 | 0.258 | 0.280 | 0.301 | 0.323 | 0.366 | 0.430 |
| λ_{low} | 0.065 | 0.072 | 0.078 | 0.085 | 0.091 | 0.098 | 0.111 | 0.130 |
| Média harmónica | 0.1 | 0.11 | 0.12 | 0.13 | 0.14 | 0.15 | 0.17 | 0.2 |

Tabela 6.51: Valores para λ_{high} e λ_{low} (exemplo do sistema).

prática para CT1 quando a carga é elevada. Isto ocorre porque foram efectuadas escolhas finais anteriormente para todas as condições possíveis de CT0, CT2 e CT3. No entanto esta tabela permitiria garantir valores por omissão aceitáveis para todos os CT mesmo que tenham existido alguns erros ou incoerências na definição das tabelas anteriores (situação que se torna mais provável se o número total de tabelas for elevado).

Um pedido de CT3, com este conjunto de tabelas, pode usar um de três esquemas de recuperação PL2, PL3 ou PG3. As tabelas definidas e o seu encadeamento pode ser tal que os pedidos de um determinado CT possam usar um esquema de entre qualquer um dos vários definidos, dependendo das condições verificadas.

Nas experiências anteriores, como foram utilizados os mesmos factores de carga para todos os esquemas de recuperação, em determinados esquemas foram obtidos bloqueios elevados. Vamos nesta última experiência considerar, além da carga nominal aumentos de carga apenas de 10%, 20%, 30%, 40%, 50%, 70% e 100%. A rede será sujeita a carga flutuante à semelhança do que foi feito na experiência da secção anterior. Para cada um vários factores de carga os valores para λ_{high} e λ_{low} (e média harmónica respectiva) são apresentados na Tabela 6.51 (o valor da carga nominal (λ_{base}) foi mantido o das experiências anteriores). Os valores dos parâmetros indicados na Tabela 6.30 foram também usados nesta experiência (excepto o factor de carga, como referido anteriormente).

As tabelas deste exemplo foram construídas tendo em consideração as necessidades das classes tal como apresentadas na Tabela 6.15 da Secção 6.5.2. No entanto mais do que tentar criar as escolhas que se supõe mais adequadas para cada CT a preocupação primordial foi criar um exemplo que mostrasse algumas das capacidades da metodologia apresentada. Isto levou por exemplo à escolha de esquemas mistos para CT1 e CT2 cujos objectivos possivelmente poderiam ser atingidos tão bem ou melhor pela utilização de esquemas base. Já para o CT3 a escolha de um esquema misto pode ser mais facilmente justificada pela intenção de considerar diferentes objectivos simultaneamente (mesmo assim pode ser discutível a escolha dos esquemas efectivamente usados e das condições para alternar entre eles).

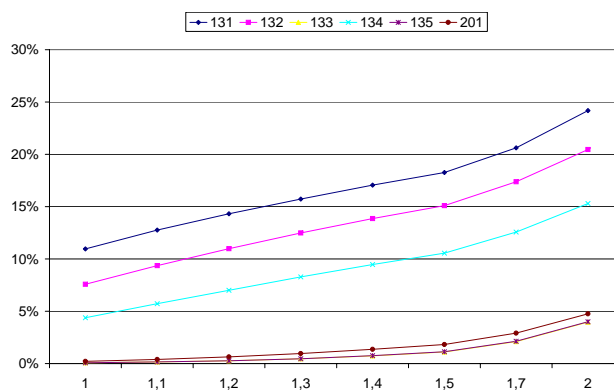


Figura 6.20: Bloqueio no estabelecimento para CT1, em função do factor de carga (rede KL-15) - “mono-esquema” versus exemplo.

Os resultados obtidos com este exemplo, por cada um dos CT, serão comparados com os resultados obtidos com as várias tabelas “mono-esquema” (sujeitas às mesmas condições de carga) correspondentes aos esquemas utilizados por esse CT. Assim, nesta experiência, assim como aconteceu na secção anterior, recorreremos novamente às tabelas “mono-esquema” correspondentes a PL3, PG3, RL5 e PL2 identificadas, respectivamente, com os tipos de tabela TT131, TT132, TT133 e TT134. Necessitamos também da tabela “mono-esquema” correspondentes a RG2 que identificamos com o tipo de tabela TT135. Obtiveram-se resultados com todas estas tabelas “mono-esquema”, sujeitas a carga flutuante, para os novos factores de carga. O exemplo em estudo é identificado com o tipo de tabela TT201. O número total de corridas distintas (adicionais) resultante para esta experiência foi $800 = ([1(\text{tabelas}) \times 8(\text{factores de carga}) + 4(\text{tabelas "mono-esquema"}) \times 6(\text{factores de carga}) + 1(\text{tabela "mono-esquema"}) \times 8(\text{factores de carga})] \times 10 (\text{réplicas}) \times 2(\text{redes}))$.

Todos os resultados apresentados nesta secção correspondem à utilização da rede KL-15 (no Apêndice D mostram-se os mesmos gráficos apresentados nesta secção mas relativos à rede COST 239) e foram obtidos quando, na Tabela 6.49 (#B.20), se usou o valor 40% para limiar de carga. O gráfico na Figura 6.20 mostra a percentagem de pedidos rejeitados no estabelecimento para CT1, para vários factores de carga, para os tipos de tabela TT131 (PL3), TT132 (PG3), TT133 (RL5), TT134 (PL2), TT135 (RG2) e TT201 (intervalos de confiança nas Tabelas 6.52 e 6.53).

O acréscimo do bloqueio do exemplo (tipo de tabela TT201) em relação à tabela TT133 (RL5) e TT135 (RG2) poderá provavelmente explicar-se porque os recursos requeridos por CT2 e CT3, nas tabelas TT133 e TT135 serem inferiores aos requeridos em TT201 (apesar deste efeito ser minorado pelo BCM usado).

| Tipo de tabela | Carga 1 | Carga 1,1 | Carga 1,2 | Carga 1,3 |
|----------------|----------------|----------------|----------------|----------------|
| TT131 | 10,96% ± 0,06% | 12,76% ± 0,06% | 14,32% ± 0,04% | 15,73% ± 0,05% |
| TT132 | 7,58% ± 0,05% | 9,36% ± 0,04% | 10,98% ± 0,06% | 12,49% ± 0,04% |
| TT133 | 0,08% ± 0,00% | 0,16% ± 0,00% | 0,26% ± 0,00% | 0,46% ± 0,01% |
| TT134 | 4,39% ± 0,04% | 5,73% ± 0,05% | 7,00% ± 0,05% | 8,28% ± 0,04% |
| TT135 | 0,08% ± 0,00% | 0,16% ± 0,00% | 0,27% ± 0,01% | 0,46% ± 0,01% |
| TT201 | 0,21% ± 0,00% | 0,39% ± 0,01% | 0,64% ± 0,01% | 0,96% ± 0,01% |

Tabela 6.52: Bloqueio no estabelecimento para CT1, rede KL-15, factores de carga entre 1 e 1,3 e tipos de tabela TT131, TT132, TT133, TT134, TT135 e TT201.

| Tipo de tabela | Carga 1,4 | Carga 1,5 | Carga 1,7 | Carga 2 |
|----------------|----------------|----------------|----------------|----------------|
| TT131 | 17,06% ± 0,06% | 18,26% ± 0,05% | 20,62% ± 0,08% | 24,18% ± 0,06% |
| TT132 | 13,86% ± 0,06% | 15,10% ± 0,06% | 17,38% ± 0,06% | 20,46% ± 0,07% |
| TT133 | 0,74% ± 0,02% | 1,11% ± 0,01% | 2,10% ± 0,03% | 3,97% ± 0,04% |
| TT134 | 9,46% ± 0,04% | 10,56% ± 0,05% | 12,57% ± 0,05% | 15,29% ± 0,06% |
| TT135 | 0,76% ± 0,01% | 1,13% ± 0,01% | 2,13% ± 0,03% | 4,00% ± 0,05% |
| TT201 | 1,36% ± 0,02% | 1,83% ± 0,02% | 2,91% ± 0,03% | 4,76% ± 0,05% |

Tabela 6.53: Bloqueio no estabelecimento para CT1, rede KL-15, factores de carga entre 1,4 e 2 e tipos de tabela TT131, TT132, TT133, TT134, TT135 e TT201.

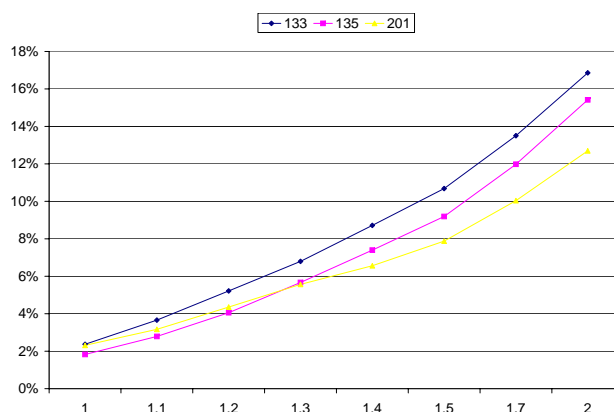


Figura 6.21: Desconexão na falha para CT1, em função do factor de carga (rede KL-15) - “mono-esquema” versus exemplo.

O gráfico na Figura 6.21 mostra a percentagem de desconexão na falha para CT1, para vários factores de carga, para os tipos de tabela TT133 (RL5), TT135 (RG2) e TT201 (intervalos de confiança na Tabela 6.54). Perante falhas o exemplo (tipo de tabela TT201) tem um comportamento melhor do que as tabelas TT133 e TT135, o que se pode justificar pelo facto de ser apenas CT1 a precisar de LB adicional para recuperação ao contrário do que acontece nas tabelas TT133 e TT135.

| | TT133 | TT135 | TT201 |
|-----|----------------|----------------|----------------|
| 1 | 2,37% ± 0,14% | 1,83% ± 0,15% | 2,31% ± 0,18% |
| 1,1 | 3,66% ± 0,16% | 2,79% ± 0,12% | 3,17% ± 0,19% |
| 1,2 | 5,22% ± 0,26% | 4,05% ± 0,20% | 4,36% ± 0,13% |
| 1,3 | 6,79% ± 0,28% | 5,67% ± 0,30% | 5,57% ± 0,26% |
| 1,4 | 8,72% ± 0,30% | 7,40% ± 0,32% | 6,56% ± 0,26% |
| 1,5 | 10,68% ± 0,28% | 9,19% ± 0,23% | 7,88% ± 0,21% |
| 1,7 | 13,50% ± 0,57% | 11,98% ± 0,27% | 10,04% ± 0,33% |
| 2 | 16,86% ± 0,43% | 15,42% ± 0,39% | 12,69% ± 0,17% |

Tabela 6.54: Desconexão na falha para CT1, rede KL-15, factores de carga entre 1 e 2 e tipos de tabela TT133, TT135 e TT201.

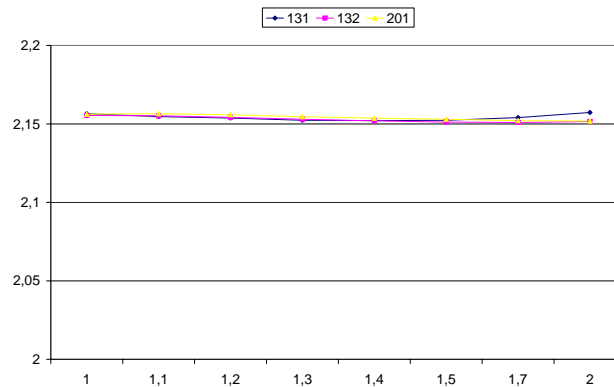


Figura 6.22: Número de ramos do CA para CT2, em função do factor de carga (rede KL-15) - "mono-esquema" versus exemplo.

| | TT131 | TT132 | TT201 |
|-----|---------------|---------------|---------------|
| 1 | 2,157 ± 0,001 | 2,155 ± 0,001 | 2,156 ± 0,001 |
| 1,1 | 2,155 ± 0,001 | 2,155 ± 0,001 | 2,156 ± 0,001 |
| 1,2 | 2,154 ± 0,001 | 2,154 ± 0,001 | 2,156 ± 0,001 |
| 1,3 | 2,152 ± 0,001 | 2,153 ± 0,001 | 2,155 ± 0,001 |
| 1,4 | 2,152 ± 0,001 | 2,152 ± 0,001 | 2,154 ± 0,001 |
| 1,5 | 2,152 ± 0,001 | 2,151 ± 0,001 | 2,153 ± 0,001 |
| 1,7 | 2,154 ± 0,001 | 2,151 ± 0,001 | 2,152 ± 0,001 |
| 2 | 2,157 ± 0,002 | 2,152 ± 0,001 | 2,152 ± 0,001 |

Tabela 6.55: Número de ramos do CA, para CT2, rede KL-15, factores de carga entre 1 e 2 e tipos de tabela TT131, TT132 e TT201.

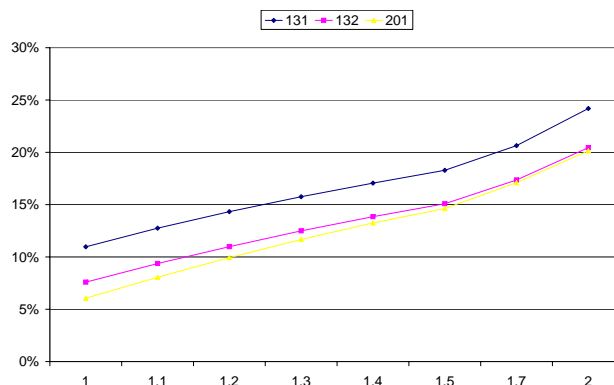


Figura 6.23: Bloqueio no estabelecimento para CT2, em função do factor de carga (rede KL-15) - "mono-esquema" versus exemplo.

A Figura 6.22 mostra o comprimento médio dos CA para CT2, para as tabelas TT131 (PL3), TT132 (PG3) e TT201 (intervalos de confiança na Tabela 6.55). A Figura 6.23 mostra a percentagem de pedidos rejeitados no estabelecimento para CT2, para vários factores de carga, para os tipos de tabela TT131, TT132, e TT201 (intervalos de confiança na Tabela 6.56). No exemplo (tipo de tabela TT201), em termos de comprimento médio de CA observa-se (Figura 6.22) um desempenho semelhante aos dos esquemas base. O bloqueio no estabelecimento é inferior a qualquer dos esquemas base, possivelmente consequência de existir mais LB disponível para CT2 no exemplo do que nas tabelas base.

| | TT131 | TT132 | TT201 |
|-----|----------------|----------------|----------------|
| 1 | 10,97% ± 0,05% | 7,59% ± 0,04% | 6,06% ± 0,03% |
| 1,1 | 12,75% ± 0,04% | 9,38% ± 0,05% | 8,06% ± 0,03% |
| 1,2 | 14,33% ± 0,05% | 10,98% ± 0,04% | 9,93% ± 0,04% |
| 1,3 | 15,75% ± 0,07% | 12,49% ± 0,05% | 11,67% ± 0,06% |
| 1,4 | 17,06% ± 0,05% | 13,85% ± 0,05% | 13,25% ± 0,05% |
| 1,5 | 18,29% ± 0,06% | 15,08% ± 0,05% | 14,63% ± 0,07% |
| 1,7 | 20,64% ± 0,05% | 17,35% ± 0,05% | 17,10% ± 0,04% |
| 2 | 24,19% ± 0,06% | 20,45% ± 0,05% | 20,15% ± 0,06% |

Tabela 6.56: Bloqueio no estabelecimento para CT2, rede KL-15, factores de carga entre 1 e 2 e tipos de tabela TT131, TT132 e TT201.

A Figura 6.24 mostra a percentagem de pedidos rejeitados no estabelecimento para CT3, para vários factores de carga, para os tipos de tabela TT131 (PL3), TT132 (PG3), TT134 (PL2) e TT201 (intervalos de confiança na Tabela 6.57). A Figura 6.25 mostra o comprimento médio dos CA para CT3, para as tabelas TT131, TT132, TT134 e TT201 (intervalos de confiança na Tabela 6.58).

Da Figura 6.24 observa-se que CT3, no exemplo (tipo de tabela TT201), apre-

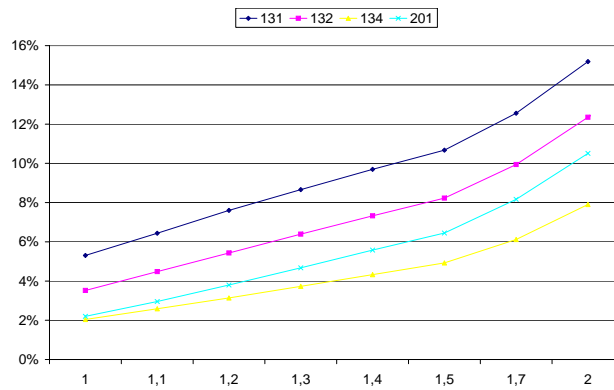


Figura 6.24: Bloqueio no estabelecimento para CT3, em função do factor de carga (rede KL-15) - "mono-esquema" versus exemplo.

| | TT131 | TT132 | TT134 | TT201 |
|-----|----------------|----------------|---------------|----------------|
| 1 | 5,30% ± 0,05% | 3,52% ± 0,04% | 2,04% ± 0,03% | 2,20% ± 0,02% |
| 1,1 | 6,44% ± 0,04% | 4,48% ± 0,03% | 2,59% ± 0,02% | 2,96% ± 0,02% |
| 1,2 | 7,60% ± 0,05% | 5,43% ± 0,05% | 3,14% ± 0,02% | 3,80% ± 0,05% |
| 1,3 | 8,66% ± 0,07% | 6,39% ± 0,05% | 3,73% ± 0,03% | 4,67% ± 0,05% |
| 1,4 | 9,70% ± 0,07% | 7,32% ± 0,05% | 4,33% ± 0,02% | 5,58% ± 0,04% |
| 1,5 | 10,67% ± 0,08% | 8,23% ± 0,05% | 4,92% ± 0,04% | 6,45% ± 0,05% |
| 1,7 | 12,56% ± 0,06% | 9,94% ± 0,07% | 6,12% ± 0,06% | 8,17% ± 0,06% |
| 2 | 15,19% ± 0,08% | 12,35% ± 0,07% | 7,90% ± 0,07% | 10,51% ± 0,08% |

Tabela 6.57: Bloqueio no estabelecimento para CT3, rede KL-15, factores de carga entre 1 e 2 e tipos de tabela TT131, TT132, TT134 e TT201.

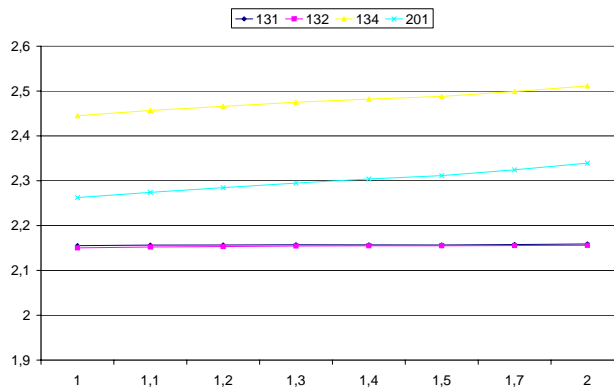


Figura 6.25: Número de ramos do CA para CT3, em função do factor de carga (rede KL-15) - "mono-esquema" versus exemplo.

| | TT131 | TT132 | TT134 | TT201 |
|-----|---------------|---------------|---------------|---------------|
| 1 | 2,155 ± 0,001 | 2,150 ± 0,001 | 2,445 ± 0,001 | 2,262 ± 0,001 |
| 1,1 | 2,157 ± 0,001 | 2,152 ± 0,001 | 2,457 ± 0,001 | 2,274 ± 0,001 |
| 1,2 | 2,157 ± 0,001 | 2,153 ± 0,001 | 2,466 ± 0,001 | 2,285 ± 0,001 |
| 1,3 | 2,157 ± 0,001 | 2,154 ± 0,001 | 2,475 ± 0,001 | 2,295 ± 0,001 |
| 1,4 | 2,157 ± 0,001 | 2,155 ± 0,001 | 2,482 ± 0,001 | 2,304 ± 0,001 |
| 1,5 | 2,157 ± 0,001 | 2,155 ± 0,001 | 2,488 ± 0,001 | 2,311 ± 0,001 |
| 1,7 | 2,158 ± 0,001 | 2,155 ± 0,001 | 2,499 ± 0,001 | 2,324 ± 0,001 |
| 2 | 2,159 ± 0,001 | 2,156 ± 0,001 | 2,511 ± 0,001 | 2,339 ± 0,001 |

Tabela 6.58: Número de ramos do CA, para CT3, rede KL-15, factores de carga entre 1 e 2 e tipos de tabela TT131, TT132, TT134 e TT201.

senta bloqueios entre os bloqueios de PL2 e os de PG3 e nitidamente melhores do que os de PL3. Também os comprimentos do CA do exemplo apresentam valores intermédios, embora por ordem inversa ou seja, menores do que PL2 e maiores do que PL3 e PG3.

A Figura 6.26 mostra o número de pedidos que tentam ser estabelecidos e os efectivamente estabelecidos por PL2, PL3 e PG3, para CT3, no exemplo apresentado (tipo de tabela TT201), e respectivos intervalos de confiança na Tabela 6.59. Esta figura mostra que embora o número de pedidos tratados por cada esquema não difira de forma muito significativa, excepto para as cargas mais elevadas, existe uma diferença significativa no bloqueio no estabelecimento por classe. Assim embora o número de pedidos tentados estabelecer por PL2 (associado a CA de comprimento longo) seja ligeiramente crescente com a carga o número de pedidos efectivamente estabelecidos por esquema não é crescente, sendo visivelmente decrescente para carga mais elevadas. O comportamento em termos de pedidos tentados de PL3 e PG3 são aproximadamente simétricos, o que é de esperar uma vez que a escolha de um ou outro se efectua com a carga da rede para CA de comprimentos mais curtos. No entanto existe uma diferença significativa entre estes no caso de pedidos efectivamente estabelecidos, sendo o bloqueio no estabelecimento de PL3 (usado quando a carga é leve) praticamente nulo, enquanto PG3 apresenta sempre bloqueio visível em todos os factores de carga (mas que ao contrário do bloqueio de PL2 não cresce significativamente com a carga).

Como conclusões finais deste exemplo podemos referir que em geral os comportamentos dos esquemas mistos se encontram próximos dos esquemas base que os constituem mas não são, excepto em poucos casos, melhores que estes em todas as medidas simultaneamente. Como tal a sua utilização parece mais justificada quando se pretende atingir resultados correspondentes a uma combinação dos resultados dos esquemas base. Quando apenas é considerado um objectivo

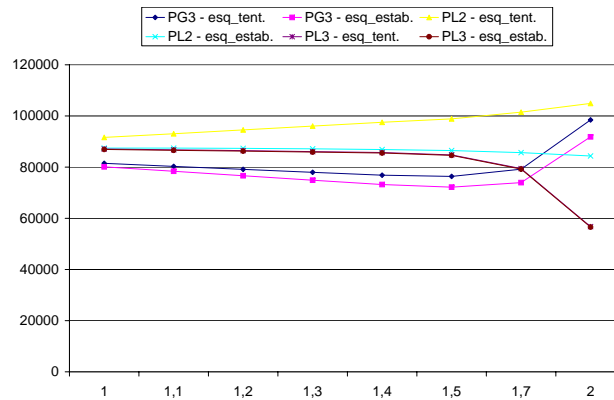


Figura 6.26: Número de pedidos tentados versus estabelecidos, por esquema, para CT3, em função do factor de carga no exemplo (rede KL-15).

| | Tent | | | Est | | |
|-----|-------------|--------------|-------------|-------------|-------------|-------------|
| | PG3 | PL2 | PL3 | PG3 | PL2 | PL3 |
| 1 | 81494 ± 152 | 91614 ± 205 | 87092 ± 141 | 80112 ± 137 | 87445 ± 163 | 86913 ± 145 |
| 1,1 | 80299 ± 177 | 93051 ± 158 | 86763 ± 191 | 78394 ± 177 | 87444 ± 113 | 86580 ± 191 |
| 1,2 | 79123 ± 166 | 94555 ± 211 | 86453 ± 162 | 76641 ± 154 | 87339 ± 126 | 86270 ± 163 |
| 1,3 | 77971 ± 151 | 96045 ± 235 | 86075 ± 126 | 74899 ± 146 | 87152 ± 172 | 85888 ± 124 |
| 1,4 | 76851 ± 138 | 97538 ± 222 | 85711 ± 218 | 73198 ± 139 | 86879 ± 169 | 85516 ± 215 |
| 1,5 | 76374 ± 150 | 98871 ± 250 | 84787 ± 169 | 72173 ± 152 | 86488 ± 188 | 84598 ± 168 |
| 1,7 | 79193 ± 225 | 101488 ± 257 | 79428 ± 232 | 73954 ± 230 | 85677 ± 167 | 79233 ± 233 |
| 2 | 98440 ± 243 | 104902 ± 241 | 56753 ± 237 | 91840 ± 265 | 84355 ± 134 | 56572 ± 237 |

Tabela 6.59: Número de pedidos tentados versus estabelecidos, por esquema, para CT3, rede KL-15, factores de carga entre 1 e 2 e tipo de tabela TT201.

de particular importância para um CT esse objectivo poderá provavelmente ser atingido com mais facilidade usando um esquema base com melhor desempenho nessa característica.

6.6 Análise final

Foi apresentada uma análise simulacional que pretende ser o ponto de partida para o estudo da escolha dinâmica do esquema de recuperação em função dos critérios de serviço desejados. Uma limitação óbvia da abordagem seguida (simulação ao fluxo) é a impossibilidade de medir directamente algumas características importantes como o *jitter* e o tempo de recuperação. Esta limitação poderá ser levantada por posterior estudo através de simulação ao pacote.

Foi efectuada a análise em duas redes distintas, em geral foram obtidos resultados consistentes nas duas redes utilizadas. Foram apresentados resultados para várias experiências que representaram no total 5840 corridas de simulação (1680

para a experiência da Secção 6.5.1, 1440 para a experiência da Secção 6.5.2, 480 para a experiência da Secção 6.5.3, 1440 para a experiência da Secção 6.5.4 e 800 para a experiência da Secção 6.5.5). O tempo de cada corrida variou entre poucos minutos e as dezenas de horas, tendo sido o tempo total de simulação cerca de 500 dias de CPU.

A metodologia desenvolvida permite-nos criar um esquema de recuperação que mais se aproxime de um compromisso aceitável das características desejadas. Constatou-se a grande importância da diferenciação da recuperação por CT e observou-se o comportamento da selecção dinâmica de esquemas de recuperação específicos para os vários CT.

É necessária também uma análise mais aprofundada ao compromisso implícito entre os ganhos potenciais (consumo de recursos, tempo/atraso de recuperação, tempos de estabelecimento dos caminhos, ...) e a complexidade adicional envolvida (complexidade de sinalização no estabelecimento e da recuperação, complexidade de notificação das falhas, complexidade na sinalização da partilha, peso de computação usando o novo sistema de recuperação).

7 Conclusões

Neste capítulo será apresentado um resumo do trabalho realizado, do que foi conseguido e do que ficou ainda por fazer.

7.1 Síntese

O trabalho desenvolvido focou-se na recuperação de falhas em redes MPLS. Para este fim, foram estudados os termos e conceitos de recuperação (em redes em geral e em redes MPLS), e diferentes esquemas que podem ser usados para atingir este fim. A caracterização destes esquemas, particularmente na sua capacidade de fornecer garantias de QoS e optimização na utilização de recursos, levou à compreensão das diferenças entre estes e da dificuldade de um só esquema suportar de forma óptima todos os objectivos divergentes associados a diferentes tipos de tráfego.

O estudo em particular de esquemas de recuperação que permitem partilhar Largura de Banda de Protecção (LBP) levou ao desenvolvimento de potenciais melhorias para alguns esquemas propostos na literatura. Assim, foi verificado para um esquema de protecção local como optimizá-lo para aumentar a partilha de LBP sem necessidade de utilizar mais informação. Foi verificado também como tirar partido da informação presente localmente na reserva de LBP para efectuar a de forma exacta. Para os esquemas de protecção global, foram analisadas e validadas alternativas para uma componente importante destes esquemas (a determinação de caminhos disjuntos de custo agregado mínimo com custos duais nos arcos).

A verificação de que potencialmente nenhum esquema isoladamente pode por si só garantir as características mais desejadas pelos gestores da rede para todo o tipo de tráfego, levou ao desenvolvimento de um sistema de recuperação que integre diversos esquemas de recuperação. O sistema proposto permite, para escolher qual o esquema a aplicar em cada situação, utilizar o conhecimento sobre o comportamento de cada esquema de recuperação, sobre as necessidades associadas e objectivos desejados para o tráfego a tratar e sobre o estado actual da rede.

Foi desenvolvido um simulador que permitiu a validação experimental do desempenho do sistema proposto e verificar até que ponto esse sistema forneceria vantagens em relação à utilização de esquemas mais simples. A implementação do simulador foi um trabalho complexo, e implicou a tomada de algumas decisões de projecto que podem eventualmente afectar a capacidade de generalização dos resultados obtidos. Contudo considerou-se que as opções tomadas não diminuíram significativamente a sua validade.

Foram enunciados e testados (recorrendo ao simulador desenvolvido) diversos conjuntos de hipóteses, que permitiram verificar os pressupostos de aplicabilidade que estiveram na origem do sistema proposto. Foi apresentada também uma experiência que apresenta a conjugação de utilização de diferentes capacidades suportadas pelo sistema, como validação da utilização de configurações complexas para gestão simultânea de vários tipos de tráfego.

7.2 Principais resultados do trabalho realizado

Da análise efectuada à recuperação em redes MPLS resultou uma resenha com um estudo comparativo dos esquemas em função das suas características mais relevantes. Este trabalho (cuja versão original aparece em Jorge e Gomes, 2005a) deu origem às seguintes publicações Jorge e Gomes (2005b, 2006b). Deste trabalho foi visível a diferença entre os esquemas propostos, e sugerido que qualquer esquema isolado dificilmente poderá ser adequado para todos os tipos de tráfego.

O estudo dos algoritmos de protecção com partilha de LBP para recuperação local levaram ao desenvolvimento de diversos melhoramentos para o algoritmo apresentado em Kodialam e Lakshman (2001). Os melhoramentos efectuados foram analisados por diversos testes, tendo a descrição destes testes e resultados aparecido em Jorge e Gomes (2006c), e sido posteriormente publicada em Jorge e Gomes (2007). Os resultados destes testes mostram uma melhoria de desempenho significativa em relação ao original.

Os trabalhos respeitantes aos algoritmos para determinação de caminhos disjuntos de custo agregado mínimo com custos duais nos arcos foram validados por experiências e testes, descritos em Gomes *et al.* (2006a) e submetido para publicação em Gomes *et al.* (2006b). Os resultados dos testes mostram melhorias em termos de tempo (em relação a implementações usando Programação Linear Inteira) com obtenção de soluções óptimas para quase todas as situações (e mesmo nos casos em que não foi possível atingir ou demonstrar o óptimo, os resultados encontram-se comprovadamente próximos deste). Trabalho associado, respeitante à sinalização para LBP, apareceu em Gomes *et al.* (2005).

O trabalho correspondente ao sistema de recuperação proposto foi apresentado numa conferência internacional e publicado em Jorge e Gomes (2006a). O desenvolvimento de um simulador para o mesmo encontra-se descrito no Capítulo 5 e Apêndice A, indicando as opções de projecto e implementação tomadas. Os resultados correspondentes à implementação prática do esquema são publicados pela primeira vez neste trabalho (no Capítulo 6). Dos resultados obtidos, foi possível constatar a diferença entre o comportamento dos diversos esquemas, e que alguns fornecem resultados mais apropriados para determinados tipos de tráfego. É também visível a capacidade de obter através do sistema proposto “esquemas mistos” que combinam as propriedades de esquemas individuais, através da escolha do esquema adequado em situações distintas.

7.3 Perspectivas de trabalho futuro

Embora o trabalho apresentado nesta dissertação tenha sido longo, muitas áreas de investigação abertas por ele ficaram por explorar, por manifesta falta de tempo para o fazer. De entre estes, podem elencar-se:

- Relativamente ao trabalho de partilha de LBP, algumas questões relacionadas com os algoritmos usados ficaram por validar, como:
 - No respeitante ao algoritmo apresentado em Kodialam e Lakshman (2003), averiguar a possibilidade de aumentar o grau de partilha e diminuir o seu peso computacional (ambos os problemas foram verificados pelo simulador construído). Tal pode ser tentado (por exemplo) usando o algoritmo descrito na Secção 3.3.2.
 - Verificar o funcionamento prático de partilha de informação de LBP por algoritmos locais e globais. No simulador construído, é possível utilizar esquemas diferentes dentro do mesmo CT, e nesta situação poderá ser vantajoso usar uma estrutura comum para permitir maior partilha.
- Relativamente ao simulador construído, algumas limitações foram identificadas (para além da opção básica de implementação ao pacote, cujo trabalho preliminar se encontra descrito no Apêndice B):
 - Expandir o simulador para suportar tanto falhas de ramos como de nós (o que implica implementar o mesmo suporte também nos esquemas de recuperação).

- Considerar outros tipos de fluxos de tráfego (como os presentes na Internet) e distribuições alternativas para a geração de falhas, para avaliar se os resultados obtidos são similares.
 - Ampliar o número de esquemas de recuperação base implementados, e otimizar os actuais. Por exemplo, pode considerar-se usar um método de encaminhamento com outros objectivos, em vez do algoritmo *Dijkstra*, considerando a ocupação para distribuir a carga pela rede (embora este tipo de optimização possa tornar mais difícil a análise de resultados).
 - Suporte para falhas múltiplas (necessários por exemplo para suportar SRLG). Embora não tenham sido considerados neste trabalho para poder usar algoritmos pré definidos de partilha de LBP, o simulador deve permitir simular este tipo de situações.
 - Como aplicação prática de algo já previsto no sistema proposto, permitir a simulação usando conjuntos de tabelas para as situações de Falha e Optimização.
- Finalmente, no respeitante ao próprio sistema de recuperação proposto, algumas melhorias possíveis foram identificadas, incluindo:
 - Optimizar o processo de construção das tabelas, por exemplo facilitando a construção de conjuntos coerentes de tabelas simples.
 - Alternativamente, pode-se considerar a geração automatizada de tabelas para determinados tipos de tráfego, por exemplo através de avaliação multicritério dos resultados da simulação para permitir escolher a melhor combinação de esquemas para determinadas condições de tráfego e objectivos dos CT. Se for usado um número limitado de esquemas por CT pode mesmo considerar testar-se os resultados de todas as combinações de alternativas criadas usando métodos geradores.
 - Fazer uma análise mais aprofundada ao compromisso implícito entre os ganhos potenciais e a complexidade adicional envolvida (complexidade de sinalização no estabelecimento e da recuperação, complexidade de notificação das falhas, complexidade na sinalização da partilha, peso de computação usando o novo sistema de recuperação).

Mesmo na ausência destas melhorias, o sistema produzido ainda deverá ser usado em testes adicionais para avaliar efectivamente com outras redes e situações de tráfego, os limites dos compromissos que é possível obter entre os dife-

rente tipos de tráfego e a complexidade adicional do sistema para os suportar na prática.

A Estrutura do programa de simulação

Neste apêndice são descritas algumas características do código que compõe o simulador construído (complementando mas não substituindo a descrição presente Capítulo 5, nomeadamente incluindo pormenores julgados inadequados para o corpo do trabalho). Para além das componentes nucleares, componentes de suporte para esquemas, componentes de armazenamento e componentes de recolha de estatísticas, já descritas na Secção 5.4 do Capítulo 5 acrescentam-se aqui também algumas componentes auxiliares, que correspondem a classes que servem de suporte em diversos pontos do programa de simulação.

Nas figuras apresentadas nas próximas secções foram incluídos alguns objectos que não foram utilizados na simulação do novo sistema de recuperação mas que fazem parte do código desenvolvido neste trabalho (por exemplo, uma outra utilização do código foi feita quando se efectuou o estudo de um algoritmo de encaminhamento para protecção local com reserva exacta apresentado no Capítulo 3 nas Secções 3.1 e 3.2).

A.1 Estruturas para armazenamento de dados

Houve a necessidade de definir diversas classes, cuja função principal é guardar informação, mas que possuem também funções de manipulação sobre essa informação. As classes `VECNOs` e `VECARCOs` foram definidas para guardar informação sobre os nós e os arcos da rede, respectivamente. Para guardar informação de um CA e do CR global ou local foram desenvolvidas as classes `CAMINHO`, `GRUPOCAMINHOS`, `CAMINHOACTIVO`, `CAMINHORECUPERACOASINGLE` e `CAMINHOSRECUPERACAO`. Foram ainda necessárias as classes `PEDIDO`, `GRUPOPEDIDOS` e `CLASSETRAFEGO` para guardar informação sobre um pedido, um grupo de pedidos e sobre um CT, respectivamente.

A.1.1 Informação dos Nós da rede e dos Arcos da rede

Para cada um dos nós da rede guarda-se apenas o seu nome e uma forma de aceder ao módulo OMNeT++ que lhe está associado (ver classe `No` na Figura A.1).

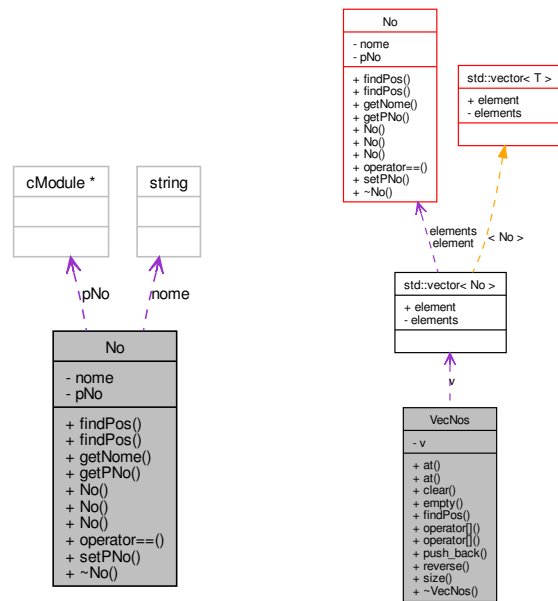


Figura A.1: Diagrama de herança – No e VecNos.

Para guardar informação de todos os nós da rede recorreu-se, na implementação da classe `VecNos` (ver Figura A.1), ao *container* `VECTOR` do *Standard Template Library*.

Um arco é representado por dois índices que guardam respectivamente a posição do nó origem e do nó destino no vector de nós em que os nós origem e destino estão representados. Cada arco guarda também um apontador para esse vector de nós, ou seja, cada arco necessita que o vector de nós usado lhe fique associado. Para cada arco é guardada também a indicação do seu estado de actividade. Para cada um dos arcos da rede guarda-se ainda uma forma de aceder às *gates*, dos módulos OMNeT++ associados ao nó origem e ao nó destino deste arco (ver classe `ARCO` na Figura A.2). À semelhança do que foi feito para o caso da informação de todos os nós da rede também aqui se recorreu, na classe `VECARCOS` (ver Figura A.2), ao *container* `VECTOR` para guardar a informação dos vários arcos da rede.

A.1.2 Informação sobre Caminhos

A classe base usada na definição dos vários tipos de caminhos foi a classe `CAMINHO`, cujo diagrama de herança é ilustrado na Figura A.3. Para um objecto do tipo `CAMINHO` é guardado o vector de nós que o constitui, uma indicação da

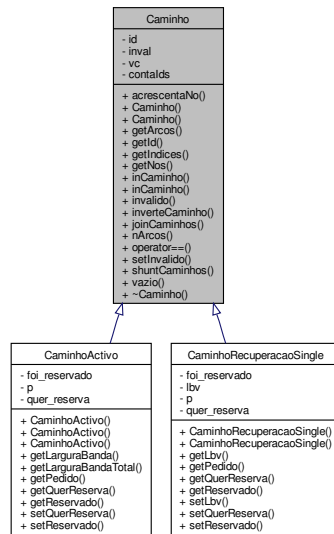


Figura A.3: Diagrama de herança – CAMINHO.

validade do caminho e a sua identificação. As funções desta classe permitem realizar várias operações sob um caminho existente, tais como acrescentar-lhe um nó, acrescentar-lhe um caminho, verificar se um ramo faz parte dele, inverter os ramos que o constituem ou determinar o seu número de ramos. Foi definida também a classe GRUPOCAMINHOS que permite armazenar vários objectos CAMINHO, destinada ao armazenamentos do ou dos CR. Inclui funções que permitem operações de manipulação dos atributos tais como adicionar um CAMINHO a GRUPOCAMINHOS. Estes objectos são utilizados no armazenamento de caminhos que não possuem um pedido associado.

Já para o armazenamento do CA e do CR com um pedido associado foram definidas as classes CAMINHOACTIVO e CAMINHOSRECUPERACAO. A Figura A.4 mostra a classe CAMINHOACTIVO que foi desenvolvida para armazenar a informação relativa aos CA. Esta figura mostra também a relação de herança de propriedades, já ilustrada na Figura A.3, e algumas dependências. Para um CA é armazenada adicionalmente informação sobre o pedido (classe PEDIDO) que lhe deu origem (que inclui informação sobre a LB do mesmo), indicação do desejo de querer ser reservado e indicação da realização da reserva. Possui várias funções, algumas permitem obter informação dos seus atributos, outras permitem a atribuição de valores a esses atributos. Já um objecto da classe CAMINHOSRECUPERACAO (ilustrada na Figura A.5) permite armazenar informação sobre os CR (um ou vários) com o pedido associado. Para tal essa classe foi implementada

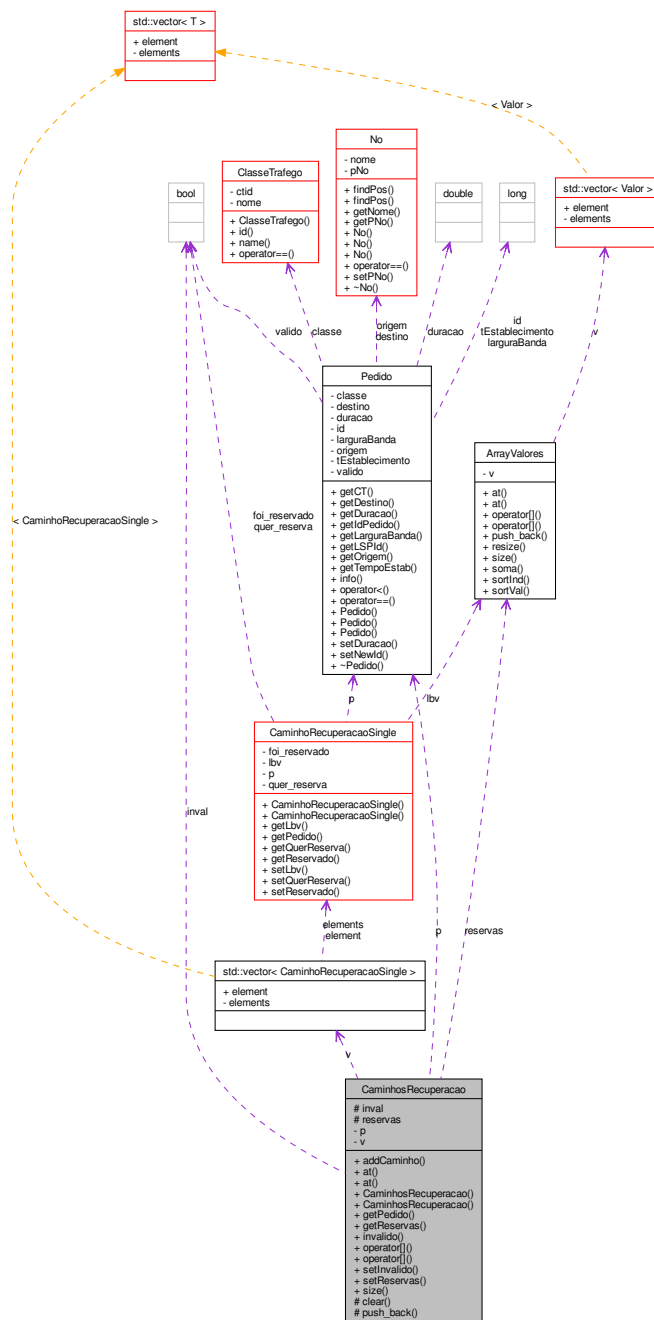


Figura A.5: Diagrama de classes – CAMINHOSRECUPERACAO.

A.1.3 Informação dos pedidos

Para armazenar informação de um pedido foi definida a classe PEDIDO. Os conteúdos desta classe incluem os nós origem e destino do tráfego associado ao pedido, a duração do pedido, uma identificação do pedido, a largura de banda necessária para o pedido, o instante em que o pedido é estabelecido e o CT do pedido. De entre as funções definidas para manipulação desses atributos, destaca-se a função para fornecer uma identificação de um LSP correspondente a esse pedido, cujo valor identifica o pedido. Uma classe auxiliar, designada de CLASSETRAFEGO foi usada para guardar o índice do CT usado.

Para permitir pesquisas eficientes foi usado o *template* HASH com os objectos PEDIDO, ARCO, CLASSETRAFEGO e INDGRUPOPEDIDOS, permitindo assim criar *hash tables* (*tabelas de hash*) indexadas por cada um destes tipos.

A.1.4 Informação da rede e do seu estado

As estruturas INFORMACAOREDE e REGISTARESERVAS armazenam informação respectivamente sobre a rede e sobre as reservas, dando conteúdo semântico às estruturas básicas apresentadas anteriormente. Assim, em INFORMACAOREDE (ver Figura A.6) é guardada alguma informação básica, como a definição dos ARCOSDAREDE e NOSDAREDE, de forma a que os diferentes componentes do simulador que necessitam de aceder a essa informação o possam fazer de uma forma uniforme. Informação global sobre as reservas na rede é gerida pela classe REGISTARESERVAS, que pretende simular um comportamento semelhante ao de reservas usando RSVP (a implementação actual não implementa, porém, todas as capacidades do RSVP).

A.2 Componentes nucleares

O suporte para a definição do programa de simulação foi conseguido através do conjunto de classes ilustrado na Figura A.7. O núcleo do programa de simulação é definido no módulo simples com o nome MODULOBASETABELAS (criado na fase de inicialização pelo OMNeT++). O módulo simples é implementado em C++ e recorre à biblioteca de classes do OMNeT++ (derivado da classe CSIMPLEMODULE). O comportamento do modelo é implementado na função HANDLEMESSAGE, do módulo simples, que é invocada, pelo *kernel* da simulação, sempre que surge um acontecimento. Além da função referida, foram também redefinidas no módulo simples a função INITIALIZE e a função FINISH. Também estas funções

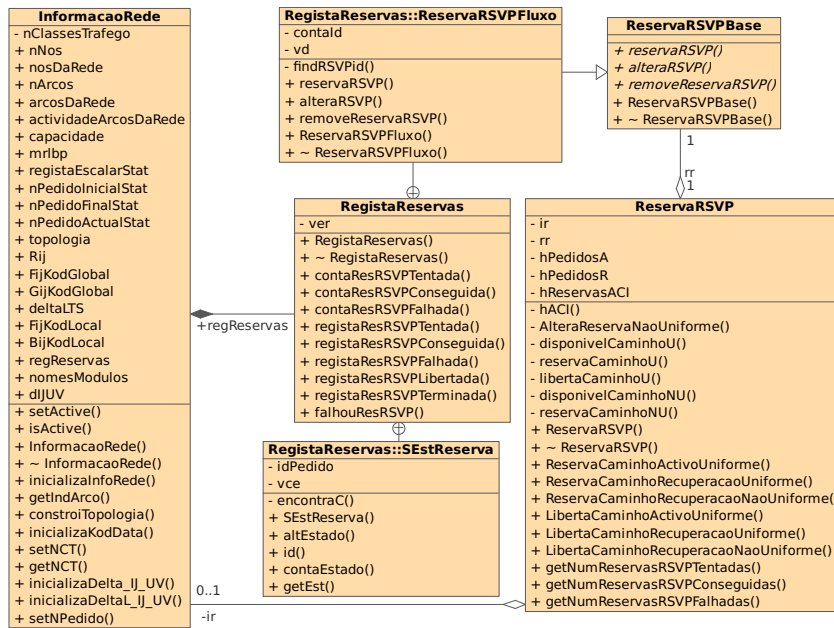


Figura A.6: Diagrama de classes – INFORMACAOREDE, REGISTARESERVAS e RESERVARSVP.

são invocadas pelo *kernel* da simulação, a primeira na fase de inicialização e a segunda quando a simulação termina com sucesso.

Foi efectuada a definição de cinco tipos de mensagens (instâncias da classe *CMESSAGE*), *MSGBASE*, as suas especializações *MSGPEDIDO*, *MSGFALHA* e *MSGGERADOR* e uma especialização desta última *MSGGERADORHIGHLOW*.

As instâncias destas classes, visíveis mais pormenorizadamente na Figura A.8, são geridas directamente pelo *kernel* do OMNeT++ (sendo usadas por este para representar acontecimentos). Por necessitarem de seguir uma estrutura muito rígida, estas classes não são codificadas directamente, mas sim descritas numa linguagem própria no ficheiro *MODULOBASE.MSG*. Foi utilizado o *script* *OPP_MSGC*, fornecido com o OMNeT++, para traduzir a definição das mensagens em C++ (resultando os ficheiros *MODULOBASE_M.H* e *MODULOBASE_M.CC*).

A.2.1 Implementação da máquina de estados

A estrutura da máquina de estados (utilizada para registar em que estado se encontra cada pedido) é definida no objecto *REGISTAESTADOSTABELAS*. Este objecto faz uso do objecto *ESCOLHEESQUEMAS* que por sua vez vai buscar as tabelas para determinar o novo estado da máquina de estados. *REGISTAESTADOSTABELAS* faz ainda uso do objecto *REGINDICECAMINHOS* (usado para armazenamento de informação) e do objecto *RESERVARSVP* (apresentado na Figura A.6). Sempre que

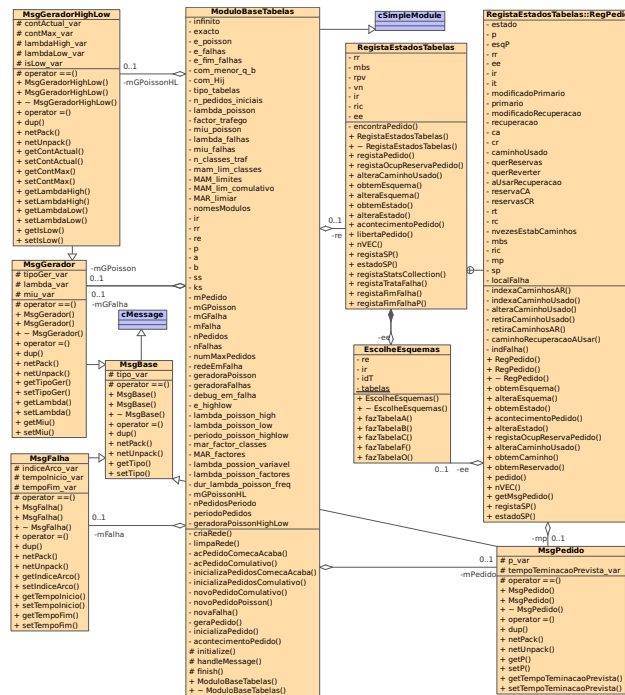


Figura A.7: Diagrama de classes – componentes nucleares.

um pedido é gerado é-lhe associado uma máquina de estados no MODULOBASE-TABELAS através do REGISTAESTADOS-TABELAS. O objecto REGISTAESTADOS-TABELAS permite escolher entre os diferentes estados (seleccionando o código apropriado de acordo com o estado em que o pedido se encontra). Um pedido passa para o próximo estado (dentro da máquina de estados) quando é invocada a função ACONTECIMENTO-PEDIDO dessa classe.

REGISTAESTADOS-TABELAS gere a informação de todos os pedidos que estão na rede. Após associar a cada pedido uma máquina de estados, coloca o pedido na lista de pedidos que têm máquinas de estado associadas (pedidos em execução). Perante um acontecimento, REGISTAESTADOS-TABELAS identifica qual o pedido afectado e desencadeia o tratamento em REGISTAESTADOS-TABELAS::REG-PEDIDO. Finalmente, o método REG-PEDIDO::ACONTECIMENTO-PEDIDO trata de executar o código correspondente ao pedido em particular (ou seja, efectivamente fazer as transições de estado para esse pedido).

Perante uma falha é necessário saber quais os pedidos que estão a usar o arco que falhou no caminho usado, porque estes são os pedidos que vão ser sujeitos a tratamento (recuperação). Para esta função são utilizadas *tabelas de hash* em REGISTAESTADOS-TABELAS para, por exemplo, a partir de um pedido obter todos os arcos usados por esse pedido ou a partir de uma arco obter todos os pedidos

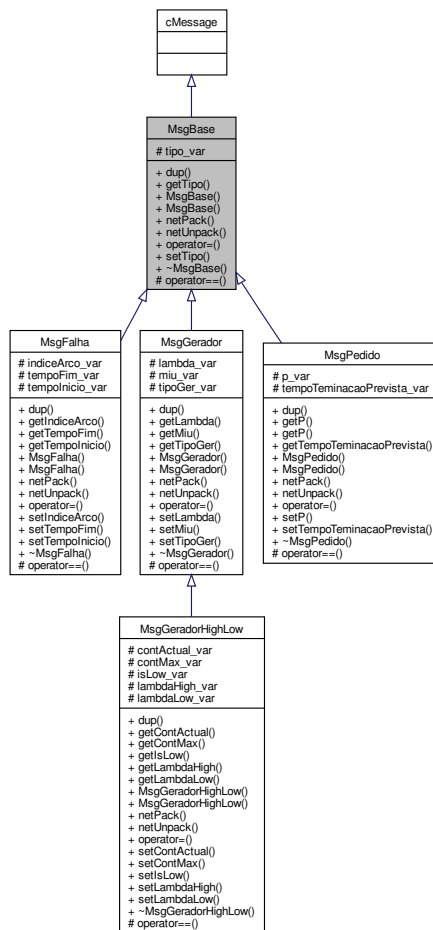


Figura A.8: Diagrama de herança – classe MsgBASE.

que estão indexados nesse arco.

Para o registo de ocupações associadas aos caminhos, foi definida a classe `REGISTAESTADOSTABELAS::REGINDICECAMINHOS` que regista os índices para os caminhos. Esta classe indexa caminhos de forma a saber quais os arcos que está a usar, e ao mesmo tempo indexar por arco, ou seja, perante uma arco saber quais os caminhos e quais os pedidos que o estão a utilizar. Um caminho pode representar um *caminho usado*, um CA ou um CR. O *caminho usado* corresponde ao caminho onde o tráfego flui no momento (pode ser o CA, parte do CA e um CR, ou apenas um CR).

A classe `REGISTAINDICECAMINHOS` serve para indexar caminhos associados a pedidos, ou seja ligar a pedidos os caminhos e ligar de volta os arcos aos pedidos. Regista os caminhos de determinado pedido (CA, CR e *caminho usado*) e por

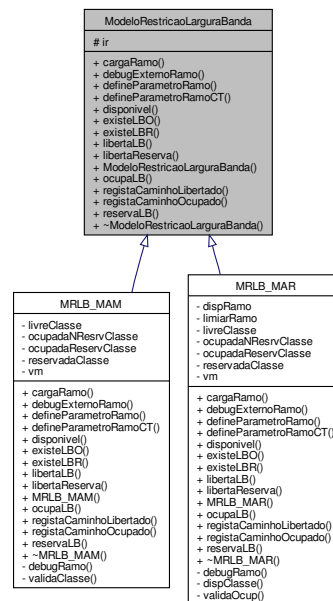


Figura A.9: Diagrama de herança – MODELORESTRICAO LARGURABANDA.

outro lado quais os pedidos que usam determinado arco (e para que tipo de caminho). Com este registo passa a ser possível a partir de um arco ou de um caminho obter informação dos pedidos e dos tipos de caminhos. Isto é necessário nomeadamente para tratamento das falhas, porque quando existe uma falha é preciso tratar de um conjunto de pedidos que são afectados por ela. Este código permite obter a partir de uma falha quais são os pedidos afectados (lista de pedidos afectados). Esta lista vai ser utilizada para colocar os pedidos no estado de falha que vai ser por sua vez utilizado para desencadear a chamada as tabelas de escolha em termos de tratamento de falhas.

A.2.2 Componentes usados na implementação dos BCM

Nesta secção apresentam-se as classes que suportam os diferentes *Bandwidth Constraints Model* (BCM) do simulador. Estes sistemas gerem as restrições de LB – designadas por *Bandwidth Constraint* (BC) – entre diversos CT. Num sistema com DS-TE é indispensável a utilização de pelo menos um BCM. Com DS-TE a cada CT está associada uma ou mais BC. Um BCM define o número máximo de BC e a que CT se aplica cada BC.

Foram implementados dois BCM, o *Maximum Allocation Bandwidth Constraints Model* (MAM) e o *Max Allocation with Reservation Bandwidth Constraints Model*

(MAR). O modelo *Russian Dolls Bandwidth Constraints Model* (RDM) não oferece isolamento entre diferentes CT quando não for utilizada preempção, pelo que não pode ser considerado para implementação, uma vez que no modelo usado não foi implementada preempção. Estes modelos são utilizados tanto na determinação dos caminhos como na reserva (apresentando funções específicas para reservar ou para ocupar espaço sem reserva).

Nos testes apresentados, foi usado o modelo MAR em todos os ramos. Este modelo foi escolhido devido à sua grande eficiência em termos de partilha de LB pelas várias CT, porque fornece isolamento entre os vários CT em situações de sobrecarga) e porque funciona razoavelmente bem sem preempção (ver Ash, 2005).

Os objectos associados aos BCM são os objectos MRLB_MAM e MRLB_MAR, respectivamente para os modelos MAM e MAR (ver Figura A.9). Estes objectos gerem um registo de BC e guardam informação sobre qual a ocupação ou reserva efectuada nos ramos, de forma a poder verificar que estas são cumpridas e dar informação acerca de disponibilidades.

A.2.3 Estatísticas

O objectivo principal do modelo de simulação implementado é a obtenção de parâmetros estatísticos que expressem os resultados do comportamento ao longo de cada corrida. O componente estatísticas fornece as funções de recolha de informação (chamadas sempre que adequado pelos componentes nucleares), e algumas funções de consolidação da informação (chamadas no fim da simulação para apresentar a informação consolidada).

Registar os resultados da simulação

O OMNeT++ permite registar os resultados da simulação de duas formas distintas, através do que é designado saída de vectores (valores vectoriais) e saída de escalares (valores escalares).

Quando os resultados são registados através da saída de vectores, os resultados são registados como séries temporais num ficheiro, registando o comportamento transitório de toda a corrida de simulação (desta forma uma grande quantidade de dados será registada durante uma corrida). O registo desta informação é conseguida através do uso de objectos do tipo `cOUTVECTOR` (no simulador este objecto foi encapsulado dentro de uma estrutura `OUTVEC`). Todos os objectos `cOUTVECTOR` (`OUTVEC`) registam a informação num ficheiro comum, a definir no

ficheiro de configuração do *run*. A classe `KODSTATS` possui os vários atributos `OUTVEC` usados pelo simulador.

Cada uma destas informações é armazenada no ficheiro imediatamente após recolha, gerando ficheiros de grande dimensão. Para facilitar o seu tratamento, estes ficheiros são manipulados e transformados (a partir de ficheiros com uma estrutura simples) em tabelas de uma base de dados (em que cada pedido apenas aparece uma vez) e armazenada de forma cumulativa (somando os valores para cada medida). Esta informação consolidada é finalmente preparada para análise recolhendo das tabelas apenas uma fracção dos dados (nos testes efectuados foram recolhidos resultados cumulativos com intervalo de 1000 pedidos entre si). Assim, torna-se possível (se desejado) analisar a evolução dos valores sem ter de lidar com todos os resultados (à custa de perder alguma resolução na análise evolução dos resultados, mas sem perder precisão nos mesmos).

No entanto, se se pretender registar apenas um valor por corrida de simulação, podemos recorrer a classes que armazenam (e agregam) a informação enquanto a simulação decorre e registam o resultado apenas no final da simulação (designados de resultados escalares). Para suportar esta recolha e agregação foram desenvolvidos os objectos `TIMEDVALUE`, `TIMEDVALUECL` e `VECTIMEVALUECL` descritos na Secção 5.4.5 do Capítulo 5. Estas estruturas permitem recolher valores de uma série, com a característica adicional de armazenarem também a duração durante a qual determinado valor se manteve (desta forma pode ser recolhido automaticamente uma média da série ponderada por tempo). Tal como para os resultados vectoriais, todos os resultados escalares partilham de um mesmo ficheiro de resultados, cujo nome é definido no ficheiro de configuração *ini*.

Estas classes permitem suportar a recolha de informação variada, mas esta informação deve ser recolhida em diversos pontos durante a execução do simulador. Para definir a informação efectivamente a recolher foram definidas diversas especializações de uma classe básica de recolha de informação, designada por `MBSTATBASE` (ver Figura A.10). As diferentes especializações foram criadas para permitir recolher informação de forma diferente sem alterar no restante o simulador. Deve ser notado, porém, que nos exemplos apresentados neste trabalho foi sempre usada a especialização `KSTATSGENERICO`. Para poder aceder à especialização activa da classe de estatísticas, foi definido um `STATSSINGLETON`, que pode ser usado por qualquer classe para aceder às funções de recolha de informação.



Figura A.10: Diagrama de herança – classe MBSTATBASE.

A.3 Componentes específicos do novo sistema de recuperação

O novo sistema de recuperação proposto é baseado em Tabelas que incluem um conjunto de condições e acções (escolha total ou parcial do esquema) a serem tomadas se as condições se verificarem. As classes definidas para o suporte das tabelas são `TABELAMATCHING`, que representa as tabelas efectivas, e `IMBASE` (e suas derivadas, visíveis na Figura A.11). As especializações de `IMBASE` servem para representar as condições das tabelas do novo sistema. A sua especialização `MATCHOBJECTOFUNCIONAL` e especializações desta representam todas as condições excepto as relativas ao teste do CT que são representadas pela classe `INPUTMATCH` (uma outra especialização da classe `IMBASE`). `TABELAMATCHING` é um objecto que inclui a classe `CELULAMATCHING`, que guarda as condições para essa célula (para um elemento da tabela) e as acções correspondentes ao verificar (*matching*) dessas condições. As condições para essa célula são armazenadas num vector de elementos do tipo `IMBASE` e as acções correspondentes são armazenadas no atributo `OUTRES` do tipo `INFORMACAO TABELA`.

A classe `INFORMACAO TABELA` serve para representar os resultados esperados do *matching* de uma célula. `INFORMACAO TABELA` faz uso da estrutura `INFORMACAO ESQUEMA RECUPERACAO` que possui um conjunto de funções que lhe permitem obter informação sobre as características do esquema de recuperação escolhido (modelo, âmbito ou etiqueta do esquema de recuperação), armazenadas nos atributos. As `CELULAMATCHING` são organizadas na `TABELAMATCHING` em linhas através da classe `LINHAMATCHING`.

As classes cujos nomes se iniciam por MOF são especializações sucessivas de `MATCHOBJECTOFUNCIONAL` associadas a um teste específico (condição para activação da acção). Assim `MOF_CARGAREDE`, `MOF_CARGAREDELEVE`, `MOF_CARGAREDELEVEG` e `MOF_CARGAREDEPESADA` verificam o nível de carga na rede. `MOF_LOCALFALHA`, `MOF_LOCALFALHA_LA1`, `MOF_LOCALFALHA_LA2`, `MOF_LOCALFALHA_LA3` verificam onde ocorreu a falha. E `MOF_TAMANHO LSP`, `MOF_TAMANHO LSP CURTO` e `MOF_TAMANHO LSP LONGO` verificam a dimensão do LSP mínimo possível de estabelecer para esse pedido. A função `MATCH` definida em `IMBASE` e redefinida em todas as suas especializações, devolve o valor verdadeiro se a condição correspondente a esse objecto for verificada.

O funcionamento do esquema é gerido pela classe `ESCOLHE ESQUEMA`. Esta classe possui um vector de vectores chamado `TABELAS` que armazena objectos `TABELAMATCHING`. Estes objectos são inicializados pela função `INICIALIZA TABELAS` da classe `INIC INFORMACAO TABELAS`. Uma chamada à função `INICIALIZA TABELAS` faz com que todas as tabelas do sistema, que necessitem ser usada nessa simula-

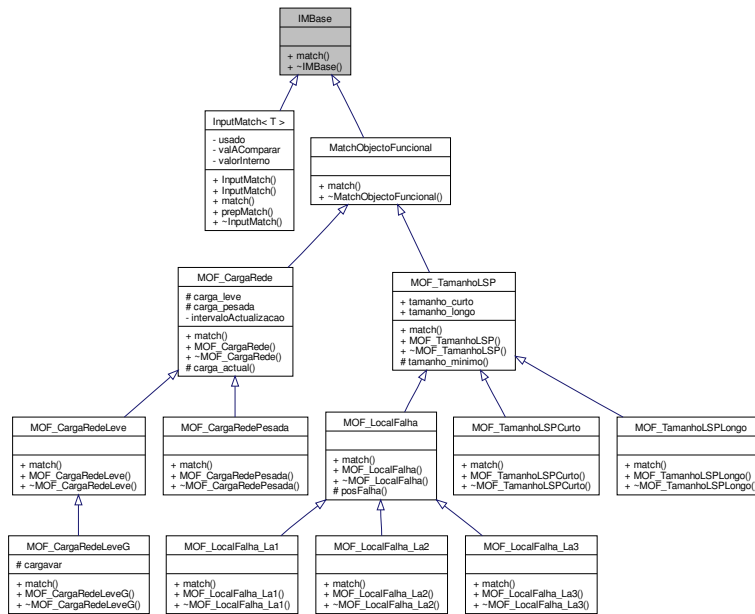


Figura A.11: Diagrama de herança – classe IMBase e derivadas.

ção, sejam instanciadas, criando uma ou várias instâncias de TABELAMATCHING.

Para inicialização do simulador, a função INICIALIZAÍNFOMACAOETIQUETAS, da classe INICINFORMACAOETIQUETAS, define todas as etiquetas de todos os esquemas de recuperação e associa-lhes o seu âmbito e método de recuperação. Esta informação é posteriormente usada pela classe TRADUZESQRECUPERACAO. O objecto INICINFORMACAOESQUEMAS traduz as etiquetas dos esquemas, usadas nas tabelas, nas classes que os suportam (especializações de ESQUEMARECUPERACAO).

A.4 Componentes associados aos esquemas de recuperação

Para suportar os diversos esquemas de recuperação implementados foram definidas as classes da Figura A.12.

ESQREENCGLOBAL foi a classe implementada para suportar o esquema de reenaminhamento global sem reserva. A implementação mais simples das funções virtuais puras de ESQUEMARECUPERACAO ocorre nesta classe. DETERMINACAMINHOS utiliza o algoritmo de *Dijkstra* para determinar o caminho mais curto entre a origem e o destino que será usado como CA. O CR obtido por essa função aparece como inválido (o que significa que não foi determinado). DETERMINACAMINHORECUPERACONAFALHA calcula o caminho mais curto entre a origem e

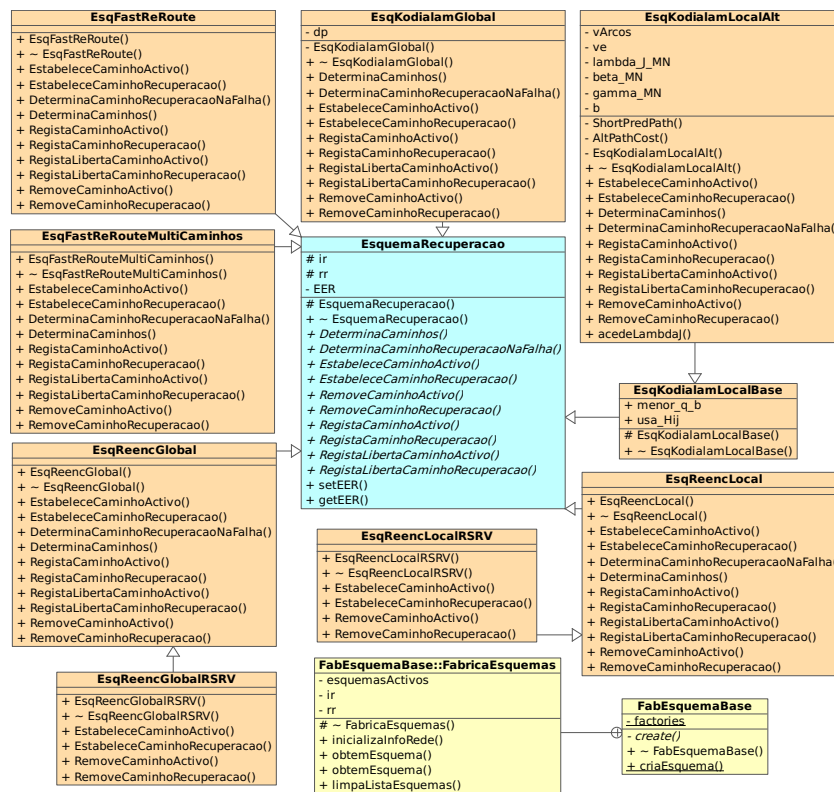


Figura A.12: Diagrama de classes – classe ESQUEMARECUPERACAO e associadas.

o destino tendo sido retirado da rede o ramo que falhou. ESTABELECECAMINHOACTIVO não faz nada uma vez que não há reservas a serem feitas. O mesmo se passa no ESTABELECECAMINHORECUPERACAO. Os métodos REGISTACAMINHOACTIVO, REGISTACAMINHORECUPERACAO, REGISTALIBERTACAMINHOACTIVO e REGISTALIBERTACAMINHORECUPERACAO possuem todos implementação vazia, porque este esquema calcula os novos caminhos em função do estado actual da rede sem qualquer informação do histórico anterior.

O esquema de reencaminhamento global com reserva, classe ESQREENCGLOBALRSRV, é implementado como uma especialização da classe ESQREENCGLOBAL (reencaminhamento global sem reserva). As únicas funções redefinidas são aquelas que tem a ver justamente com a reserva, ESTABELECECAMINHOACTIVO que trata de fazer a reserva do caminho activo uniforme e ESTABELECECAMINHORECUPERACAO que reserva um caminho uniforme para o CR, utilizando respectivamente as funções RESERVACAMINHOACTIVOUNIFORME e RESERVACAMINHORECUPERACAOUNIFORME da estrutura RESERVARSVP mencionada anteriormente); e REMOVECAMINHOACTIVO e REMOVECAMINHORECUPERACAO que libertam as reservas feitas para o pedido, recorrendo respectivamente às funções LIBERTACAMI-

NHOACTIVOUNIFORME e LIBERTACAMINHORECUPERACAOUNIFORME, da estrutura RESERVARSVP. Nos esquemas de reencaminhamento global, quando é determinado o CR este só precisa de ser disjunto do CA no ramo que falhou.

Para suportar esquemas de reencaminhamento local sem reserva foi definida a classe ESQREENCLOCAL, muito semelhante ao reencaminhamento global. A determinação do CA é feita exactamente da mesma forma. A determinação do CR (na falha) é um pouco mais complexa, é necessário ver qual é o ramo que falhou e determinar um caminho do nó imediatamente anterior ao ramo que falhou até ao destino, sendo depois feito uma junção entre a parte inicial do CA (parte antes da falha) e o caminho determinado, para obter o caminho de recuperação a usar. Neste esquema, como no caso do reencaminhamento global, as funções ESTABELECE..., REGISTA..., REGISTALIBERTA... e REMOVE... não desempenham qualquer trabalho. Na determinação do CA é usado o algoritmo de *Dijkstra*, tomando em consideração a informação relativa às ocupações fornecida pelo BCM, para a determinação do CR.

No reencaminhamento local com reserva, classe ESQREENCLOCALRSRV especialização de ESQREENCLOCAL, são definidas as funções ESTABELECECAMINHOS... e REMOVECAMINHOS... para tratar das reservas de forma semelhante ao que se faz no reencaminhamento global com reserva. As funções ESTABELECECAMINHO-RECUPERACAO e REMOVECAMINHORECUPERACAO são definidas mas apenas são invocadas perante uma falha que implique a determinação do CR. Outra semelhança é a definição das funções REGISTA... e REGISTALIBERTA... ser vazia uma vez que não é necessário guardar informação interna para estes esquemas.

Em todas as classes definidas para os restantes esquemas (de protecção) a função DETERMINACAMINHORECUPERACAOINFALHA é vazia porque nestes esquemas o CR não é determinado na falha. Nestes esquemas, o método DETERMINACAMINHOS obtém o CA e um ou vários CR dependendo do âmbito da recuperação (global ou local).

Para suportar o *Fast Reroute* foi definida a classe ESQFASTREROUTE. As funções de determinação dos caminhos usam o algoritmo *Dijkstra* como no caso dos esquemas de reencaminhamento (a diferença principal relativamente a estes está na altura em que os CR são determinados). A forma de os obter é semelhante ao referido na obtenção dos CR no reencaminhamento local. Aqui os vários CR de um CA partilham recursos entre eles, ou seja se vários CR utilizam o mesmo ramo esse ramo só será reservado uma vez. Mais uma vez as funções REGISTA... e REGISTALIBERTA... são vazias, pelas mesmas razões atrás referidas. As funções ESTABELECE... tratam da reserva, utiliza para tal a classe RESERVARSVP (as reservas são sempre geridas por esta classe).

Para suportar outra versão do *Fast Reroute* foi desenvolvida a classe `EsqFAST-REROUTEMULTICAMINHOS`. Esta implementação é semelhante embora ligeiramente mais complexa que a do esquema anterior. Aqui a determinação do CA não é feita utilizando directamente o *Dijkstra* mas sim utilizando uma implementação do algoritmo MPS, cedida pela Prof. Teresa Gomes. Essa implementação, vai fornecendo os sucessivos caminhos mais curtos. Ir-se-ão permitir CA com até mais dois ramos que o caminho mais curto mínimo. Desta forma tentar-se-á obter-se um novo caminho mais curto, que poderá vir a ser usado como CA, enquanto não tiver sido possível encontrar CR para todos os seus ramos e o número de ramos do novo caminho mais curto não seja superior em dois ao número de ramos do caminho mais curto mínimo. Sendo um esquema com reserva, as funções `ESTABELECECAMINHO...` tratam dessa questão recorrendo a uma instância da classe `RESERVARVSP`. As funções `REGISTA...` e `REGISTALIBERTA...` não têm nada a fazer. As funções `REMOVECAMINHOACTIVO` e `REMOVECAMINHORECUPERACAO` vão chamar respectivamente as funções `LIBERTACAMINHOACTIVOUNIFORME` e `LIBERTACAMINHORECUPERACAO_NAOUNIFORME`, da estrutura `RESERVARVSP`, para remover arcos que não formam um caminho (essa função suporta também variação nas capacidades dos arcos, mas tal não ocorre neste esquema).

Finalmente para suportar os esquemas de protecção com partilha explícita¹ de LBP, usando os esquemas referidos no Capítulo 3, foram desenvolvidas as classes `EsqKODIALAMGLOBAL` (para recuperação global), `EsqKODIALAMLOCALBASE` e `EsqKODIALAMLOCALALT` (para recuperação local). A criação de duas classes para protecção local com partilha foi necessária para permitir o estudo do comportamento do esquema com as diferentes variações detalhadas na Secção 3.2.2. Em todas estas classes, o método `DETERMINACAMINHOS` efectua os cálculos para determinar o CA e CR (vários CR no caso local) de forma a fornecer protecção com partilha. Os métodos `REGISTACAMINHOACTIVO` e `REGISTACAMINHORECUPERACAO` tratam de actualizar a estrutura de informação (parcial) que representam o estado de protecção dos ramos dos CA pelos ramos de CR, sendo esta informação armazenada de forma agregada (e portanto inexata). Os métodos `ESTABELECECAMINHOACTIVO` e `ESTABELECECAMINHORECUPERACAO` tratam, respectivamente, de efectuar a reserva (exacta) dos ramos do CA (reserva de um caminho uniforme) e dos ramos do(s) CR (reserva de um conjunto de ramos não uniforme, com valores reservados por ramo potencialmente diferentes). De novo, estes mé-

¹Nas duas versões do algoritmo *Fast Reroute*, o sistema suporta a partilha intra-pedido (ou seja, não reserva duplamente a LBP usada por diferentes CR para o mesmo pedido), mas tal ocorre apenas incidentalmente, uma vez que os algoritmos usados não funcionam com o objectivo de aumentar a partilha.

todos usam a classe `RESERVARVSP` para este fim. Os métodos `REGISTALIBERTACAMINHOACTIVO` e `REGISTALIBERTACAMINHORECUPERACAO` tratam de actualizar o registo de protecção de ramos aquando do termo de um pedido (esta tarefa é mais complexa do que a tarefa inicial de actualização, implicando verificar caso a caso a forma como a remoção de cada ramo do CA determina a possibilidade de libertar alguma LB reservada para permitir CR). Finalmente, os métodos `REMOVECAMINHOACTIVO` e `REMOVECAMINHORECUPERACAO` permitem remover (ou actualizar) as reservas feitas anteriormente.

Para permitir ao simulador criar quando necessário (pelas tabelas) novas instâncias das classes anteriores, foi desenvolvida a classe `FABRICAESQUEMAS`. Nas tabelas, cada esquema de recuperação implementado é identificado por uma etiqueta. Cada instância de um esquema de recuperação é gerada recorrendo à classe `FABRICAESQUEMAS` conhecendo-se essa etiqueta. Desta forma é possível trabalhar com diversos esquemas numa mesma simulação e variar de corrida para corrida esses esquemas sem necessidade de alterar a implementação.

A.5 Componentes auxiliares

Para além dos componentes genéricos descritos anteriormente, foi também necessário desenvolver um conjunto de classes de apoio a esses componentes.

A classe `DIJKSTRA` é uma classe principalmente funcional, a sua razão principal é determinar a árvore de caminhos mais curtos usando o algoritmo de Dijkstra (1959). O construtor recebe os nós da rede, os ramos da rede e um vector de custos. No vector de custos dos ramos além dos custos de cada ramo é também guardada a informação que indica se um ramo está ou não activo (com a implementação de ramos activos ou inactivos podem evitar-se os erros numéricos que poderiam surgir se se usassem custos infinitos). Na classe foi implementada uma função que permite alterar os custos mantendo tudo o resto igual, dado que ocorre frequentemente a necessidade de calcular vários problemas de caminhos mais curtos em que a única coisa que varia são os custos dos ramos.

O objecto suporta várias opções para determinar as árvores de caminhos mais curtos: de uma origem para todos os destinos, de uma origem para um só destino (esta opção foi feita apenas por questões de optimização de tempo uma vez que o código é o mesmo que o da função anterior, mas podendo terminar um pouco mais cedo), reversa para um destino de todas as origens e reversa entre uma origem e um destino. Estas funções levam a que seja armazenado internamente informação das etiquetas calculadas associadas aos nós. Estas funções (cálculo da árvore de caminhos mais curtos) apenas actualizam as etiquetas internas as-

sociadas à rede. Depois de terem sido actualizadas as etiquetas, para obter os caminhos são usadas funções como `GETCAMINHO` e `GETCAMINHOREVERSO` para obter os caminhos completos, ou eventualmente `GETETIQUETA` e `GETANTECESSOR` se se pretender uma visão mais parcelar dos resultados (nó a nó da rede).

No desenvolvimento do código foram ocasionalmente usados Padrões (um padrão pode ser definido como uma solução generalizável para a resolução de uma classe de problemas, ver Gamma *et al.*, 1994) para resolver problemas comuns. Tal ocorre por exemplo na classe `FABRICAESQUEMAS` descrita anteriormente, que implementa os padrões `FACTORY` e `SINGLETON`. A implementação do padrão `SINGLETON` é usada para garantir que existe em cada momento apenas uma instância da classe `FABRICAESQUEMAS`. A implementação do padrão `FACTORY` é usada para gerir a criação de instâncias dos esquemas de recuperação e tornar fácil a introdução de novos esquemas de recuperação. O código `SINGLETON` é constituído por classes e *templates* que implementam ou auxiliam a implementação do padrão `SINGLETON` (algumas das quais são usadas no objecto `FABRICAESQUEMAS` e noutros componentes do código do simulador).

B O simulador - Ao nível do pacote

O simulador ao nível do pacote não foi desenvolvido completamente. No entanto grande parte da implementação realizada poderá vir a ser reutilizada na implementação desse simulador, isto porque foi considerada essa preocupação na implementação efectuada. Neste apêndice serão descritos alguns desenvolvimentos, específicos da simulação ao pacote, que foram feitos tendo por objectivo a sua implementação futura.

Este apêndice começa com uma ligeira descrição das características mais importantes do enquadramento simulacional, INET, desenvolvido no OMNeT++. Esta descrição é seguida de algumas alterações internas ou parcialmente internas ao ambiente simulacional.

B.1 O INET

No OMNeT++ foi desenvolvido um enquadramento simulacional modular designado INET, que permite a simulação de redes de computadores, estando disponíveis vários protocolos (incluindo IPv4, IPv6 e MPLS) e modelos de equipamentos (incluindo *hosts* e LSR). Este enquadramento bem como o próprio OMNeT++ são fornecidos em código aberto.

No INET, devido à sua complexidade (número elevado de ficheiros NED e “makefile”), os ficheiros NED são *preload* para memória (ou seja, compilados todos para memória no arranque).

O INET é como que uma grande biblioteca de módulos simples (o código destes encontra-se integrado na aplicação INET), estando a cada um destes associado um ficheiro NED. Existe um conjunto de módulos compostos que agregam módulos simples em outros ficheiros NED. Por sua vez, existe outro conjunto de módulos que agregam módulos compostos em outros ficheiros NED e/ou módulos simples e por aí diante até que finalmente se constroem os modelos/redes a partir dos ficheiros NED dos módulos compostos (de mais alto nível e de outros).

O INET permite a construção de uma enorme quantidade de modelos distintos simplesmente através da integração dos componentes existentes, no entanto, é

possível desenvolver modelos com alterações simples ou mesmo profundas nos módulos existentes.

B.2 Alterações/adições no INET

Podemos dividir as alterações/adições ao INET nos três grupos seguintes:

1. Substituir/escolher capacidades dos módulo, como possibilidade prevista ao nível do INET;
2. Alterar internamente o código do INET;
3. Integrar novas funcionalidades com o código do INET:
 - a) Substituir um módulo simples (ou composto) existente por um novo ao INET;
 - b) Acrescentar novos módulos (simples ou compostos) a módulos compostos.

A alteração em 1 é a mais simples de fazer e implica apenas alterações ao nível do modelo/rede (módulo de nível mais alto) ou do ficheiro de configuração “*omnetpp.ini*”.

A alteração 2 torna-se possível devido ao OMNeT++/INET ser uma aplicação de código aberto. Alterações que possam ser consideradas relevantes (de âmbito geral) podem ser submetidas ao responsável do projecto INET, para uma possível inclusão na aplicação. Esta alteração pode ser uma alteração ao nível de módulos existentes ou a criação de novos módulos (ou modelos).

Como já foi referido, o INET permite a construção de uma grande quantidade de modelos sem ter que se desenvolverem novos módulos simples. No entanto, devido à grande diversidade de características das redes de comunicação, geralmente será necessário, nas análises que se pretendem realizar em projectos de investigação, acrescentar novas funcionalidades ao INET (alteração 3). Os módulos desenvolvidos para tal funcionarão de forma integrada com os já existentes.

Quando a alteração que se pretende corresponde ao tipo 3a, isto é, quando se pretender substituir um módulo existente por um novo (ao INET) uma forma de o conseguir é ver em que módulos é que ele era incorporado e alterar esses módulos para passar a incorporar o novo módulo (estas alterações devem ser feitas nas posições em que era utilizado o módulo antigo). Para acrescentar um novo módulo (alteração 3b) é necessário alterar o ficheiro NED correspondente

ao módulo composto que passa a incorporar o novo módulo. Para garantir independência relativa a variações no INET as alterações de tipo 3a e 3b podem ser conseguidas através da criação adicional de novos módulos (baseados ou não nos pré-definidos no INET) ao invés da alteração directa dos módulos do INET. Foi esta a abordagem seguida prioritariamente na implementação realizada. De notar então que, uma vez criado um módulo composto adicional, para garantir independência relativa a variações no INET, todos os seus módulos de níveis hierarquicamente superiores terão de ser módulos novos ao INET (baseados ou não nos pré-definidos no INET).

Quando a alteração envolveu a substituição de um módulo simples existente por um novo a abordagem seguida foi uma das duas seguintes:

- o módulo novo foi obtido por derivação da classe associada ao módulo que se pretende alterar, ou
- o funcionamento do módulo original foi duplicado (por cópia) no novo, e seguidamente foram acrescentadas neste as funcionalidades necessárias.

A abordagem preferível seria a primeira, pois dessa forma as futuras alterações do módulo original, feitas internamente ao nível da aplicação INET, seriam reflectidas automaticamente no módulo novo. No entanto, para situações que envolvam a necessidade de manipulação de estruturas internas a derivação não é adequada.

Um exemplo da alteração 1, envolvendo a substituição de um módulo simples, é: A substituição do gerador de tráfego na definição de um LSR (módulo *RSVP_LSR*). O gerador usado pode ser um gerador de pacotes *User Datagram Protocol* (UDP) ou *Transmission Control Protocol* (TCP).

No entanto, na maior parte dos casos as alterações que se pretendem são mais profundas, pois dizem respeito à utilização de novos protocolos ou ao funcionamento mais alargado de determinados protocolos. Pode tomar-se como exemplos da alteração 3 o caso seguinte:

- Pretendia-se que os LSR tratassem de forma distintas as várias classes de serviço.
- E também tornar o funcionamento de vários dos seus protocolos (módulos simples) mais abrangente.

Para tal criou-se um novo módulo adicional *RSVP_LSR_QoS*, baseado no que já existe (*RSVP_LSR*), mas incorporando nele um novo módulo simples adicional, que trata de fazer a diferenciação do tráfego, este módulo foi colocado entre

o módulo simples *MPLS* e o módulo simples *PPPInterface*. Foram também substituídos os módulos simples *TED*, *RSVP* e *MPLS* pelos novos módulos *TED_QoS*, *RSVP_QoS* e *MPLS_QoS*, respectivamente. O procedimento utilizado na substituição destes módulos será referido nas subsecções seguintes.

Um exemplo da alteração 2 é a alteração do módulo *RSVP_LSR* para fazer o tratamento de falhas (ver subsecção B.3.1).

Na subsecção seguinte descrevem-se as alterações principais que foram feitas ao INET para obter as funcionalidade desejadas para o sistema.

B.3 Alterações/adições efectuadas no INET no desenvolvimento do nosso modelo

B.3.1 Alteração interna para permitir a simulação de falhas de ramos

O INET possui um módulo simples, designado por *FailureManager*, para a geração de falhas de nós. A geração da falha é efectuada de uma forma um pouco simplista pois consiste na substituição em *run time* do LSR (módulo composto *RSVP_LSR*) por outro módulo, designado por *RSVP_FAILED*, cuja tarefa é simplesmente deitar fora os pacotes que lá chegam.

Foi necessário alterar o módulo *FailureManager* para suportar também a geração de falhas de ramos¹. Agora quando ocorre a falha de um ramo o *FailureManager* sinaliza uma *flag* no módulo *PPPInterface*. Foi necessário alterar o interface com os ramos (módulo *PPPInterface* – implementação de ligações ponto a ponto usada pelo MPLS) para passar a analisar o valor dessa *flag* e em função do seu valor tomar a acção apropriada (enviar ou descartar os pacotes, respectivamente quando o interface estiver a funcionar ou em baixo). No módulo simples *RSVP* foi também necessário fazer alterações para possibilitar a ocorrência de falhas bidireccionais dos ramos. A alteração necessária neste módulo teve a ver com a necessidade do envio de *hellos*, mesmo quando o ramo falhou, caso contrário, no caso de falhas bidireccionais de um ramo, quando o ramo fosse recuperado o RSVP continuava a achar que estava em baixo. Isto acontecia porque um nó esperava receber um *hello* antes de enviar um e o outro nó fazia a mesma coisa, logo ambos ficavam à espera eternamente. O que se fez foi enviar sempre *hellos*, mesmo quando existe falha no ramo (mas obviamente, neste caso, estes pacotes irão ser perdido).

¹Assume-se que quando um ramo falha a falha ocorre nos dois sentidos, esta restrição é imposta pelo módulo simples *RSVP*.

B.3.2 Alteração para permitir a criação de pedidos RSVP não *scripted*

O INET só permitia a geração de pedidos RSVP *scripted*, gerados todos no início da simulação ou ao longo delas, no entanto tinham que ser pré-planeados em ambos os casos antes da simulação. A alteração efectuada passou a permitir a geração de pedidos *online*. Foi criado para tal um novo *ScenarioManager* para substituir o *ScenarioManager* existente ao nível do INET. O novo *ScenarioManager* foi obtido por derivação da classe associada ao *ScenarioManager* existente no INET. Este módulo possui uma estrutura muito semelhante ao módulo *FailureManager*, ambos permitem desencadear acções em módulos que esperam que essas acções sejam desencadeadas para efectuarem determinada tarefa, e ao mesmo tempo alertar outros módulos de que essas acções foram desencadeadas. O *ScenarioManager* do INET lê as instruções da criação de acções para serem desempenhadas por diversos componentes (módulos simples geralmente) aquando da sua inicialização. A alteração efectuada passou a permitir ao *ScenarioManager* receber informação ao longo da simulação. Isto foi o suficiente para permitir pedidos *online* uma vez que os outros módulos já estavam preparados para receber ordens do *ScenarioManager* a qualquer altura.

Criação de uma estrutura XML Ainda para possibilitar a criação de pedidos não *scripted* foi necessário traduzir esses pedidos numa estrutura *eXtensible Markup Language* (XML). Pois o módulo simples *ScenarioManager* (do INET) obtém os dados nesse formato.

B.3.3 Geração de pedidos

Existem no INET geradores de tráfego (pacotes). Foi no entanto necessário criar um módulo de geração de pedidos para gerar pedidos com as características específicas do tráfego que se pretende gerar (por exemplo: com determinada LB, duração, classe de tráfego, com determinada origem/destino, etc).

B.3.4 LSR com tratamento diferenciado por classe

Os módulos simples *MPLS*, *RSVP* e *TED* foram substituídos pelos módulos *TED_QoS*, *RSVP_QoS*, *MPLS_QoS*, respectivamente. Estes módulos são um pouco complexos e por isso não foram obtidos por derivação mas sim por duplicação inicialmente e alterações posteriores. Seguiu-se esta abordagem pois existia em geral a necessidade de manipulação de estruturas internas. Partiu-se do pressuposto que a estrutura do INET não será alterada. Se forem feitas alterações

nestes módulos simples, no INET, e se pretendermos também ter essas alterações no nosso código é necessário inclui-las nos módulos simples que redefinimos.

C Tipos de tabelas usadas nas experiências

Neste apêndice são apresentadas as tabelas usadas nas experiências realizadas no Capítulo 6.

As Tabelas C.1, C.2, C.3, C.4, C.5, C.6, C.7, C.8, C.9, C.10, C.11, C.12, C.13 e C.14 correspondem respectivamente aos tipos de tabela 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13 e 14 e são usadas na experiência da Secção 6.5.1 do Capítulo 6. O número total de corridas distintas resultante para este conjunto de tabelas foi $1680 = 14(\text{tabelas}) \times 10(\text{réplicas}) \times 2(\text{redes}) \times 6(\text{factores de carga})$.

| <i>Condição</i> | CT0 | CT1 | CT2 | CT3 |
|-----------------|-----|-----|-----|-----|
| <i>Sempre</i> | RL4 | PL1 | PL1 | PL1 |

Tabela C.1: Tabela “mono-esquema” (RL4-PL1).

| <i>Condição</i> | CT0 | CT1 | CT2 | CT3 |
|-----------------|-----|-----|-----|-----|
| <i>Sempre</i> | RL4 | PL2 | PL2 | PL2 |

Tabela C.2: Tabela “mono-esquema” (RL4-PL2).

| <i>Condição</i> | CT0 | CT1 | CT2 | CT3 |
|-----------------|-----|-----|-----|-----|
| <i>Sempre</i> | RL4 | PG1 | PG1 | PG1 |

Tabela C.3: Tabela “mono-esquema” (RL4-PG1).

| <i>Condição</i> | CT0 | CT1 | CT2 | CT3 |
|-----------------|-----|-----|-----|-----|
| <i>Sempre</i> | RL4 | RG2 | RG2 | RG2 |

Tabela C.4: Tabela “mono-esquema” (RL4-RG2).

| <i>Condição</i> | CT0 | CT1 | CT2 | CT3 |
|-----------------|-----|-----|-----|-----|
| <i>Sempre</i> | RL4 | RL5 | RL5 | RL5 |

Tabela C.5: Tabela “mono-esquema” (RL4-RL5).

| <i>Condição</i> | CT0 | CT1 | CT2 | CT3 |
|-----------------|------------|------------|------------|------------|
| <i>Sempre</i> | RG1 | PL1 | PL1 | PL1 |

Tabela C.6: Tabela “mono-esquema” (RG1-PL1).

| <i>Condição</i> | CT0 | CT1 | CT2 | CT3 |
|-----------------|------------|------------|------------|------------|
| <i>Sempre</i> | RG1 | PL2 | PL2 | PL2 |

Tabela C.7: Tabela “mono-esquema” (RG1-PL2).

| <i>Condição</i> | CT0 | CT1 | CT2 | CT3 |
|-----------------|------------|------------|------------|------------|
| <i>Sempre</i> | RG1 | PG1 | PG1 | PG1 |

Tabela C.8: Tabela “mono-esquema” (RG1-PG1).

| <i>Condição</i> | CT0 | CT1 | CT2 | CT3 |
|-----------------|------------|------------|------------|------------|
| <i>Sempre</i> | RG1 | RG2 | RG2 | RG2 |

Tabela C.9: Tabela “mono-esquema” (RG1-RG2).

| <i>Condição</i> | CT0 | CT1 | CT2 | CT3 |
|-----------------|------------|------------|------------|------------|
| <i>Sempre</i> | RG1 | RL5 | RL5 | RL5 |

Tabela C.10: Tabela “mono-esquema” (RG1-RL5).

| <i>Condição</i> | CT0 | CT1 | CT2 | CT3 |
|-----------------|------------|------------|------------|------------|
| <i>Sempre</i> | RL4 | PL3 | PL3 | PL3 |

Tabela C.11: Tabela “mono-esquema” (RL4-PL3).

| <i>Condição</i> | CT0 | CT1 | CT2 | CT3 |
|-----------------|------------|------------|------------|------------|
| <i>Sempre</i> | RG1 | PL3 | PL3 | PL3 |

Tabela C.12: Tabela “mono-esquema” (RG1-PL3).

| <i>Condição</i> | CT0 | CT1 | CT2 | CT3 |
|-----------------|------------|------------|------------|------------|
| <i>Sempre</i> | RL4 | PG3 | PG3 | PG3 |

Tabela C.13: Tabela “mono-esquema” (RL4-PG3).

| <i>Condição</i> | CT0 | CT1 | CT2 | CT3 |
|-----------------|------------|------------|------------|------------|
| <i>Sempre</i> | RG1 | PG3 | PG3 | PG3 |

Tabela C.14: Tabela “mono-esquema” (RG1-PG3).

As Tabelas C.15, C.16, C.17, C.18, C.19, C.20, C.21, C.22, C.23, C.24, C.25 e C.26, correspondem respectivamente aos tipos de tabela 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71 e 72, e são usadas na experiência da Secção 6.5.2 do Capítulo 6. O número total de corridas distintas efectuadas com este conjunto de tabelas foi $1440 = 12(\text{tabelas}) \times 10(\text{réplicas}) \times 2(\text{redes}) \times 6(\text{factores de carga})$.

| <i>Condição</i> | CT0 | CT1 | CT2 | CT3 |
|-----------------|-----|-----|-----|-----|
| <i>Sempre</i> | RL4 | RG2 | PL3 | PL2 |

Tabela C.15: Tabela “multi-esquema” (RL4-RG2-PL3-PL2).

| <i>Condição</i> | CT0 | CT1 | CT2 | CT3 |
|-----------------|-----|-----|-----|-----|
| <i>Sempre</i> | RL4 | RG2 | PL3 | PL3 |

Tabela C.16: Tabela “multi-esquema” (RL4-RG2-PL3-PL3).

| <i>Condição</i> | CT0 | CT1 | CT2 | CT3 |
|-----------------|-----|-----|-----|-----|
| <i>Sempre</i> | RL4 | RG2 | PG3 | PL2 |

Tabela C.17: Tabela “multi-esquema” (RL4-RG2-PG3-PL2).

| <i>Condição</i> | CT0 | CT1 | CT2 | CT3 |
|-----------------|-----|-----|-----|-----|
| <i>Sempre</i> | RL4 | RG2 | PG3 | PL3 |

Tabela C.18: Tabela “multi-esquema” (RL4-RG2-PG3-PL3).

| <i>Condição</i> | CT0 | CT1 | CT2 | CT3 |
|-----------------|-----|-----|-----|-----|
| <i>Sempre</i> | RL4 | PL2 | PL3 | PL2 |

Tabela C.19: Tabela “multi-esquema” (RL4-PL2-PL3-PL2).

As Tabelas C.27, C.28, C.29 e C.30, correspondem respectivamente aos tipos de tabela 81, 82, 83 e 84 e são usadas na experiência da Secção 6.5.3 do Capítulo 6. O número total de corridas distintas efectuadas com este conjunto de tabelas foi $480 = 4(\text{tabelas}) \times 10(\text{réplicas}) \times 2(\text{redes}) \times 6(\text{factores de carga})$.

As Tabelas C.31, C.32, C.33, C.34 e C.35 são usadas na experiência da Secção 6.5.4 do Capítulo 6. A cada uma destas tabelas correspondem 4 tipos de tabela, que diferem apenas no valor do limiar (limiar de carga) entre carga leve e não leve. A Tabela C.31 corresponde aos tipos de tabela 121, 151, 161 e 141 respectivamente para limiares de carga de 40%, 50%, 60% e 70%. Da mesma forma a Tabela

| <i>Condição</i> | CT0 | CT1 | CT2 | CT3 |
|-----------------|-----|-----|-----|-----|
| <i>Sempre</i> | RL4 | PL2 | PL3 | PL3 |

Tabela C.20: Tabela “multi-esquema” (RL4-PL2-PL3-PL3).

| <i>Condição</i> | CT0 | CT1 | CT2 | CT3 |
|-----------------|-----|-----|-----|-----|
| <i>Sempre</i> | RL4 | PL2 | PG3 | PL2 |

Tabela C.21: Tabela “multi-esquema” (RL4-PL2-PG3-PL2).

| <i>Condição</i> | CT0 | CT1 | CT2 | CT3 |
|-----------------|-----|-----|-----|-----|
| <i>Sempre</i> | RL4 | PL2 | PG3 | PL3 |

Tabela C.22: Tabela “multi-esquema” (RL4-PL2-PG3-PL3).

| <i>Condição</i> | CT0 | CT1 | CT2 | CT3 |
|-----------------|-----|-----|-----|-----|
| <i>Sempre</i> | RL4 | PG1 | PL3 | PL2 |

Tabela C.23: Tabela “multi-esquema” (RL4-PG1-PL3-PL2).

| <i>Condição</i> | CT0 | CT1 | CT2 | CT3 |
|-----------------|-----|-----|-----|-----|
| <i>Sempre</i> | RL4 | PG1 | PL3 | PL3 |

Tabela C.24: Tabela “multi-esquema” (RL4-PG1-PL3-PL3).

| <i>Condição</i> | CT0 | CT1 | CT2 | CT3 |
|-----------------|-----|-----|-----|-----|
| <i>Sempre</i> | RL4 | PG1 | PG3 | PL2 |

Tabela C.25: Tabela “multi-esquema” (RL4-PG1-PG3-PL2).

| <i>Condição</i> | CT0 | CT1 | CT2 | CT3 |
|-----------------|-----|-----|-----|-----|
| <i>Sempre</i> | RL4 | PG1 | PG3 | PL3 |

Tabela C.26: Tabela “multi-esquema” (RL4-PG1-PG3-PL3).

| <i>Condição</i> | CT0 | CT1 | CT2 | CT3 |
|--------------------|-----|-----|-----|-----|
| <i>Comp. do CA</i> | | | | |
| ≤ 2 | RG1 | PG1 | PG1 | PG1 |
| > 2 | RL4 | PL1 | PL1 | PL1 |

Tabela C.27: Tabela por comprimento esperado de CA (RG1/RL4-PG1/PL1).

| <i>Condição</i> <i>Comp. do CA</i> | CT0 | CT1 | CT2 | CT3 |
|---------------------------------------|-----|-----|-----|-----|
| ≤ 2 | RG1 | PG1 | PG1 | PG1 |
| > 2 | RL4 | PL3 | PL3 | PL3 |

Tabela C.28: Tabela por comprimento esperado de CA (RG1/RL4-PG1/PL3).

| <i>Condição</i> <i>Comp. do CA</i> | CT0 | CT1 | CT2 | CT3 |
|---------------------------------------|-----|-----|-----|-----|
| ≤ 2 | RG1 | PG3 | PG3 | PG3 |
| > 2 | RL4 | PL1 | PL1 | PL1 |

Tabela C.29: Tabela por comprimento esperado de CA (RG1/RL4-PG3/PL1).

| <i>Condição</i> <i>Comp. do CA</i> | CT0 | CT1 | CT2 | CT3 |
|---------------------------------------|-----|-----|-----|-----|
| ≤ 2 | RG1 | PG3 | PG3 | PG3 |
| > 2 | RL4 | PL3 | PL3 | PL3 |

Tabela C.30: Tabela por comprimento esperado de CA (RG1/RL4-PG3/PL3).

C.32 corresponde aos tipos de tabela 122, 152, 162 e 142 respectivamente para limiares de carga de 40%, 50%, 60% e 70%. Analogamente a Tabela C.33 corresponde aos tipos de tabela 123, 153, 163 e 143, a Tabela C.34 corresponde aos tipos de tabela 124, 154, 164 e 144 e a Tabela C.35 corresponde aos tipos de tabela 125, 155, 165 e 145. Na experiência da Secção 6.5.4 do Capítulo 6 são também utilizadas as Tabelas “mono-esquema” C.11, C.13, C.5 e C.2 mas com carga flutuante, as quais foram identificadas respectivamente com os tipos de tabela 131, 132, 133 e 134. O número total de corridas distintas efectuadas com este conjunto de tabelas foi $1440 = 24(\text{tabelas}) \times 10(\text{réplicas}) \times 2(\text{redes}) \times 3(\text{factores de carga})$.

| <i>Condição</i> <i>Carga</i> | CT0 | CT1 | CT2 | CT3 |
|---------------------------------|-----|-----|-----|-----|
| <i>Leve</i> | RL4 | PL3 | PL3 | PL3 |
| <i>Não leve</i> | RL4 | RL5 | RL5 | RL5 |

Tabela C.31: Tabela por carga da rede (RL4-PL3/RL5).

| <i>Condição</i> | CT0 | CT1 | CT2 | CT3 |
|-----------------|-----|-----|-----|-----|
| <i>Carga</i> | | | | |
| <i>Leve</i> | RL4 | PG3 | PG3 | PG3 |
| <i>Não leve</i> | RL4 | RL5 | RL5 | RL5 |

Tabela C.32: Tabela por carga da rede (RL4-PG3/RL5).

| <i>Condição</i> | CT0 | CT1 | CT2 | CT3 |
|-----------------|-----|-----|-----|-----|
| <i>Carga</i> | | | | |
| <i>Leve</i> | RL4 | PL3 | PL3 | PL3 |
| <i>Não leve</i> | RL4 | PL2 | PL2 | PL2 |

Tabela C.33: Tabela por carga da rede (RL4-PL3/PL2).

| <i>Condição</i> | CT0 | CT1 | CT2 | CT3 |
|-----------------|-----|-----|-----|-----|
| <i>Carga</i> | | | | |
| <i>Leve</i> | RL4 | PG3 | PG3 | PG3 |
| <i>Não leve</i> | RL4 | PL2 | PL2 | PL2 |

Tabela C.34: Tabela por carga da rede (RL4-PG3/PL2).

| <i>Condição</i> | CT0 | CT1 | CT2 | CT3 |
|-----------------|-----|-----|-----|-----|
| <i>Carga</i> | | | | |
| <i>Leve</i> | RL4 | PL3 | PL3 | PL3 |
| <i>Não leve</i> | RL4 | PG3 | PG3 | PG3 |

Tabela C.35: Tabela por carga da rede (RL4-PL3/PG3).

D Resultados adicionais

Neste apêndice serão apresentados mais alguns dos resultados obtidos para as experiências do Capítulo 6. Para os resultados apresentados em todas as tabelas são indicados os valores médios \pm metade do intervalo de confiança.

D.1 Construção de uma tabela “mono-esquema” - resultados adicionais

Nesta secção são apresentados resultados adicionais para a experiência da Secção 6.5.1 do Capítulo 6.

As Tabelas D.1, D.2, D.3, D.4, D.5, D.6, D.7, D.8 e D.9 apresentam resultados para as mesma medidas que as tabelas respectivas apresentadas na Secção 6.5.1 do Capítulo 6 (6.6, 6.7, 6.8, 6.9, 6.10, 6.11, 6.12, 6.13 e 6.14), sendo a única diferença a rede utilizada.

| | | PL2 | PG1 | PG3 | PL3 | PL1 |
|---|-----|--------------------|--------------------|--------------------|--------------------|--------------------|
| 1 | CT1 | | | 0,00% \pm 0,00% | 0,02% \pm 0,00% | 0,05% \pm 0,00% |
| | CT3 | | | 0,00% \pm 0,00% | 0,00% \pm 0,00% | 0,00% \pm 0,00% |
| 2 | CT1 | 0,04% \pm 0,00% | 0,65% \pm 0,03% | 5,08% \pm 0,06% | 13,86% \pm 0,08% | 14,32% \pm 0,09% |
| | CT3 | 0,00% \pm 0,00% | 0,01% \pm 0,00% | 0,34% \pm 0,01% | 1,39% \pm 0,02% | 1,74% \pm 0,03% |
| 3 | CT1 | 7,27% \pm 0,05% | 22,55% \pm 0,07% | 24,25% \pm 0,08% | 32,85% \pm 0,05% | 32,88% \pm 0,03% |
| | CT3 | 0,21% \pm 0,01% | 1,58% \pm 0,02% | 3,95% \pm 0,03% | 9,40% \pm 0,05% | 9,88% \pm 0,04% |
| 4 | CT1 | 19,42% \pm 0,07% | 36,39% \pm 0,08% | 36,37% \pm 0,08% | 43,40% \pm 0,06% | 43,36% \pm 0,05% |
| | CT3 | 1,26% \pm 0,03% | 9,22% \pm 0,09% | 11,77% \pm 0,08% | 19,71% \pm 0,10% | 19,97% \pm 0,07% |

Tabela D.1: Bloqueio no estabelecimento para CT1 e CT3, rede COST 239 e factores de carga entre 1 e 4 - Tabelas “mono-esquema” - Protecção.

A Tabela D.10 apresenta a probabilidade de rejeição de pedidos no estabelecimento para os tipos de tabela TT2 (RL4/PL2) e TT7 (RG1/PL2), para CT1 e CT3 e para os factores de carga entre 1 e 4. Resultados para a mesma medida também são apresentados na Tabela D.11, mas para os tipos de tabela TT4 (RL4/RG2) e TT9 (RG1/RG2), para CT1 e CT3 e para os factores de carga entre 2 e 5. A Tabela D.12 apresenta a probabilidade de desconexão na falha para os tipos de tabela

| | | RL5 | RG2 |
|---|-----|----------------|----------------|
| 2 | CT1 | 0,00% ± 0,00% | 0,00% ± 0,00% |
| | CT3 | 0,00% ± 0,00% | 0,00% ± 0,00% |
| 3 | CT1 | 0,07% ± 0,00% | 0,06% ± 0,00% |
| | CT3 | 0,01% ± 0,00% | 0,01% ± 0,00% |
| 4 | CT1 | 1,79% ± 0,03% | 1,84% ± 0,04% |
| | CT3 | 0,09% ± 0,01% | 0,08% ± 0,00% |
| 5 | CT1 | 9,09% ± 0,07% | 9,21% ± 0,08% |
| | CT3 | 0,23% ± 0,01% | 0,22% ± 0,01% |
| 6 | CT1 | 17,44% ± 0,07% | 17,59% ± 0,07% |
| | CT3 | 0,59% ± 0,01% | 0,59% ± 0,02% |

Tabela D.2: Bloqueio no estabelecimento para CT1 e CT3, rede COST 239 e factores de carga entre 2 e 6 - Tabelas "mono-esquema" - Reencaminhamento.

| | | RG2 | RL5 |
|---|-----|----------------|----------------|
| 2 | CT1 | 0,00% ± 0,00% | 0,02% ± 0,01% |
| | CT3 | 0,00% ± 0,00% | 0,00% ± 0,00% |
| 3 | CT1 | 1,89% ± 0,17% | 4,69% ± 0,29% |
| | CT3 | 0,88% ± 0,13% | 3,04% ± 0,26% |
| 4 | CT1 | 15,58% ± 0,54% | 24,85% ± 0,45% |
| | CT3 | 5,43% ± 0,32% | 11,82% ± 0,50% |
| 5 | CT1 | 33,44% ± 0,44% | 45,03% ± 0,97% |
| | CT3 | 6,99% ± 0,44% | 15,06% ± 0,73% |

Tabela D.3: Desconexão na falha para CT1 e CT3, rede COST 239 e factores de carga entre 2 e 5 - Tabelas "mono-esquema" - Reencaminhamento.

| | | PG3 | PL3 | PL1 | PL2 | PG1 |
|---|-----|---------------|---------------|---------------|---------------|---------------|
| 1 | CT1 | 1,569 ± 0,001 | 1,572 ± 0,001 | 1,573 ± 0,001 | 1,834 ± 0,001 | 2,073 ± 0,001 |
| | CT3 | 1,569 ± 0,001 | 1,569 ± 0,001 | 1,569 ± 0,001 | 1,851 ± 0,001 | 1,992 ± 0,001 |
| 2 | CT1 | 1,670 ± 0,001 | 1,702 ± 0,001 | 1,692 ± 0,001 | 1,844 ± 0,001 | 2,266 ± 0,002 |
| | CT3 | 1,599 ± 0,001 | 1,621 ± 0,001 | 1,621 ± 0,001 | 1,837 ± 0,001 | 2,076 ± 0,001 |
| 3 | CT1 | 1,744 ± 0,001 | 1,732 ± 0,001 | 1,719 ± 0,001 | 2,020 ± 0,001 | 2,417 ± 0,003 |
| | CT3 | 1,668 ± 0,001 | 1,688 ± 0,001 | 1,681 ± 0,001 | 1,880 ± 0,001 | 2,218 ± 0,002 |
| 4 | CT1 | 1,755 ± 0,001 | 1,733 ± 0,001 | 1,719 ± 0,001 | 2,088 ± 0,001 | 2,436 ± 0,005 |
| | CT3 | 1,715 ± 0,001 | 1,717 ± 0,001 | 1,707 ± 0,001 | 1,933 ± 0,001 | 2,339 ± 0,002 |

Tabela D.4: Número de ramos dos CA para CT1 e CT3, rede COST 239 e factores de carga entre 1 e 4 - Tabelas "mono-esquema" - Protecção.

| | | RL5 | RG2 |
|---|-----|---------------|---------------|
| 1 | CT1 | 1,569 ± 0,001 | 1,569 ± 0,001 |
| | CT3 | 1,568 ± 0,001 | 1,568 ± 0,001 |
| 2 | CT1 | 1,574 ± 0,001 | 1,574 ± 0,001 |
| | CT3 | 1,570 ± 0,001 | 1,570 ± 0,001 |
| 3 | CT1 | 1,618 ± 0,001 | 1,619 ± 0,001 |
| | CT3 | 1,587 ± 0,001 | 1,587 ± 0,001 |
| 4 | CT1 | 1,773 ± 0,002 | 1,775 ± 0,001 |
| | CT3 | 1,633 ± 0,001 | 1,633 ± 0,001 |
| 5 | CT1 | 1,950 ± 0,001 | 1,951 ± 0,001 |
| | CT3 | 1,674 ± 0,001 | 1,674 ± 0,001 |

Tabela D.5: Número de ramos dos CA para CT1 e CT3, rede COST 239 e factores de carga entre 1 e 5 – Tabelas “mono-esquema” – Reencaminhamento.

| | | PG3 | PL1 | PL3 | PL2 | PG1 |
|---|-----|---------------|---------------|---------------|---------------|---------------|
| 1 | CT1 | 7,492 ± 0,006 | 7,508 ± 0,006 | 7,520 ± 0,006 | 8,524 ± 0,007 | 9,563 ± 0,007 |
| | CT3 | 7,479 ± 0,007 | 7,488 ± 0,007 | 7,493 ± 0,007 | 8,406 ± 0,009 | 9,212 ± 0,008 |
| 2 | CT1 | 7,312 ± 0,006 | 7,041 ± 0,004 | 7,085 ± 0,006 | 8,582 ± 0,006 | 9,742 ± 0,009 |
| | CT3 | 7,450 ± 0,008 | 7,352 ± 0,007 | 7,391 ± 0,006 | 8,584 ± 0,008 | 9,542 ± 0,012 |
| 3 | CT1 | 6,728 ± 0,006 | 6,367 ± 0,006 | 6,400 ± 0,004 | 8,278 ± 0,006 | 9,119 ± 0,005 |
| | CT3 | 7,200 ± 0,007 | 6,971 ± 0,007 | 7,010 ± 0,007 | 8,597 ± 0,008 | 9,496 ± 0,012 |
| 4 | CT1 | 6,249 ± 0,006 | 5,888 ± 0,003 | 5,908 ± 0,003 | 7,802 ± 0,006 | 8,510 ± 0,009 |
| | CT3 | 6,897 ± 0,008 | 6,607 ± 0,006 | 6,646 ± 0,007 | 8,469 ± 0,009 | 9,232 ± 0,009 |

Tabela D.6: LB média dos CA para CT1 e CT3, rede KL-15 e factores de carga entre 1 e 4 – Tabelas “mono-esquema” – Protecção.

| | | RL5 | RG2 |
|---|-----|---------------|---------------|
| 1 | CT1 | 7,496 ± 0,006 | 7,496 ± 0,006 |
| | CT3 | 7,490 ± 0,007 | 7,490 ± 0,007 |
| 2 | CT1 | 7,632 ± 0,006 | 7,633 ± 0,006 |
| | CT3 | 7,556 ± 0,007 | 7,557 ± 0,007 |
| 3 | CT1 | 8,067 ± 0,008 | 8,069 ± 0,008 |
| | CT3 | 7,808 ± 0,009 | 7,812 ± 0,008 |
| 4 | CT1 | 8,404 ± 0,008 | 8,407 ± 0,007 |
| | CT3 | 8,127 ± 0,008 | 8,132 ± 0,009 |

Tabela D.7: LB média dos CA para CT1 e CT3, rede KL-15 e factores de carga entre 1 e 4 – Tabelas “mono-esquema” – Reencaminhamento.

| | | PG1 | PL2 | PG3 | PL1 | PL3 |
|---|-----|---------------|---------------|----------------|----------------|----------------|
| 1 | CT1 | 4,217 ± 0,005 | 4,783 ± 0,005 | 10,643 ± 0,005 | 15,799 ± 0,011 | 15,882 ± 0,013 |
| | CT3 | 5,242 ± 0,008 | 5,995 ± 0,012 | 10,599 ± 0,009 | 15,612 ± 0,013 | 15,701 ± 0,014 |
| 2 | CT1 | 3,461 ± 0,003 | 4,060 ± 0,006 | 10,990 ± 0,008 | 15,950 ± 0,009 | 16,030 ± 0,014 |
| | CT3 | 4,149 ± 0,008 | 4,687 ± 0,007 | 10,889 ± 0,011 | 16,145 ± 0,015 | 16,243 ± 0,015 |
| 3 | CT1 | 2,958 ± 0,004 | 3,595 ± 0,003 | 10,572 ± 0,009 | 14,770 ± 0,013 | 14,796 ± 0,012 |
| | CT3 | 3,590 ± 0,007 | 4,247 ± 0,007 | 10,929 ± 0,012 | 15,785 ± 0,017 | 15,850 ± 0,022 |
| 4 | CT1 | 2,627 ± 0,004 | 3,363 ± 0,007 | 10,035 ± 0,008 | 13,788 ± 0,011 | 13,773 ± 0,011 |
| | CT3 | 3,207 ± 0,009 | 3,920 ± 0,007 | 10,722 ± 0,012 | 15,192 ± 0,015 | 15,238 ± 0,019 |

Tabela D.8: LB média para o/s CR de um CA para CT1 e CT3, rede KL-15 e factores de carga entre 1 e 4 – Tabelas "mono-esquema" – Protecção.

| | | RL5 | RG2 |
|---|-----|---------------|----------------|
| 1 | CT1 | 8,297 ± 0,048 | 10,812 ± 0,057 |
| | CT3 | 8,325 ± 0,057 | 10,817 ± 0,083 |
| 2 | CT1 | 9,000 ± 0,049 | 11,131 ± 0,047 |
| | CT3 | 8,846 ± 0,062 | 11,033 ± 0,066 |
| 3 | CT1 | 9,619 ± 0,067 | 11,380 ± 0,059 |
| | CT3 | 9,493 ± 0,118 | 11,259 ± 0,114 |
| 4 | CT1 | 9,268 ± 0,082 | 10,679 ± 0,060 |
| | CT3 | 9,637 ± 0,133 | 11,287 ± 0,151 |

Tabela D.9: LB média para o/s CR de um CA para CT1 e CT3, rede KL-15 e factores de carga entre 1 e 4 – Tabelas "mono-esquema" – Reencaminhamento.

| | | RL4/PL2 | RG1/PL2 |
|---|-----|----------------|----------------|
| 1 | CT1 | 0,63% ± 0,02% | 0,63% ± 0,01% |
| | CT3 | 0,53% ± 0,01% | 0,53% ± 0,01% |
| 2 | CT1 | 6,68% ± 0,04% | 6,67% ± 0,03% |
| | CT3 | 2,92% ± 0,02% | 2,91% ± 0,02% |
| 3 | CT1 | 18,59% ± 0,05% | 18,60% ± 0,05% |
| | CT3 | 8,24% ± 0,05% | 8,24% ± 0,05% |
| 4 | CT1 | 27,51% ± 0,07% | 27,52% ± 0,06% |
| | CT3 | 13,71% ± 0,08% | 13,68% ± 0,07% |

Tabela D.10: Bloqueio no estabelecimento para CT1 e CT3, rede KL-15 e factores de carga entre 1 e 4 – Tipos de tabela TT2 (RL4/PL2) e TT7 (RG1/PL2).

| | | RL4/RG2 | RG1/RG2 |
|---|-----|--------------------|--------------------|
| 2 | CT1 | 0,11% \pm 0,00% | 0,11% \pm 0,00% |
| | CT3 | 0,04% \pm 0,00% | 0,04% \pm 0,00% |
| 3 | CT1 | 1,55% \pm 0,02% | 1,55% \pm 0,02% |
| | CT3 | 0,46% \pm 0,01% | 0,46% \pm 0,01% |
| 4 | CT1 | 6,50% \pm 0,04% | 6,51% \pm 0,05% |
| | CT3 | 1,76% \pm 0,02% | 1,76% \pm 0,03% |
| 5 | CT1 | 12,82% \pm 0,05% | 12,84% \pm 0,06% |
| | CT3 | 3,71% \pm 0,02% | 3,72% \pm 0,03% |

Tabela D.11: Bloqueio no estabelecimento para CT1 e CT3, rede KL-15 e factores de carga entre 2 e 5 – Tipos de tabela TT4 (RL4/RG2) e TT9 (RG1/RG2).

| | | RG1/RG2 | RL4/RG2 |
|---|-----|--------------------|--------------------|
| 2 | CT1 | 2,52% \pm 0,15% | 2,64% \pm 0,16% |
| | CT3 | 1,43% \pm 0,13% | 1,45% \pm 0,15% |
| 3 | CT1 | 14,73% \pm 0,31% | 14,88% \pm 0,35% |
| | CT3 | 9,47% \pm 0,39% | 9,26% \pm 0,41% |
| 4 | CT1 | 27,79% \pm 0,45% | 28,11% \pm 0,59% |
| | CT3 | 17,28% \pm 0,52% | 17,19% \pm 0,51% |
| 5 | CT1 | 36,63% \pm 0,59% | 36,69% \pm 0,62% |
| | CT3 | 23,06% \pm 0,51% | 23,26% \pm 0,37% |

Tabela D.12: Desconexão na falha para CT1 e CT3, rede KL-15 e factores de carga entre 2 e 5 – Tipos de tabela TT4 (RL4/RG2) e TT9 (RG1/RG2).

TT4 (RL4/RG2) e TT9 (RG1/RG2), para CT1 e CT3 e para os factores de carga entre 2 e 5.

Nas Tabelas D.10, D.11 e D.12 apresentam-se exemplos que mostram que não se pode afirmar que existem diferenças entre os resultados quando se utiliza em CT0 o esquema RL4 ou RG1, uma vez que existem sobreposições em todos os intervalos de confiança.

A Tabela D.13 apresenta a probabilidade de rejeição de pedidos no estabelecimento para o tipo de tabela TT2 (RL4/PL2), para CT1 e CT2 e para os factores de carga entre 1 e 4. Resultados para a mesma medida também são apresentados na Tabela D.14, mas para o tipo de tabela TT4 (RL4/RG2), para CT1 e CT2 e para os factores de carga entre 2 e 5. A Tabela D.15 apresenta a probabilidade de desconexão na falha para o tipo de tabela TT4 (RL4/RG2), para CT1 e CT2 e para os factores de carga entre 2 e 5.

As Tabelas D.13, D.14 e D.15 mostram exemplos em que se verifica que os resultados para CT1 e CT2 são semelhantes (sobreposições de todos os intervalos de confiança).

| | CT1 | CT2 |
|---|----------------|----------------|
| 1 | 0,63% ± 0,02% | 0,63% ± 0,01% |
| 2 | 6,68% ± 0,04% | 6,68% ± 0,03% |
| 3 | 18,59% ± 0,05% | 18,59% ± 0,05% |
| 4 | 27,51% ± 0,07% | 27,53% ± 0,07% |

Tabela D.13: Bloqueio no estabelecimento para CT1 e CT2, rede KL-15 e factores de carga entre 1 e 4 – Tipo de tabela TT2 (RL4/PL2).

| | CT1 | CT2 |
|---|----------------|----------------|
| 2 | 0,11% ± 0,00% | 0,11% ± 0,00% |
| 3 | 1,55% ± 0,02% | 1,55% ± 0,02% |
| 4 | 6,50% ± 0,04% | 6,51% ± 0,03% |
| 5 | 12,82% ± 0,05% | 12,85% ± 0,05% |

Tabela D.14: Bloqueio no estabelecimento para CT1 e CT2, rede KL-15 e factores de carga entre 2 e 5 – Tipo de tabela TT4 (RL4/RG2).

| | CT1 | CT2 |
|---|----------------|----------------|
| 2 | 2,64% ± 0,16% | 2,66% ± 0,16% |
| 3 | 14,88% ± 0,35% | 14,43% ± 0,29% |
| 4 | 28,11% ± 0,59% | 28,32% ± 0,59% |
| 5 | 36,69% ± 0,62% | 36,56% ± 0,55% |

Tabela D.15: Desconexão na falha para CT1 e CT2, rede KL-15 e factores de carga entre 2 e 5 – Tipo de tabela TT4 (RL4/RG2).

D.2 Construção de uma tabela de esquema por CT – resultados adicionais

Nesta secção são apresentados resultados adicionais para a experiência da Secção 6.5.2 do Capítulo 6.

A Figura D.1 mostra a percentagem de pedidos desconectados na falha para CT1, para vários factores de carga, para a rede KL-15.

Na Figura D.2 mostra-se o bloqueio no estabelecimento para CT1 (intervalos de confiança na Tabela D.16), para as tabelas TT1 (PL1), TT2 (PL2), TT3 (PG1), TT4 (RG2), TT5 (RL5), TT11 (PL3), TT13 (PG3), TT63 e TT67, para o factor de carga 1, 2, 3 e 4, para a rede COST 239.

A Figura D.3 mostra o comprimento do CA para CT2 (intervalos de confiança na Tabela D.17), para as tabelas TT2 (PL2), TT4 (RG2), TT5 (RL5) e TT63, para o factor de carga 1, 2, 3 e 4, para a rede COST 239.

As Figuras D.4 e D.5 mostram, respectivamente, o bloqueio no estabelecimento

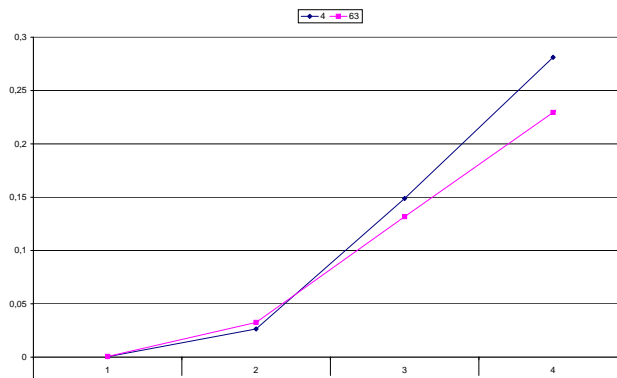


Figura D.1: Desconexão na falha para CT1, em função do factor de carga (rede KL-15) – “mono-esquema” versus “multi-esquema”.

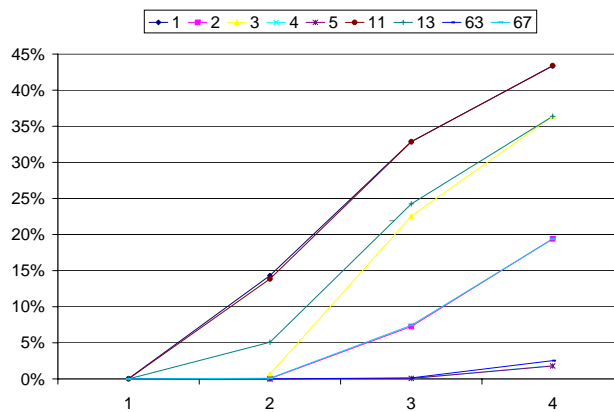


Figura D.2: Bloqueio no estabelecimento para CT1, em função do factor de carga (rede COST 239) – “mono-esquema” versus “multi-esquema”.

| Tipo de tabela | Carga 1 | Carga 2 | Carga 3 | Carga 4 |
|----------------|---------------|----------------|----------------|----------------|
| 5 | | 0,00% ± 0,00% | 0,07% ± 0,00% | 1,79% ± 0,03% |
| 4 | | 0,00% ± 0,00% | 0,06% ± 0,00% | 1,84% ± 0,04% |
| 63 | 0,00% ± 0,00% | 0,00% ± 0,00% | 0,14% ± 0,00% | 2,53% ± 0,05% |
| 67 | 0,00% ± 0,00% | 0,09% ± 0,01% | 7,43% ± 0,05% | 19,37% ± 0,07% |
| 2 | | 0,04% ± 0,00% | 7,27% ± 0,05% | 19,42% ± 0,07% |
| 13 | 0,00% ± 0,00% | 5,08% ± 0,06% | 24,25% ± 0,08% | 36,37% ± 0,08% |
| 3 | | 0,65% ± 0,03% | 22,55% ± 0,07% | 36,39% ± 0,08% |
| 1 | 0,05% ± 0,00% | 14,32% ± 0,09% | 32,88% ± 0,03% | 43,36% ± 0,05% |
| 11 | 0,02% ± 0,00% | 13,86% ± 0,08% | 32,85% ± 0,05% | 43,40% ± 0,06% |

Tabela D.16: Bloqueio no estabelecimento para CT1, rede COST 239 e factores de carga entre 1 e 4 – “mono-esquema” versus “multi-esquema”.

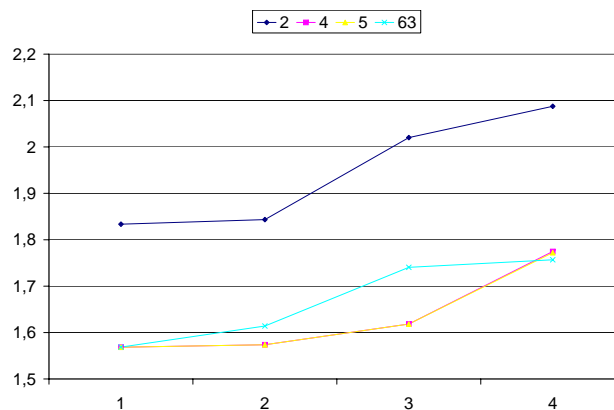


Figura D.3: Número de ramos do CA para CT2, em função do factor de carga (rede COST 239) – “mono-esquema” versus “multi-esquema”.

| Tipo de tabela | Carga 1 | Carga 2 | Carga 3 | Carga 4 |
|----------------|---------------|---------------|---------------|---------------|
| 63 | 1,569 ± 0,001 | 1,614 ± 0,001 | 1,741 ± 0,001 | 1,757 ± 0,001 |
| 5 | 1,568 ± 0,001 | 1,574 ± 0,001 | 1,618 ± 0,001 | 1,772 ± 0,002 |
| 4 | 1,568 ± 0,001 | 1,574 ± 0,001 | 1,618 ± 0,001 | 1,775 ± 0,002 |
| 2 | 1,834 ± 0,001 | 1,843 ± 0,001 | 2,020 ± 0,002 | 2,088 ± 0,001 |

Tabela D.17: Número de ramos do CA para CT2, rede COST 239 e factores de carga entre 1 e 4 – “mono-esquema” versus “multi-esquema”.

e o comprimento do CA, para CT3, para as tabelas TT1 (PL1), TT2 (PL2), TT3 (PG1), TT11 (PL3), TT13 (PG3), TT61, TT62, TT63 e TT64, para o factor de carga 1, 2, 3 e 4, para a rede COST 239. As Tabelas D.18 e D.19 apresentam os intervalos de confiança respectivos.

A Figura D.1 mostra a percentagem de pedidos desconectados na falha para CT1, para vários factores de carga, para a rede COST 239.

D.3 Construção de uma tabela por comprimento esperado de CA – resultados adicionais

Nesta secção são apresentados resultados adicionais para a experiência da Secção 6.5.3 do Capítulo 6.

As Figuras D.7, D.8 e D.9 apresentam resultados para as mesmas medidas que as Figuras 6.7, 6.8 e 6.9 do Capítulo 6, a única diferença é o tipo de tabelas utilizado.

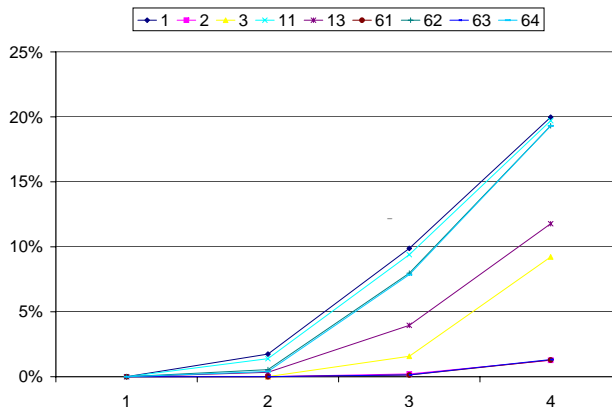


Figura D.4: Bloqueio no estabelecimento para CT3, em função do factor de carga (rede COST 239) – “mono-esquema” versus “multi-esquema”.

| Tipo de tabela | Carga 1 | Carga 2 | Carga 3 | Carga 4 |
|----------------|---------------|---------------|---------------|----------------|
| 2 | | 0,00% ± 0,00% | 0,21% ± 0,01% | 1,26% ± 0,03% |
| 61 | 0,00% ± 0,00% | 0,00% ± 0,00% | 0,13% ± 0,01% | 1,29% ± 0,02% |
| 63 | 0,00% ± 0,00% | 0,00% ± 0,00% | 0,14% ± 0,00% | 1,32% ± 0,02% |
| 3 | | 0,01% ± 0,00% | 1,58% ± 0,02% | 9,22% ± 0,09% |
| 13 | 0,00% ± 0,00% | 0,34% ± 0,01% | 3,95% ± 0,03% | 11,77% ± 0,08% |
| 64 | 0,00% ± 0,00% | 0,41% ± 0,01% | 7,86% ± 0,04% | 19,31% ± 0,09% |
| 62 | 0,00% ± 0,00% | 0,55% ± 0,02% | 7,97% ± 0,05% | 19,31% ± 0,08% |
| 11 | 0,00% ± 0,00% | 1,39% ± 0,02% | 9,40% ± 0,05% | 19,71% ± 0,10% |
| 1 | 0,00% ± 0,00% | 1,74% ± 0,03% | 9,88% ± 0,04% | 19,97% ± 0,07% |

Tabela D.18: Bloqueio no estabelecimento para CT3, rede COST 239 e factores de carga entre 1 e 4 – “mono-esquema” versus “multi-esquema”.

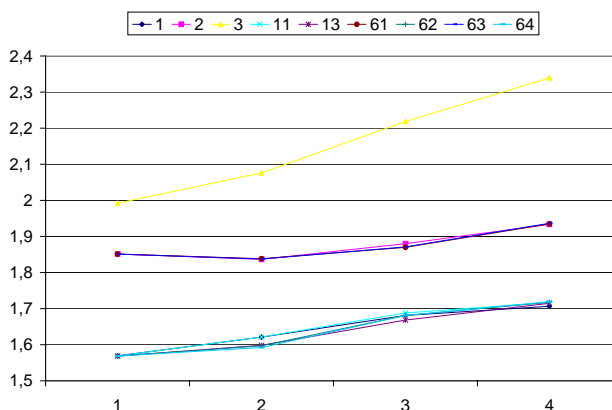


Figura D.5: Número de ramos do CA para CT3, em função do factor de carga (rede COST 239) – “mono-esquema” versus “multi-esquema”.

| Tipo de tabela | Carga 1 | Carga 2 | Carga 3 | Carga 4 |
|----------------|---------------|---------------|---------------|---------------|
| 1 | 1,569 ± 0,001 | 1,621 ± 0,001 | 1,681 ± 0,001 | 1,707 ± 0,001 |
| 64 | 1,569 ± 0,001 | 1,592 ± 0,001 | 1,681 ± 0,001 | 1,719 ± 0,001 |
| 62 | 1,569 ± 0,001 | 1,597 ± 0,001 | 1,681 ± 0,001 | 1,718 ± 0,001 |
| 11 | 1,569 ± 0,001 | 1,621 ± 0,001 | 1,688 ± 0,001 | 1,717 ± 0,001 |
| 13 | 1,569 ± 0,001 | 1,599 ± 0,001 | 1,668 ± 0,001 | 1,715 ± 0,001 |
| 63 | 1,851 ± 0,002 | 1,838 ± 0,001 | 1,871 ± 0,001 | 1,936 ± 0,001 |
| 2 | 1,851 ± 0,001 | 1,837 ± 0,001 | 1,880 ± 0,001 | 1,933 ± 0,001 |
| 61 | 1,851 ± 0,001 | 1,838 ± 0,001 | 1,870 ± 0,001 | 1,935 ± 0,001 |
| 3 | 1,992 ± 0,001 | 2,076 ± 0,001 | 2,218 ± 0,002 | 2,339 ± 0,002 |

Tabela D.19: Número de ramos do CA para CT3, rede COST 239 e factores de carga entre 1 e 4 – “mono-esquema” versus “multi-esquema”.

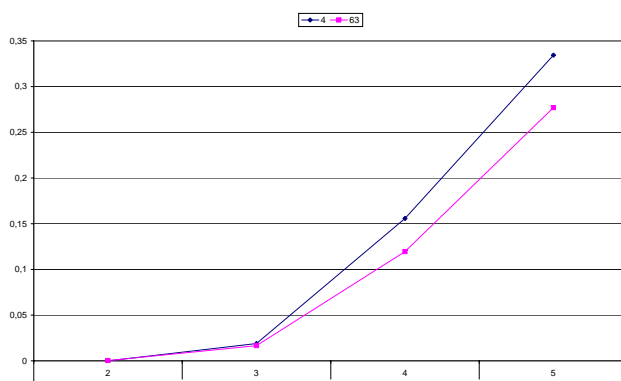


Figura D.6: Bloqueio na falha para CT1, em função do factor de carga (rede COST 239) – “mono-esquema” versus “multi-esquema”.

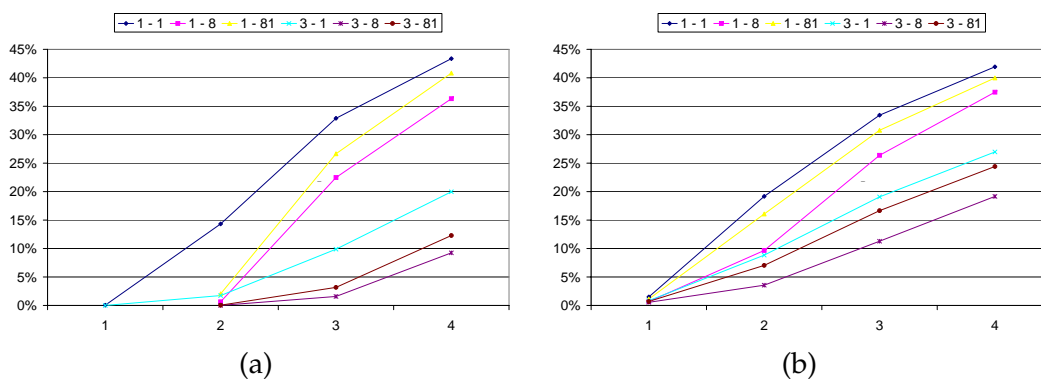


Figura D.7: Bloqueio no estabelecimento para CT1 e CT3 – “mono-esquema” versus misto por comprimento de CA (PG1-PL1): (a) Rede COST 239. (b) Rede KL-15.

D.4 Construção de uma tabela por carga da rede – resultados adicionais

Nesta secção são apresentados resultados adicionais para a experiência da Secção 6.5.4 do Capítulo 6.

As Figuras D.10, D.11, D.12, D.13, D.14, D.15, D.16 e D.17 apresentam resultados para as mesmas medidas que as Figuras 6.12, 6.13, 6.14, 6.15, 6.16, 6.17, 6.18 e 6.19 do Capítulo 6, a única diferença é o CT a que se referem, no Capítulo 6 foram apresentados resultados para CT3 e aqui são apresentados resultados para CT1 (os resultados para CT2 são idênticos aos obtidos para CT1).

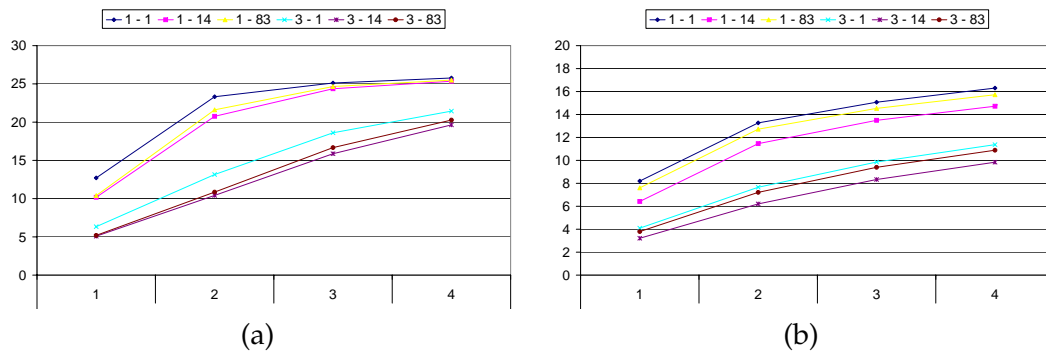


Figura D.8: Média para todos os ramos da LB gasta, para CT1 e CT3 – “mono-esquema” versus misto por comprimento de CA (PG3-PL1): (a) Rede COST 239. (b) Rede KL-15.

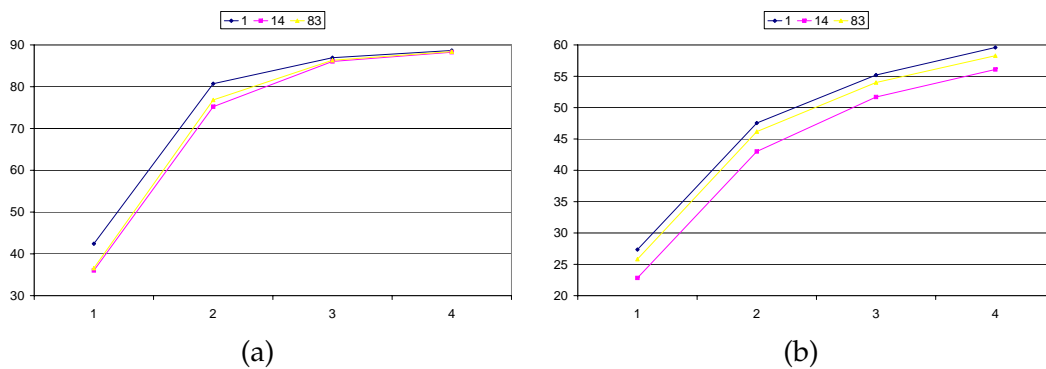


Figura D.9: Média para todos os ramos da LB gasta, por todos os CT – “mono-esquema” versus misto por comprimento de CA (PG3-PL1): (a) Rede COST 239. (b) Rede KL-15.

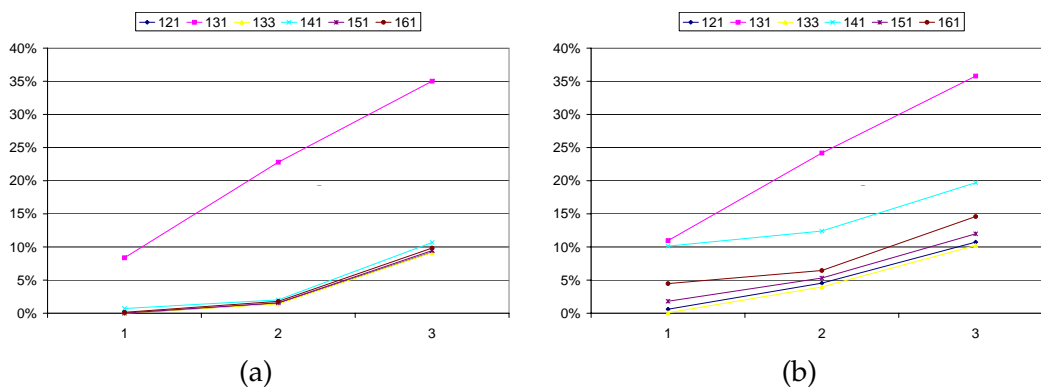


Figura D.10: Bloqueio no estabelecimento para CT1, em função do factor de carga – “mono-esquema” versus misto por carga da rede (PL3-RL5): (a) Rede COST 239. (b) Rede KL-15.

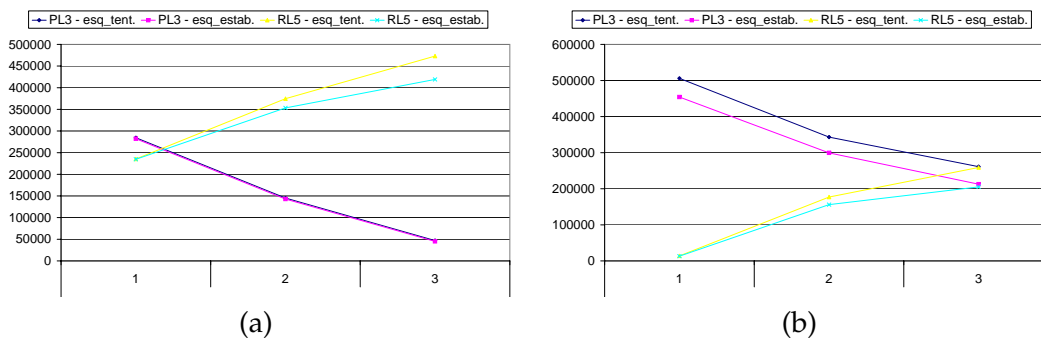


Figura D.11: Número de pedidos tentados versus estabelecidos, por esquema, para CT1, em função do factor de carga (rede KL-15) – misto por carga da rede (PL3-RL5): (a) Limiar de carga = 40. (b) Limiar de carga = 70.

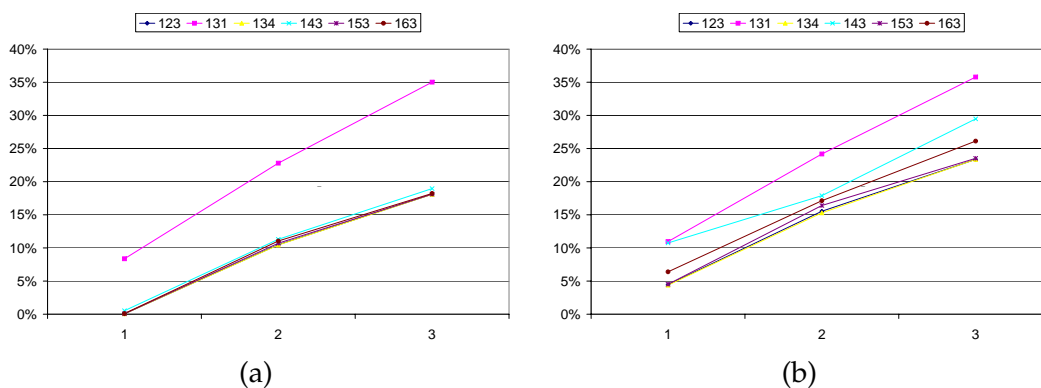


Figura D.12: Bloqueio no estabelecimento para CT1, em função do factor de carga – “mono-esquema” versus misto por carga da rede (PL3-PL2): (a) Rede COST 239. (b) Rede KL-15.

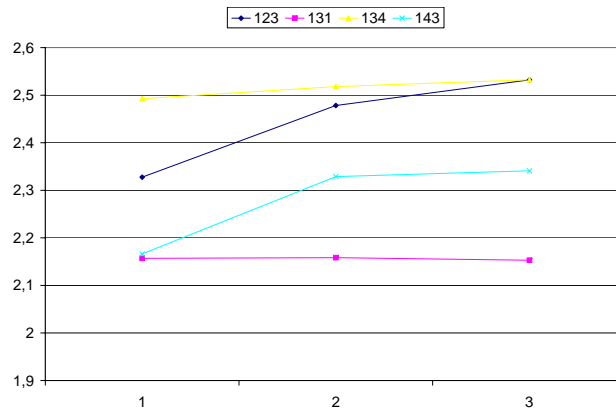


Figura D.13: Número de ramos do CA, para CT1, em função do factor de carga (rede KL-15) – “mono-esquema” versus misto por carga da rede (PL3-PL2)

D.5 Várias tabelas e combinação de condições – resultados adicionais

Nesta secção são apresentados resultados adicionais para a experiência da Secção 6.5.5 do Capítulo 6.

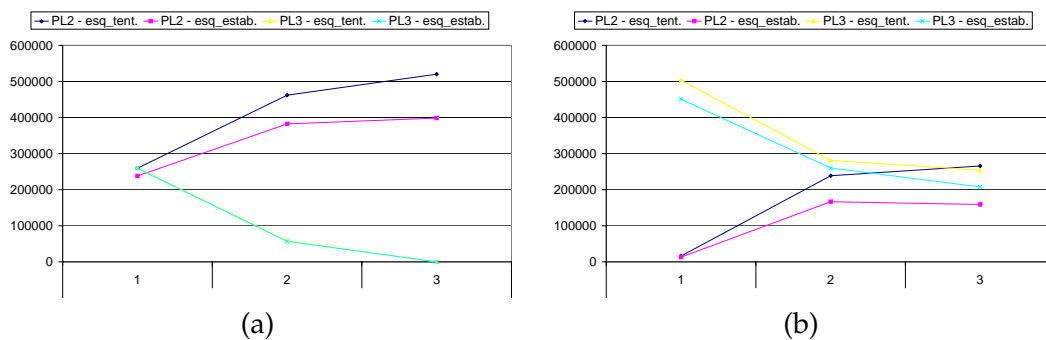


Figura D.14: Número de pedidos tentados versus estabelecidos, por esquema, para CT1, em função do factor de carga (rede KL-15) – misto por carga da rede (PL3-PL2): (a) Limiar de carga = 40. (b) Limiar de carga = 70.

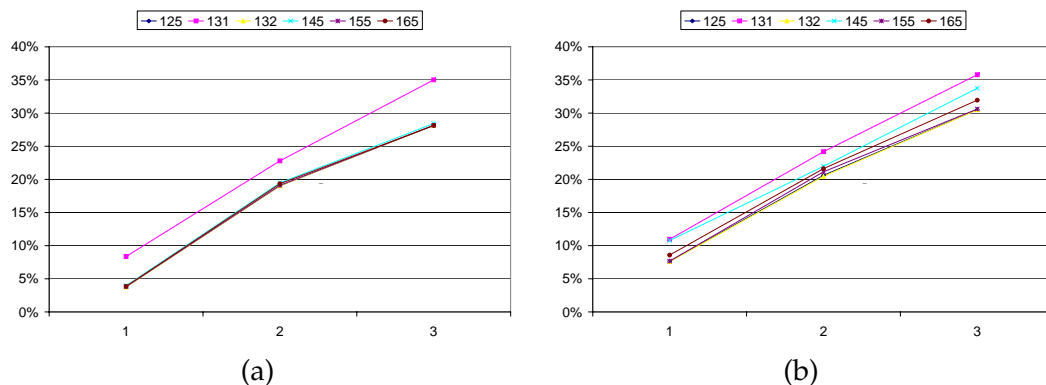


Figura D.15: Bloqueio no estabelecimento para CT1, em função do factor de carga – “mono-esquema” versus misto por carga da rede (PL3-PG3): (a) Rede COST 239. (b) Rede KL-15.

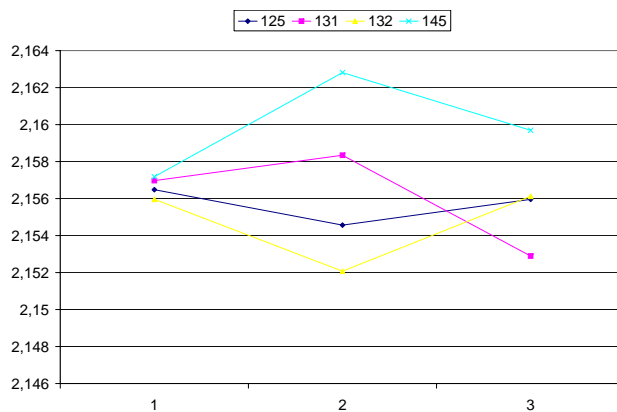


Figura D.16: Número de ramos do CA, para CT1, em função do factor de carga (rede KL-15) – “mono-esquema” versus misto por carga da rede (PL3-PG3)

As Figuras D.18, D.19, D.20, D.21, D.22, D.23 e D.24, apresentam os resultados para as mesmas medidas que as Figuras 6.20, 6.21, 6.22, 6.23, 6.24, 6.25 e 6.26 do Capítulo 6 a única diferença é a rede a que se referem, no Capítulo 6 foram apresentados resultados para a rede KL-15 e aqui são apresentados os resultados para a rede COST 239.

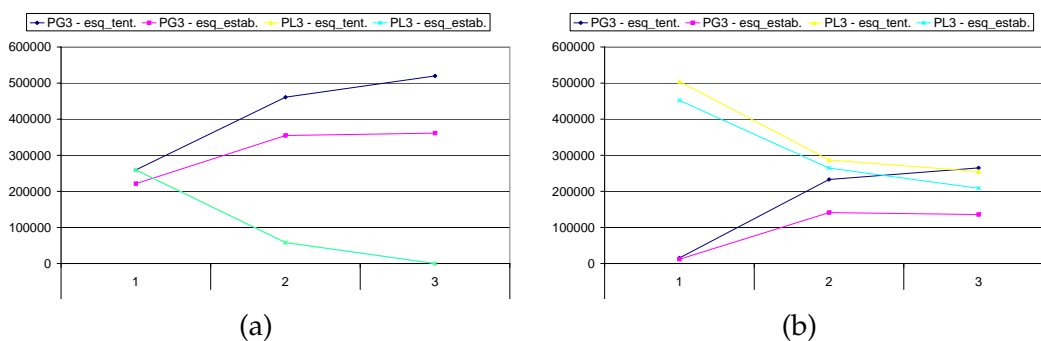


Figura D.17: Número de pedidos tentados versus estabelecidos, por esquema, para CT1, em função do factor de carga (rede KL-15) – misto por carga da rede (PL3-PG3): (a) Limiar de carga = 40. (b) Limiar de carga = 70.

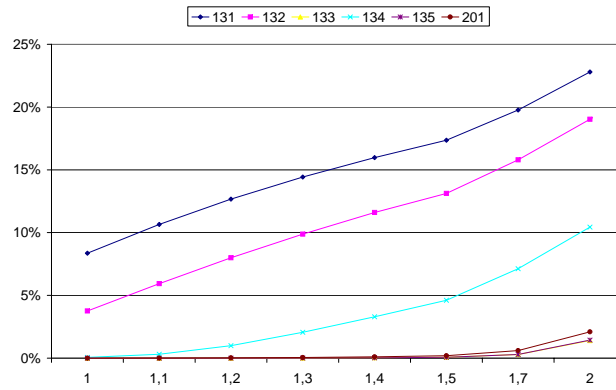


Figura D.18: Bloqueio no estabelecimento para CT1, em função do factor de carga (rede COST 239) – “mono-esquema” versus exemplo.

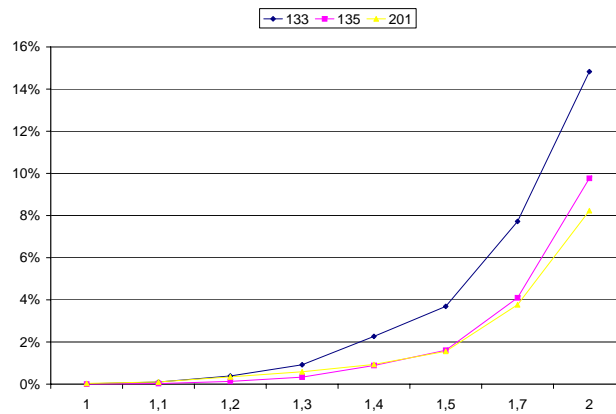


Figura D.19: Bloqueio na falha para CT1, em função do factor de carga (rede COST 239) – “mono-esquema” versus exemplo.

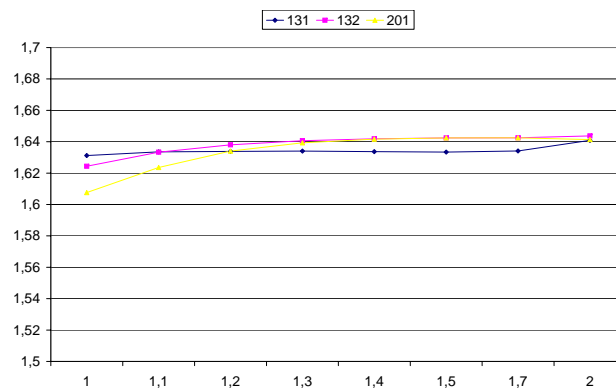


Figura D.20: Número de ramos do CA para CT2, em função do factor de carga (rede COST 239) – “mono-esquema” versus exemplo.

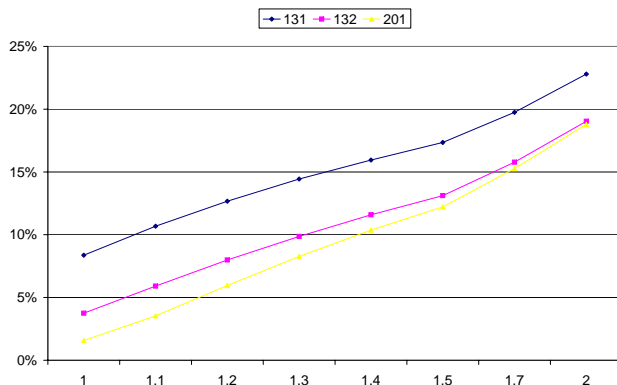


Figura D.21: Bloqueio no estabelecimento para CT2, em função do factor de carga (rede COST 239) – “mono-esquema” versus exemplo.

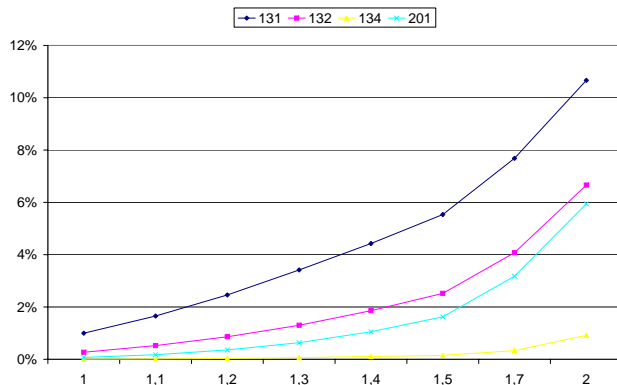


Figura D.22: Bloqueio no estabelecimento para CT3, em função do factor de carga (rede COST 239) – “mono-esquema” versus exemplo.

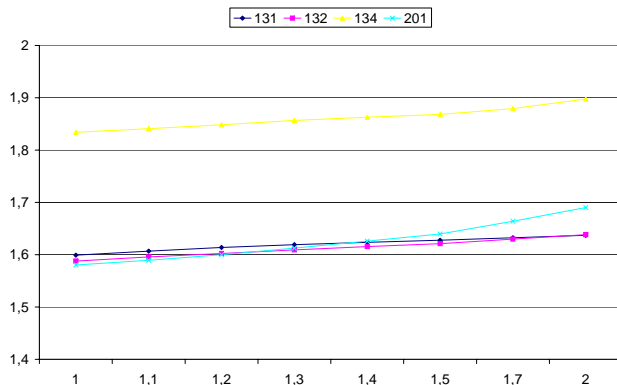


Figura D.23: Número de ramos do CA para CT2, em função do factor de carga (rede COST 239) – “mono-esquema” versus exemplo.

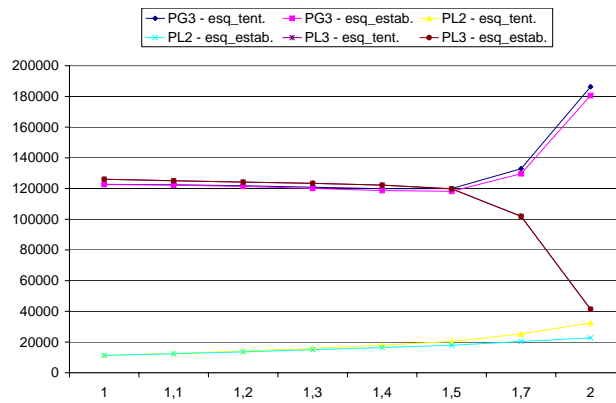


Figura D.24: Número de pedidos tentados versus estabelecidos, por esquema, para CT3, em função do factor de carga no exemplo (rede COST 239).

E Cálculo de Frequências Variáveis

A fórmula da média harmónica permite calcular a frequência H equivalente à media de um conjunto de frequências x_i , ou seja a frequência que corresponde ao inverso da média aritmética dos tempos entre pedidos:

$$H = \frac{1}{\frac{\sum_{i=1}^n \frac{1}{x_i}}{n}} = \frac{n}{\sum_{i=1}^n \frac{1}{x_i}} \text{ onde } x_i > 0 \quad (\text{E.1})$$

O que com apenas 2 frequências se pode simplificar para:

$$H = \frac{2x_1x_2}{x_1 + x_2} \quad (\text{E.2})$$

ou usando a notação do Capítulo 6.5:

$$\lambda = \frac{2\lambda_{high}\lambda_{low}}{\lambda_{high} + \lambda_{low}} \quad (\text{E.3})$$

de notar que podemos exprimir também a intensidade como o produto de um factor de carga por λ_{base} :

$$\lambda = \text{FACTOR_CARGA} * \lambda_{base} \quad (\text{E.4})$$

É fácil mostrar que existe um grande número de pares $(\lambda_{high}, \lambda_{low})$ a que corresponde uma mesma intensidade λ . Na prática, pretende-se fazer a simulação de forma a que a intensidade de pedidos em situações de carga leve (λ_{low}) não seja muito reduzida e que a intensidade de pedidos em situações de carga pesada (λ_{high}) não apresente uma sobrecarga exagerada para a rede. Para parametrizar estes valores pode-se escolher qual a distância pretendida entre λ_{high} e λ_{low} , que será designada de amplitude de frequências. Para permitir diferenças significativas mesmo quando a intensidade de carga for elevada (factor de carga elevado), vamos exprimir a amplitude de frequências como o produto de um factor de amplitude (f_a) pela intensidade escolhida (λ ou seja $\lambda_{base} * \text{factor_carga}$):

$$amplitude = f_{-a} * \lambda = f_{-a} * FACTOR_CARGA * \lambda_{base} \quad (E.5)$$

É possível verificar que a um determinado par (λ, f_{-a}) corresponde apenas um par $(\lambda_{high}, \lambda_{low})$ de valores positivos, tal que:

$$\begin{cases} f_{-a} * \lambda = \lambda_{high} - \lambda_{low} \\ \lambda = \frac{2\lambda_{high}\lambda_{low}}{\lambda_{high} + \lambda_{low}} \end{cases} \quad (E.6)$$

Se definirmos duas variáveis positivas, a e b tais que $\lambda_{high} = \lambda + a$ e $\lambda_{low} = \lambda - b$, podemos reescrever o sistema de equações anterior usando a e b como:

$$\begin{cases} f_{-a} * \lambda = a + b \\ \lambda = \frac{2(\lambda + a)(\lambda - b)}{2\lambda + a - b} \end{cases} \quad (E.7)$$

Das equações anteriores retira-se que:

$$b = \frac{a\lambda}{\lambda + 2a} \quad (E.8)$$

Reescrevendo a primeira equação do sistema (E.7) considerando a equação (E.8) ficamos com:

$$f_{-a} = \frac{a}{\lambda} + \frac{a}{\lambda + 2a} \quad (E.9)$$

Considerando uma variável auxiliar $y = a/\lambda$ na equação anterior, obtém-se:

$$2y^2 + (2 - 2f_{-a})y - f_{-a} = 0 \quad (E.10)$$

Resolvendo a equação quadrática (E.10) em ordem a y , vem:

$$y = \frac{-(2 - 2f_{-a}) + \sqrt{(2 - 2f_{-a})^2 + 8f_{-a}}}{4} \quad (E.11)$$

A partir do anterior podemos obter a e b , e a partir destes λ_{high} e λ_{low} .

F Intervalos de confiança

Uma questão importante na análise de resultados de simulação é a determinação dos intervalos de confiança dos valores estimados, o que se torna particularmente ainda mais relevante quando se pretendem comparar valores de desempenho próximos entre si.

Neste apêndice é descrita a forma utilizada na determinação dos intervalos de confiança das estimativas obtidas, para as diversas medidas do modelo.

Se usarmos o parâmetro θ para designar uma das medidas do sistema, o resultado de um conjunto de simulações será uma estimativa $\hat{\theta}$ de θ . O grau de incerteza acerca dos parâmetros que se pretende estimar pode ser descrito quantitativamente através de técnicas estatísticas. A determinação de intervalos de confiança é uma delas.

O método usado para estimar as diversas medidas pelo programa de simulação foi o método das replicações independentes. Este método consiste em repetir a simulação n vezes (n replicações) cada uma das quais usando sequências de números aleatórios diferente e considerando apenas as observações após o período de estabilização.

Obtém-se uma amostra de dimensão n de uma população com distribuição normal de variância (σ^2) e média (μ) desconhecidas. A partir desta amostra vamos estimar o parâmetro θ e obter o intervalo de confiança para essa estimativa. A amplitude do intervalo estimado é uma medida da precisão ou exactidão da estimativa pontual.

Seja Y_j uma medida tomada na j -ésima replicação. Considerando que Y_1, Y_2, \dots, Y_n são variáveis aleatórias independentes e identicamente distribuídas (Law e Kelton, 1991), podem-se aplicar os métodos clássicos para estimação pontual e intervalar do parâmetro θ . A estimativa pontual de θ , $\hat{\theta}$, é igual à média aritmética da amostra, \bar{Y} , isto é:

$$\hat{\theta} = \bar{Y} = \frac{\sum_{j=1}^n Y_j}{n} \quad (\text{F.1})$$

A exactidão do estimador, $\hat{\theta}$, pode ser medida através da variância de $\hat{\theta}$. A variação das realizações, Y_j , em relação à média da amostra dá informação acerca

da dispersão dessas realizações em relação à média da amostra, \bar{Y} . A variância da amostra, s^2 (que é um estimador não enviesado da variância σ^2), define-se por (Dudewicz e Mishra, 1988):

$$s^2 = \frac{1}{n-1} \sum_{j=1}^n (Y_j - \bar{Y})^2 \quad (\text{F.2})$$

O cálculo de intervalos de confiança baseados na distribuição t de Student pode ser efectuado uma vez que se consideram os Y_1, Y_2, \dots, Y_n aproximadamente independentes e aproximadamente normais. Deseja determinar-se ϵ tal que:

$$\text{Prob} [|\bar{Y} - \mu| \leq \epsilon] = 1 - \alpha \quad (\text{F.3})$$

em que $1 - \alpha$ é o grau de confiança.

Passando à variável normal reduzida Z , $N(0, 1)$:

$$Z = \frac{\bar{Y} - \mu}{\sigma/\sqrt{n}} \quad (\text{F.4})$$

com

$$\epsilon' = \frac{\epsilon}{\sigma\sqrt{n}} \quad (\text{F.5})$$

vem:

$$\text{Prob} [|Z| \leq \epsilon'] = 1 - \alpha \quad (\text{F.6})$$

Como a variância é desconhecida vamos ter que a substituir pela sua estimativa s^2 . Temos que utilizar a variável aleatória T tal que:

$$T = \frac{\bar{Y} - \mu}{s/\sqrt{n}} \quad (\text{F.7})$$

em vez da variável Z .

A variável aleatória T segue a distribuição t de Student com $n - 1$ graus de liberdade (Kobayashi, 1978). Rescrevendo a equação F.6, de acordo com as condições presentes:

$$\text{Prob} \left[\left| \frac{\bar{Y} - \mu}{s/\sqrt{n}} \right| \leq \epsilon' \right] = 1 - \alpha \quad (\text{F.8})$$

Considerando que $t_{f,\alpha}$ é o ponto de percentagem $100(1 - \alpha)\%$ de uma distribuição de t de Student com f graus de liberdade. Resolvendo a desigualdade F.8 em ordem a μ obtém-se:

$$\text{Prob} \left[\bar{Y} - t_{n-1; \alpha/2} \times \frac{s}{\sqrt{n}} \leq \mu \leq \bar{Y} + t_{n-1; \alpha/2} \times \frac{s}{\sqrt{n}} \right] = 1 - \alpha \quad (\text{F.9})$$

O intervalo de confiança com grau de confiança $1 - \alpha$ é

$$\left[\bar{Y} - t_{n-1; \alpha/2} \times \frac{s}{\sqrt{n}}, \bar{Y} + t_{n-1; \alpha/2} \times \frac{s}{\sqrt{n}} \right] \quad (\text{F.10})$$

G Lista de Acrónimos e Siglas

AF *Assured Forwarding*

BC *Bandwidth Constraint*

BCM *Bandwidth Constraints Model*

BE *Best Effort*

CA *Caminho Activo (Active Path)*

CBR *Constraint-based routing*

CR *Caminho de Recuperação (Recovery Path)*

CR-LDP *Constraint-based Routing – Label Distribution Protocol*

CT *Class Type*

DiffServ *Differentiated Services*

DSCP *DiffServ Code Point*

DS-TE *Diff-Serv-aware Traffic Engineering*

k **DSP** *k-Disjoint Shortest Paths*

EF *Expedited Forwarding*

E-LSP *EXP-Inferred-PHB Scheduling Class LSP*

EXP *Experimental*

FRR *Fast Reroute*

FEC *Forwarding Equivalence Class*

FI *Full Information*

FIS *Fault Indication Signal*

- FRS** *Fault Recovery Signal*
- FSM** *Fully Shared Mechanism*
- GNED** *Graphical Network EDitor*
- GMPLS** *Generalized Multiprotocol Label Switching*
- IETF** *Internet Engineering Task Force*
- IGP** *Interior Gateway Protocol*
- IP** *Internet Protocol*
- IS-IS** *Itermediate System – to – Itermediate System*
- ISP** *Internet Service Provider*
- LB** *Largura de Banda*
- LBP** *Largura de Banda de Protecção*
- LDP** *Label Distribution Protocol*
- L-LSP** *Label-inferred-PHB Scheduling Class LSP*
- LSA** *Link-State Advertisement*
- LSP** *Label Switched Path*
- LSR** *Label Switching Router*
- MAM** *Maximum Allocation Bandwidth Constraints Model*
- MAR** *Max Allocation with Reservation Bandwidth Constraints Model*
- MPLS** *MultiProtocol Label Switching*
- MTBF** *Mean Time Between Failures*
- MTTR** *Mean Time to Repair*
- NED** *NEtwork Description*
- NI** *No Information*
- oAIS** *optimized Aggregate Information Scenario*
- O-D** *Origem-Destino*

- OMNeT++** *Object-oriented Modular discrete event NeTwork simulator*
- OSPF** *Open Shortest Path First*
- PHB** *Per-Hop-Behavior*
- PI** *Partial Information*
- PLR** *Point of Local Repair*
- PML** *Path Merge LSR*
- POR** *Point of Repair*
- PPG** *Protected Path Group*
- PSC** *PHB Scheduling Class*
- PSL** *Path Switch LSR*
- PSM** *Partially Shared Mechanism*
- QoS** *Quality of Service*
- RDM** *Russian Dolls Bandwidth Constraints Model*
- RFC** *Request For Comments*
- RIP** *Routing Information Protocol*
- RNT** *Reverse Notification Tree*
- RSVP** *Resource Reservation Protocol*
- RSVP-TE** *Resource Reservation Protocol with Traffic Engineering*—Neste trabalho chamamos protocolo RSVP-TE à conjugação do protocolo RSVP (Braden *et al.*, 1997) com as extensões RSVP-TE (Awduche *et al.*, 2001)
- SDH** *Synchronous Digital Hierarchy*
- SLA** *Service Level Agreement*
- SONET** *Synchronous Optical Network*
- SPM** *Self-Protecting Multi-Paths*
- SRLG** *Shared Risk Link Group*
- TA** *Traffic Aggregate*

TE *Traffic Engineering*

TT *Traffic Trunk*

TCP *Transmission Control Protocol*

UDP *User Datagram Protocol*

WDM *Wavelength Division Multiplexing*

XML *eXtensible Markup Language*

Bibliografia

- Aamodt, A. e E. Plaza (1994). Case-based reasoning: Foundational Issues, Methodological Variations, and System Approaches. *AI Communications*, 7(1):39–59.
- Ahn, G., J. Jang e W. Chun (2002). An Efficient Rerouting Scheme for MPLS-Based Recovery and Its Performance Evaluation. *Telecommunication Systems*, 19(3–4):481–495.
- Andersson, L., P. Doolan, N. Feldman, A. Fredette e B. Thomas (2001). LDP specification. IETF RFC 3036.
- Apostolopoulos, G., R. Guérin, S. Kamat e S. K. Tripathi (1998). Quality of service based routing: a performance perspective. *SIGCOMM Comput. Commun. Rev.*, 28(4):17–28.
- Ash, J. (2005). Max Allocation with Reservation Bandwidth Constraints Model for Diffserv-aware MPLS Traffic Engineering & Performance Comparisons. IETF RFC 4126.
- Ash, J., M. Girish, E. Gray, Sandburst, B. Jamoussi e G. Wright (2002). Applicability Statement for CR-LDP. IETF RFC 3213.
- Autenrieth, A. e A. Kirstadter (2002). Engineering End-to-End IP Resilience Using Resilience-Differentiated QoS. *IEEE Communications Magazine*, 40(1):50–57.
- Awduche, D., L. Berger, D. Gan, T. Li, V. Srinivasan e G. Swallow (2001). RSVP-TE: Extensions to RSVP for LSP tunnels. IETF RFC 3209.
- Bartoš, R. e M. Raman (2001). A heuristic approach to service restoration in MPLS networks. Em *Proceedings of the IEEE International Conference on Communications (ICC 01)*, volume 1, págs. 117–121.
- Bartoš, R., M. Raman e A. Gandhi (2001). New approaches to service restoration in MPLS-based networks. Em *Proceedings of the International Conference on Trends in Communications (EUROCON 01)*, volume 1, págs. 58–61.
- Batchelor, P., B. Daino, P. Heinzmann, D. R. Hjélme, R. Inkret, H. A. Jäger, M. Join-dot, A. Kuchar, E. L. Coquil, P. Leuthold, G. D. Marchis, F. Matera, B. Mikac, H.-P. Nolting, J. Späth, F. Tillerot, B. V. Caenegem, N. Wauters e C. Weinert (2000). Study on the implementation of optical transparent transport networks in the european environment – results of the research project COST 239. *Photonic Network Communications*, 2(1):15–32.

- Borner, K. (1994). Structural Similarity as a guidance in case-based design. Em *Proceedings of the European Workshop on Case-Based Reasoning (EWCBR 94)*, págs. 197–208.
- Braden, R., L. Zhang, S. Berson, S. Herzog e S. Jamin (1997). Resource reservation protocol (RSVP) – version 1 functional specification. IETF RFC 2205.
- Calle, E. (2004). *Enhanced fault recovery methods for protected traffic services in GMPLS networks*. Tese de Doutoramento, Department of Electronics, Computer Science and Automatic Control – Universitat de Girona, Girona, Spain.
- Calle, E., J. Marzo, A. Urrea e P. Vila (2003). Enhancing MPLS QoS routing algorithms by using the Network Protection Degree paradigm. Em *Proceedings of the IEEE GLOBAL COMMUNICATIONS Conference (GLOBECOM 03)*, págs. 3053–3057.
- Calle, E., J. L. Marzo e A. Urrea (2004). Protection performance components in MPLS networks. *Computer Communications*, 27:1220–1228.
- Dana, A., A. K. Zadeh, K. Badie, M. E. Kalantari e N. Reyhani (2003). LSP Restoration in MPLS Network Using Case-Based Reasoning Approach. Em *Proceedings of the International Conference on Communication Technology (ICCT 03)*, págs. 462–468.
- Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271.
- Dong, S., C. Phillips e R. Friskney (2003). Differentiated-resilience provisioning in wavelength-routed optical networks. Em *Proceedings of the 18th International Teletraffic Congress (ITC 18)*, volume 5b, págs. 921–930.
- Dudewicz, E. J. e S. N. Mishra (1988). *Modern Mathematical Statistics*.
- Faucheur, F. L. (2005a). Maximum Allocation Bandwidth Constraints Model for Diffserv-aware MPLS Traffic Engineering. IETF RFC 4125.
- Faucheur, F. L. (2005b). Requirements for Support of Differentiated Services-aware MPLS Traffic Engineering. IETF RFC 4124.
- Faucheur, F. L. (2005c). Russian Dolls Bandwidth Constraints Model for Diffserv-aware MPLS Traffic Engineering. IETF RFC 4127.
- Faucheur, F. L. e W. Lai (2003). Requirements for Support of Differentiated Services-aware MPLS Traffic Engineering. IETF RFC 3564.
- Faucheur, F. L., L. Wu, B. Davie, S. Davari, P. Vaananen, R. Krishnan, P. Chevval e J. Heinanen (2002). Multi-Protocol Label Switching (MPLS) Support of Differentiated Services. IETF RFC 3270.
- Gamma, E., R. Helm, R. Johnson e J. M. Vlissides (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional, illustrated edition ed. ISBN 0201633612.

- Gomes, T., P. Nunes e L. Jorge (2005). Explorando a recuperação de redes baseada em mecanismos do MPLS. Rel. Téc. 11, INESC - Coimbra, Coimbra, Portugal.
- Gomes, T., J. Craveirinha e L. Jorge (2006a). An effective algorithm for obtaining the minimal cost pair of disjoint paths with dual arc costs. Rel. Téc. 5, INESC - Coimbra, Coimbra, Portugal.
- Gomes, T., J. Craveirinha e L. Jorge (2006b). An effective algorithm for obtaining the minimal cost pair of disjoint paths with dual arc costs. *Computers & Operations Research*. Submetido para publicação.
- Gomes, T., J. Craveirinha e L. Jorge (2006c). Obtaining Optimal Disjoint Paths with Dual Costs in Undirected Networks. Em Soares e Zio (2006), págs. 2257–2262.
- Grover, W. D. e D. Stamatelakis (1998). Cycle-Oriented Distributed Preconfiguration: Ring-like Speed with Mesh-like Capacity for Self-planning Network Restoration. Em *Proceedings of the IEEE International Conference on Communications (ICC 98)*, págs. 537–543. Atlanta, Georgia.
- Haskin, D. e R. Krishnan (2001). A method for setting an alternative label switched paths to handle fast reroute. IETF Draft.
- Hong, D.-K., C. S. Hong e Dongsik-Yun (2004). A hierarchical restoration scheme with dynamic adjustment of restoration scope in an MPLS network. Em *Proceedings of the Network Operations and Management Symposium (NOMS 04)*, volume 1, págs. 191–204.
- Huang, C., V. Sharma, K. Owens e S. Makam (2002). Building Reliable MPLS Networks Using a Path Protection Mechanism. *IEEE Communications Magazine*, 40(3):156–162.
- Hussain, I. (2005). *Fault-Tolerant IP and MPLS Networks*. Cisco Press.
- Iselt, A., A. Kirstädter, A. Pardigon e T. Schwabe (2004). Resilient Routing Using MPLS and ECMP. Em *Proceedings of the International Workshop on High Performance Switching and Routing (HPSR 04)*. Phoenix, Arizona, USA.
- Jamoussi, B., L. Andersson, R. Callon, R. Dantu, L. Wu, P. Doolan, T. Worster, N. Feldman, A. Fredette, M. Girish, E. Gray, J. Heinanen, T. Kilty e A. Mallis (2002). Constraint-based LSP setup using LDP. IETF RFC 3212.
- Jorge, L. e T. Gomes (2005a). Recuperação em redes MPLS – uma resenha de esquemas de recuperação. Rel. Téc. 10, INESC - Coimbra, Coimbra, Portugal.
- Jorge, L. e T. Gomes (2005b). Resenha de esquemas de recuperação em redes MPLS. Em C. G. Soares, A. P. Teixeira e P. Antão (eds.), *Actas do Encontro Nacional de Riscos, Segurança e Fiabilidade*, volume 2, págs. 417 – 434. ESRA.
- Jorge, L. e T. Gomes (2006a). Multi-scheme Recovery in MPLS Networks. Em Soares e Zio (2006), págs. 2263–2271.

- Jorge, L. e T. Gomes (2006b). Survey of Recovery Schemes in MPLS Networks. Em *Proceedings of the International Conference on Dependability of Computer Systems (DepCos 06)*. Wroclaw University of Technology, Poland.
- Jorge, L. e T. Gomes (2006c). Uma versão melhorada de um algoritmo de encaminhamento para protecção local com partilha de largura de banda de protecção. Rel. Téc. 09/2006, INESC - Coimbra, Coimbra, Portugal.
- Jorge, L. e T. Gomes (2007). An on-line routing algorithm of locally protected paths with exact reservations. Em PalauSalvador, CE (ed.), *Proceedings of the 6th IASTED International Conference on Communication Systems and Networks (CSN 07)*, págs. 76–83. ISBN 978-0-88986-695-9.
- Kang, J. e M. J. Reed (2003). Bandwith Protection in MPLS Networks Using p -Cycle Structure. Em *Proceedings of Design of Reliable Communication Networks (DRCN 03)*, págs. 356–362. Banff, Alberta, Canada.
- Kar, K., M. Kodialam e T. Lakshman (2003). Routing restorable bandwidth guaranteed connections using maximum 2-route flows. *IEEE/ACM Transactions on Networking*, 11(5):772–781.
- Katz, D., K. Kompella e D. Yeung (2003). Traffic Engineering (TE) Extensions to OSPF Version 2. IETF RFC 3630.
- Kim, S.-Y. (2003). Effect of load distribution in path protection of MPLS. *International Journal of Communication Systems*, 16(4):321–335.
- Kobayashi, H. (1978). *Modelling and Analysis*.
- Kodialam, M. e T. V. Lakshman (2000). Dynamic Routing of Bandwidth Guaranteed Tunnels with Restoration. Em *Proceedings of the IEEE Conference on Computer Communications (IEEE INFOCOM 00)*, págs. 902–911.
- Kodialam, M. e T. V. Lakshman (2001). Dynamic Routing of Locally Restorable Bandwidth Guaranteed Tunnels using Aggregated Link Usage Information. Em *Proceedings of the IEEE Conference on Computer Communications (IEEE INFOCOM 01)*, págs. 376–385.
- Kodialam, M. e T. V. Lakshman (2002a). Dynamic backup routing of network tunnel paths for local restoration in a packet network, Patent US 2002/0067693A1.
- Kodialam, M. e T. V. Lakshman (2002b). Restorable Dynamic Quality of Service Routing. *IEEE Communications Magazine*, 40(6):72–81.
- Kodialam, M. e T. V. Lakshman (2003). Dynamic routing of restorable bandwidth-guaranteed tunnels using aggregated network resource usage information. *IEEE/ACM Transactions on Networking*, 11(3):399–410.
- Koubàa, M., M. Nicolas Puech e M. Gagnaire (2005). Lightpath Rerouting for Differentiated Services in WDM All-Optical Networks. Em *Proceedings of Design of Reliable Communication Networks (DRCN 05)*.

- Laborczi, P., J. Tapolcai, P. H. Ho, T. Cinkler, A. Recski e H. T. Mouftah (2001). Algorithms for asymmetrically weighted pair of disjoint paths in survivable networks. Em *Proceedings of Design of Reliable Communication Networks (DRCN 01)*, págs. 220–227.
- Lai, W. (2005). Bandwidth Constraints Models for Differentiated Services (Diffserv)-aware MPLS Traffic Engineering: Performance Evaluation. IETF RFC 4128.
- Lang, J. P., E. E. Bala Rajagopalan, D. Brungard, S. Dharanikota, G. Li, E. Mannie, D. Papadimitriou e Y. Rekhter (2002). Generalized MPLS Recovery Functional Specification. IETF Draft.
- Laprie, J.-C. (1992). *Dependability: Basic Concepts and Terminology*. Springer-Verlag, Viena, Austria.
- Law, A. M. e W. D. Kelton (1991). *Simulation Modeling & Analysis*. McGraw-Hill, Inc, U.S.A., 2nd ed.
- Li, T. e B. Wang (2005). On Optimal Survivability Design in WDM Optical Networks Under a Scheduled Traffic Model. Em *Proceedings of Design of Reliable Communication Networks (DRCN 05)*.
- Liu, C. e L. Ruan (2005). p -Cycle Design in Survivable WDM Networks with Shared Risk Link Groups (SRLGs). Em *Proceedings of Design of Reliable Communication Networks (DRCN 05)*.
- Liu, Y., D. Tipper e P. Siripongwutikorn (2001). Approximating optimal spare capacity allocation by successive survivable routing. Em *Proceedings of the IEEE Conference on Computer Communications (IEEE INFOCOM 01)*, págs. 699–708.
- Losego, F., M. Tornatore, G. Maiery e A. Pattavina (2005). Time Constraints in an OTN Semi-Automatic Control System. Em *Proceedings of Design of Reliable Communication Networks (DRCN 05)*.
- Madeira, H. C. (1993). *Detecção de Erros com base no Comportamento*. Tese de Doutoramento, Coimbra.
- Martins, E., M. Pascoal e J. Santos (1999). An algorithm for ranking loopless paths. Rel. Téc. 99/007, CISUC.
URL <http://www.mat.uc.pt/~marta/Publicacoes/mps2.ps>
- Marzo, J. L., E. Calle, C. Scoglio e T. Anjali (2003). QoS Online Routing and MPLS Multilevel Protection: A Survey. *IEEE Communications Magazine*, 41(10):126–132.
- Matsumoto, M. e T. Nishimura (1998). Mersenne twister: A 623-dimensionally equidistributed uniform pseudorandom number generator. *ACM Trans. on Modeling and Computer Simulation*, 8(1):3–30.

- Mellah, H. e A. F. Mohamed (2003). Local path protection/restoration in MPLS-based networks. Em *The 9th Asia-Pacific Conference on Communications (APCC 03)*, volume 2, págs. 620–622.
- Mendiratta, V. B. e C. Witschorik (2003). Introduction to Telecommunications Systems Survivability – A Practitioner’s View. apresentação convidada em "Design of Reliable Communication Networks (DRCN) 2003, Banff, Alberta, Canada, October 19-22, 2003".
- Menth, M. e R. Martin (2005). Network Resilience through Multi-Topology Routing. Em *Proceedings of Design of Reliable Communication Networks (DRCN 05)*.
- Menth, M., J. Milbrandt e A. Reifert (2004). End-to-End Protection Switching Mechanisms for MPLS Networks. Rel. Téc. 320, University of Wurzburg, Institute of Computer Science.
- Minei, I. (2004). MPLS DiffServ-aware Traffic Engineering. Rel. Téc. 200048-001, Juniper Networks, Inc., Sunnyvale, USA.
- Minei, I. e J. Lucek (2005). *MPLS-Enabled Applications*. John Wiley & Sons, Ltd.
- Osborne, E. e A. Simha (2003). *Engenharia de tráfego com MPLS*. Editora Campus - Elsevier Science.
- Pan, P., G. Swallow e A. A. (Eds.) (2005). Fast reroute extensions to RSVP-TE for LSP tunnels. IETF RFC 4090.
- Park, P.-K., H.-S. Yoon, S. C. Kim, J. Park e S. Yang (2004). Design of a dynamic path protection mechanism in MPLS networks. Em *The 6th International Conference on Advanced Communication Technology (ICACT 04)*, volume 2, págs. 857–861.
- Pasqualini, S., A. Iselt, A. Kirstädter e A. Frot (2004). MPLS Protection Switching vs. OSPF Rerouting: A Simulative Comparison. Em *Proceedings of the International Workshop on Quality of future internet Services (QofIS 04)*, págs. 174–183.
- Pinto, M. J. M. R. C. N. (2005). *Encaminhamento Diferenciado para Comunicações em Grupo com Requisitos de Qualidade de Serviço*. Tese de Doutorado, Escola de Engenharia, Departamento de Informática, Universidade do Minho.
- Prinz, R. G., A. Autenrieth e D. A. Schupke (2005). Dual Failure Protection in Multilayer Networks based on Overlay or Augmented Model. Em *Proceedings of Design of Reliable Communication Networks (DRCN 05)*.
- Qiao, C. e D. Xu (2002). Distributed Partial Information Management (DPIM) Schemes for Survivable Networks - Part I. Em *Proceedings of the IEEE Conference on Computer Communications (IEEE INFOCOM 02)*, págs. 302–311.
- Raza, S., F. Aslam e Z. Uzmi (2005). Online routing of bandwidth guaranteed paths with local restoration using optimized aggregate usage information. Em

- Proceedings of the IEEE International Conference on Communications (ICC 05)*, volume 1, págs. 201–207.
- Ricciato, F., M. Listasti, A. Belmonte e D. Perla (2003). Performance evaluation of a distributed scheme for protection against single and double faults for MPLS. Em M. A. M. et al. (ed.), *Proceedings of the International Workshop on QoS in Multiservice IP Networks (QoS-IP 2003)*, nº 2601 em Lecture Notes on Computer Science, págs. 218–232. Springer Verlag, Berlin.
- Rosen, E., A. Viswanathan e R. Callon (2001). Multiprotocol label switching architecture. IETF RFC 3031.
- Schwabe, T. e C. Gruber (2005). Traffic Variations Caused by Inter-domain Rerouting. Em *Proceedings of Design of Reliable Communication Networks (DRCN 05)*.
- Sharma, V., F. Hellstrand, B. Mack-Crane, S. Makam, K. Owens, C. Huang, J. Weil, B. Cain, L. Anderson, B. Jamoussi, A. Chiu e S. Civanlar (2003). Framework for multi-protocol label switching (MPLS)-based recovery. IETF RFC 3469.
- Smit, H. e T. Li (2004). Intermediate System to Intermediate System (IS-IS) Extensions for Traffic Engineering (TE). IETF RFC 3784.
- Soares, C. G. e E. Zio (eds.) (2006). *Safety and Reliability for Managing Risk (Proceedings of ESREL 2006)*, volume 3. Unit of Marine Technology and Engineering of Instituto Superior Técnico, Engineering Faculty of the Technical University of Lisbon and European Safety and Reliability Association (ESRA), Taylor & Francis/Balkema, Estoril, Portugal. ISBN 0-415-41620-5.
- Stamatelakis, D. e W. D. Grover (2000a). IP layer restoration and network planning based on virtual protection cycles. *IEEE Journal on Selected Areas in Communications*, 18(10):1938–1949.
- Stamatelakis, D. e W. D. Grover (2000b). Theoretical underpinnings for the efficiency of restorable networks using preconfigured cycles ("p-cycles"). *IEEE Transactions on Communications*, 48(8):1262–1265.
- Suri, S., M. Waldvogel e P. R. Warkhede (2001). Profile-Based Routing: A New Framework for MPLS Traffic Engineering. Em F. Boavida (ed.), *Proceedings of the 2rd International Workshop on Quality of Future Internet Services (QofIS 01)*, nº 2156 em Lecture Notes in Computer Science, págs. 138–157. Springer Verlag, Berlin.
- Suurballe, J. W. e R. E. Tarjan (1984). A quick method for finding shortest pairs of disjoint paths. *Networks*, 14(2):325–336.
- Tanenbaum, A. S. (2003). *Computer Networks*. Prentice Hall, fourth ed.
- Urra, A., E. Calle e J. Marzo (2005a). Enhanced protection using shared segment backups in a multiservice GMPLS-based networks. Em *Proceedings of the 10th*

- IEEE International Symposium on Computers and Communications (ISCC 05)*, págs. 752–757.
- Urrea, A., E. Calle e J. Marzo (2005b). Partial Disjoint Path for Multi-layer Protection in GMPLS Networks. Em *Proceedings of Design of Reliable Communication Networks (DRCN 05)*.
- Vasseur, J.-P., M. Pickavet e P. Demeester (2004). *NETWORK RECOVERY - Protection and restoration of optical, SONET-SDH, IP, and MPLS*. Morgan Kaufmann, Elsevier.
- Xiong, Y., D. Xu e C. Qiao (2003). Achieving Fast and Bandwidth-Efficient Shared-Path Protection. *JOURNAL OF LIGHTWAVE TECHNOLOGY*, 21(2):365–371.
- Xu, D., C. Qiao e Y. Xiong (2002). An Ultra-fast Shared Path Protection Scheme-Distributed Partial Information Management, Part II. Em *Proceedings of the 10th IEEE International Conference on Network Protocols (ICNP 02)*.
- Xu, D., Y. Chen, Y. Xiong, C. Qiao e X. He (2004). On finding disjoint paths in single and dual link cost networks. Em *Proceedings of the IEEE Conference on Computer Communications (IEEE INFOCOM 04)*, volume 1, págs. 705–715.
- Yetginer, E. e E. Karasan (2002). Robust path design algorithms for traffic engineering with restoration in MPLS networks. Em *Proceedings of the 7th IEEE International Symposium on Computer and Communications (ISCC 02)*, págs. 933–938.
- Yoon, S., H. Lee, D. Choi, Y. Kim, G. Lee e M. Lee (2001). An Efficient Recovery Mechanism for MPLS-Based Protection LSP. Em *Joint 4th IEEE International Conference on ATM (ICATM 01)*, págs. 75–79. Seoul, Korea.