



Universidade de Coimbra  
Faculdade de Ciências e Tecnologia  
Departamento de Engenharia Electrotécnica

# Um Estudo Simulacional de Redes Inter-centrais com Encaminhamento Dinâmico

Incluindo redes com integração de serviços

**Lúisa Maria Garcia Jorge**

COIMBRA  
2001



Luísa Maria Garcia Jorge

Um Estudo Simulacional de Redes  
Inter-centrais com Encaminhamento Dinâmico  
Incluindo redes com integração de serviços

*Dissertação submetida para satisfação parcial dos requisitos do  
programa de Mestrado em Sistemas e Automação (área de  
especialização de Análise e Planeamento de Sistemas)*

Universidade de Coimbra  
Faculdade de Ciências e Tecnologia  
Departamento de Engenharia Electrotécnica

COIMBRA  
2001



Esta dissertação foi realizada sob a orientação  
do Professor Doutor José Manuel Fernandes Craveirinha,  
Professor Catedrático do Departamento de Engenharia Electrotécnica  
da Faculdade de Ciências e Tecnologia da  
Universidade de Coimbra



# Agradecimentos

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

Em particular, gostaria de agradecer:

Ao meu orientador, Professor José Fernandes Craveirinha, por todo o apoio, disponibilidade e rigor científico com que apoiou o desenvolvimento deste trabalho.

Aos investigadores do INESC/Coimbra, nomeadamente a Doutora Teresa Gomes, pelo apoio prestado, em discussões e revisões frequentes durante a elaboração do trabalho. E ao Eng<sup>o</sup> Paulo Melo por toda a colaboração prestada, incluindo o desenvolvimento de ferramentas computacionais.

Aos elementos que integram o Laboratório de Simulação e Teletráfego do Departamento de Engenharia Electrotécnica, em particular à Eng<sup>a</sup> Lúcia Martins e ao Eng<sup>o</sup> Tiago Falcão e Sá pelo auxílio prestado.

À Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Bragança as facilidades concedidas para a elaboração desta dissertação.

Ao gabinete PRAXIS XXI da JNICT, que apoiou este trabalho com a bolsa PRAXIS XXI/BM/1495/94.

E ainda a todos que de alguma forma contribuíram para a elaboração deste trabalho.



# Sumário

A análise ao comportamento de redes inter-centrais, no respeitante ao grau de serviço, pressupõe o desenvolvimento de modelos analíticos ou simulacionais. O comportamento de tais redes é dependente do método de encaminhamento utilizado, sendo esta uma função indispensável nas redes de telecomunicações.

Nas últimas décadas tem sido propostos e/ou implementados um grande número de métodos de encaminhamento alternativos dinâmicos, entre outros casos devidos às necessidades de desempenho de redes cada vez maiores e mais complexas e à introdução de novos serviços. Neste trabalho é feita uma recolha e estudo de alguns desses métodos, que têm sofrido grandes alterações/evoluções desde o seu aparecimento oferecendo consideráveis benefícios no desempenho da rede a custos cada vez mais reduzido.

Alguns dos métodos de encaminhamento alternativos dinâmicos estudados são implementados para análise através de simulação. Para tal é construído um simulador do comportamento de redes de teletráfego, nomeadamente no respeitante à análise do grau de serviço, para redes de comutação de circuitos com a possibilidade de integração de diferentes classes de serviços. Este simulador é genérico, e independente do método de encaminhamento que se pretenda simular.

A implementação dos métodos é em geral baseada na descrição do comportamento dos métodos presentes na literatura, ocasionalmente suplementada de capacidades adicionais e parametrizações para permitir o estudo do desempenho dos métodos com alterações em relação à descrição original, tal como tornar um método, apresentado originalmente para encaminhar tráfego de apenas um tipo de tráfego, capaz de encaminhar simultaneamente várias classes de tráfego.

Finalmente é feito um estudo comparativo dos vários métodos de encaminhamento implementados, do ponto de vista do desempenho da rede em diferentes situações de carga. Os métodos são estudados no seu comportamento em diferentes redes (apenas um tipo de tráfego ou vários tipos de tráfego simultaneamente – rede com integração de serviços).

# Summary

The analysis of inter-switch networks, regarding Grade of Service, assumes the development of analytic and simulational models. The performance of these networks is dependent on the routing method used, and as such this is an indispensable function of modern telecommunication networks.

On the last previous decades a large number of dynamic alternative routing methods have been proposed and/or implemented. This has happened, among other causes, because of the always-bigger performance requirements on bigger and more complex networks, as well as the introduction of new services. This work includes a survey and study of some of those methods, that have been changing and evolving since their inception, offering considerable benefits on the network performance at ever-lower costs.

Some of the dynamic alternative routing methods studied are then implemented for simulational analysis. To do so, a real-time teletraffic simulator was built. This simulator focus on Grade of Service analysis on circuit switching networks, which may allow service integration. It's generic and routing methods independent simulation tool.

The methods are usually implemented according to the behaviour described in the literature, but sometimes, additional capabilities and parameterisations are considered and proposed. This is made to allow the study of the methods performance with some changes to their original formulation. This is also made to allow some methods, originally developed to route a single traffic class, to take into consideration several traffic classes on their routing algorithm.

Finally, a comparative study of the implemented routing methods is made, regarding the network performance when subject to different loads. The methods performances are analysed on different networks (both with a single traffic class, and with multiple simultaneous traffic classes – integrated services network).

# Índice

<b>Capítulo 1: Introdução</b> .....	<b>1</b>
1.1 Apresentação do trabalho.....	1
1.2 Objectivos propostos.....	2
1.3 Estrutura da dissertação .....	3
<b>Capítulo 2: Conceitos sobre Encaminhamento em Redes de Comutação por Circuitos</b> .....	<b>7</b>
2.1 Conceitos básicos.....	7
2.1.1 Seleccção dos caminhos alternativos .....	9
2.2 Definição de método de encaminhamento .....	11
2.2.1 Princípios de encaminhamento .....	13
2.2.1.1 Hierárquico .....	13
2.2.1.2 Fixo .....	13
2.2.1.3 Alternativo dinâmico .....	14
2.2.2 Comparação de alguns princípios de encaminhamento .....	16
2.2.3 Princípios de controlo de gestão do tráfego na rede (“network traffic management control”) .....	17
2.2.3.1 Controlos do Volume de Tráfego.....	19
2.2.3.2 Controlos do Encaminhamento.....	22
2.2.4 Princípios de controlo do transbordo .....	25
2.3 Mais algumas considerações sobre encaminhamento .....	27
2.3.1 Entradas e saídas múltiplas (“Multiple ingress/egress”) .....	27
2.3.2 Funcionamento conjunto de vários métodos de encaminhamento dinâmico .....	28
2.3.2.1 Redes com centrais que não suportam encaminhamento dinâmico .....	28
2.4 Parâmetros do desempenho da rede.....	29
<b>Capítulo 3: Resenha sobre os Métodos de Encaminhamento Dinâmico</b> .....	<b>31</b>
3.1 Introdução .....	31
3.2 Encaminhamento Alternativo Hierárquico .....	32
3.3 Método de encaminhamento sequencial alternativo fixo .....	36
3.3.1 Encaminhamento Alternativo Fixo (“Fixed Alternate Routing” - FAR) .....	36
3.4 Métodos de encaminhamento dependentes do tempo .....	37
3.4.1 Encaminhamento Dinâmico Não Hierárquico (“Dynamic Non-Hierarchical Routing” - DNHR) ..	37
3.5 Métodos de encaminhamento dependentes do estado.....	40
3.5.1 Métodos Baseados na Capacidade Residual (“Residual-Capacity Adaptative Routing” - RCAR) ..	40
3.5.1.1 Encaminhamento no caminho de menor carga (“Least Loaded Routing” - LLR) .....	41
3.5.1.2 Encaminhamento Alternativo no Agregado de Menor Carga (“Aggregated Least Busy Alternative Routing” - ALBA).....	41
3.5.1.3 Encaminhamento em Tempo Real (“Real Time Network Routing” - RTNR).....	45
3.5.1.4 Encaminhamento Controlado Dinamicamente (“Dynamically Controlled Routing” ou “Dynamic Call Routing” - DCR) .....	65
3.5.1.5 Sistema para Testar o Encaminhamento Adaptativo (“System to Test Adaptative Routing” - STAR).....	67
3.5.1.6 Gestão Dinâmica de tráfego (“Dynamic Traffic Management” – DTM).....	68
3.5.2 Métodos baseados nos processos de decisão de Markov .....	71
3.5.2.1 Encaminhamento Dependente do Estado Separável (“Separable State Dependent Routing” - SSDR) .....	71
3.5.2.2 Encaminhamento com previsão (“Forward Looking Routing” - FLR).....	75
3.5.2.3 Encaminhamento Adaptativo Dependente do Estado (“Adaptative State Dependent Routing” - ASDR) .....	78
3.6 Métodos de encaminhamento dependentes do acontecimento.....	80
3.6.1 Autómato com Aprendizagem (“Learning Automata” – LA).....	80
3.6.2 Encaminhamento Dinâmico Alternativo (“Dynamic Alternate Routing” - DAR).....	81
3.6.2.1 Extensões do DAR.....	82
3.6.3 Encaminhamento Aleatório com Aprendizagem (“Learning with Random Routing” - LRR).....	83
3.6.3.1 Caminho de sucesso para o topo (“Success To the Top” - STT) .....	84

3.6.4	Sequências Aleatórias Persistentes (“Sticky Random Sequences” – SRS).....	84
3.7	Métodos de encaminhamento híbridos/misto .....	85
3.7.1	Encaminhamento baseado no mapa de estados dos feixes (“Trunk Status Map Routing” - TSMR) .....	85
3.7.2	Encaminhamento Dependente do Tempo e do Estado (“State and Time-dependent Routing” - STR) .....	91
3.7.2.1	Extensões do STR.....	92
<b>Capítulo 4: Parametrização da Rede e Modelos de Tráfego Utilizados.....</b>		<b>95</b>
4.1	Representação do sistema .....	95
4.1.1	Tipos de tráfego exógeno.....	97
4.1.1.1	Fluxos de Tráfego Gerados por Processos de Poisson.....	98
4.1.1.2	Fluxos de Tráfego Gerados por Populações Finitas.....	99
4.1.2	Tráfego Multiclasse com Perda .....	100
4.1.3	Parâmetros do tráfego – notação.....	100
4.2	Especificação dos três casos particulares do sistema em estudo.....	103
4.2.1	Definição da Rede original .....	103
4.2.2	Sistema 1.....	104
4.2.3	Sistema 2.....	105
4.2.3.1	Redimensionamento grosseiro da rede original .....	105
4.2.3.2	Resultados obtidos para o redimensionamento .....	108
4.2.4	Sistema 3.....	111
4.2.4.1	Novo redimensionamento grosseiro da rede original.....	113
4.2.4.2	Resultados obtidos para o novo redimensionamento .....	115
4.2.4.3	Cálculo do número de fontes e das intensidades de chamadas por fonte livre para os tipos de fluxos de tráfego $k$ gerados por populações finitas .....	119
4.2.4.4	Valores obtidos para os parâmetros $F^{(k)}(v)$ , $\gamma^{(k)}(v)$ e $\alpha_i$ .....	120
<b>Capítulo 5: Descrição e Implementação do Simulador .....</b>		<b>123</b>
5.1	Introdução.....	123
5.2	Conceitos introdutórios sobre simulação .....	123
5.2.1	Simulação por acontecimentos discretos .....	125
5.2.2	Outros tipos de simulação.....	126
5.2.3	Tipos de software de simulação para simular redes de comunicação.....	126
5.3	Breve descrição da linguagem de simulação utilizada - OMNeT++ .....	127
5.3.1	Abordagem geral ao OMNeT++.....	127
5.3.2	Linguagem utilizada para definir o modelo - Linguagem NED.....	128
5.3.3	Desenvolvimento dos módulos simples – Programação dos algoritmos. ....	130
5.3.4	Geração de números aleatórios.....	130
5.3.5	Compilar e executar uma simulação .....	131
5.3.5.1	Ficheiro de configuração .....	131
5.3.6	Interfaces com o utilizador .....	132
5.4	Análise do sistema a implementar .....	133
5.4.1	Descrição do sistema .....	134
5.5	Descrição do modelo de simulação implementado.....	135
5.5.1	Representação do modelo na linguagem NED .....	136
5.5.2	Módulos simples implementados.....	136
5.5.2.1	Processo Central .....	143
5.5.2.2	Processo Gerador .....	153
5.5.3	Componentes gerais para todos os encaminhamentos .....	158
5.5.3.1	Componentes base .....	160
5.5.3.2	Outros componentes .....	162
5.5.4	Componentes específicos de cada encaminhamento.....	164
5.5.4.1	DAR.....	165
5.5.4.2	DCR.....	166
5.5.4.3	RTNR.....	167
5.5.5	Estrutura global da implementação (Diagrama de classes).....	167
5.6	Execução de uma simulação .....	169
5.6.1	Entradas para uma simulação .....	169
5.6.1.1	Configuração da rede (ficheiro de topologia) .....	169

5.6.1.2	Ficheiro de configuração (ficheiro <i>ini</i> ) .....	170
5.6.2	Os vários passos para executar uma simulação.....	170
5.6.3	Apresentação dos resultados .....	176
5.6.4	Intervalos de confiança das estimativas .....	177
<b>Capítulo 6:</b>	<b>Resultados da Simulação.....</b>	<b>181</b>
6.1	Introdução .....	181
6.2	Detalhes do estudo e características comuns a todas as simulações .....	182
6.2.1	Alguns valores comuns a todas as simulações .....	183
6.3	Experiência computacional com o Sistema 1.....	186
6.3.1	Análise da probabilidade de bloqueio global .....	186
6.3.2	Análise da probabilidade de bloqueio ponto a ponto .....	192
6.3.3	Análise da probabilidade de bloqueio nos feixes .....	193
6.3.4	Análise ao comportamento da simulação.....	194
6.4	Experiência computacional com o Sistema 2.....	195
6.4.1	Análise da probabilidade de bloqueio global na rede .....	196
6.4.2	Estudo da variação dos valores dos parâmetros específicos do RTNR, no RTNR .....	202
6.4.3	Análise da probabilidade de bloqueio global por classe .....	204
6.4.4	Análise da probabilidade de bloqueio ponto a ponto por classe .....	208
6.5	Experiência computacional com o Sistema 3.....	209
6.5.1	Análise da probabilidade de bloqueio global na rede .....	210
6.5.2	Estudo da variação do valor do parâmetro Factor1 no RTNR.....	216
6.5.3	Análise da probabilidade de bloqueio global por classe .....	217
6.5.4	Análise da probabilidade de bloqueio ponto a ponto por classe .....	224
6.6	Resultados mais relevantes .....	227
<b>Capítulo 7:</b>	<b>Conclusões e Trabalho Futuro .....</b>	<b>231</b>
7.1	Trabalho realizado .....	231
7.1.1	Resultados fundamentais do estudo simulacional .....	232
7.2	Contribuições desta dissertação .....	233
7.3	Direcções posteriores de pesquisa.....	234
<b>Apêndice A:</b>	<b>Descrição do Modelo usando a Linguagem NED.....</b>	<b>235</b>
<b>Apêndice B:</b>	<b>Programação de Módulos Simples – Mecanismos de Suporte.....</b>	<b>239</b>
B.1	Acesso aos parâmetros e portas dos módulos .....	239
B.2	Envio e recepção de mensagens.....	240
B.2.1	Associar parâmetros e objectos às mensagens .....	240
B.2.2	Duplicação de mensagens .....	241
B.2.3	Enviar mensagens .....	241
B.2.4	Atrasar mensagens .....	241
B.2.5	Receber mensagens.....	241
B.2.6	Escalonamento de mensagens.....	242
B.3	Geração de números aleatórios .....	242
B.3.1	Usar os geradores de números aleatórios directamente.....	242
B.3.2	Geração de variáveis aleatóreas .....	243
B.3.3	Números aleatórios obtidos a partir de histogramas .....	243
B.4	Parâmetros – classe <i>cPar</i> .....	243
B.5	Funções estatísticas.....	244
B.5.1	A classe <i>cStatistic</i> e as suas derivadas .....	244
B.5.2	Detecção do fim do período de transição e validação dos resultados .....	245
B.6	Terminar uma simulação.....	245
B.7	Registo de resultados .....	245
B.7.1	Vectores de saída– A classe <i>cOutVector</i> .....	245
B.7.2	Saída de escalares .....	246
B.7.3	Saída dos objectos estatísticos .....	246
B.8	Outros .....	246
B.8.1	Contentores .....	246
B.8.1.1	Fila de espera – classe <i>cQueue</i> .....	246
B.8.1.2	Array dinâmico – classe <i>cArray</i> .....	247

B.8.1.3	Contentores de items que não são objectos – classe cLinkedList e cBag .....	247
B.8.2	Nome dos objectos na rede e identificadores numéricos na rede.....	247
B.8.3	Acedendo a outros módulos na rede .....	247
B.8.4	Mostrar informação acerca da actividade dos módulos .....	248
B.8.5	Conselhos para acelerar uma simulaçãoC.....	248
<b>Apêndice C: Entradas para a Simulação.....</b>		<b>249</b>
C.1	Ficheiro da topologia (ficheiro txt).....	249
C.1.1	Descrição da Estrutura do ficheiro de topologia.....	249
C.1.2	Exemplo de um ficheiro de topologia.....	251
C.2	Ficheiros de configuração (ficheiro ini).....	252
C.2.1	Descrição da estrutura do ficheiro criado para gerar o ficheiro de configuração automaticamente (ficheiro <i>cfg</i> ) .....	253
C.2.2	Exemplo de um ficheiro criado para gerar o ficheiro de configuração automaticamente .....	253
C.2.3	Descrição da estrutura do ficheiro de configuração (ficheiro <i>ini</i> ).....	254
C.2.4	Exemplo de um ficheiro de configuração .....	255
C.3	Outros ficheiros de topologia utilizados na simulação .....	259
C.3.1	Ficheiro que representa o Sistema 1 .....	259
C.3.2	Ficheiro que representa o Sistema 2 .....	259
C.3.3	Alterações feitas no ficheiro de topologia que representa o sistema 3.....	260
<b>Apêndice D: Estudo Comparativo da Congestão através de Método Analítico e Simulação.....</b>		<b>263</b>
D.1	Cálculo da Congestão no Tempo através da fórmula de Engset .....	263
D.2	Resultados analíticos .....	264
D.3	Resultados da simulação .....	265
D.4	Conclusões.....	265
<b>Apêndice E: Estrutura do Programa de Simulação Implementado.....</b>		<b>267</b>
E.1	Componentes gerais para todos os encaminhamentos .....	267
E.1.1	Componentes base .....	270
E.1.1.1	Controlador .....	270
E.1.1.2	Encaminhador .....	270
E.1.1.3	Estatísticas .....	272
E.1.2	Outros componentes .....	272
E.1.2.1	Caminhos .....	273
E.1.2.2	Armazém de dados.....	273
E.1.2.3	Tráfego.....	275
E.1.2.4	Reserva de números aleatórios.....	275
E.2	Componentes específicos de cada encaminhamento .....	276
E.2.1	Estatísticas globais.....	276
E.2.2	DIRECTO .....	279
E.2.3	FAR .....	280
E.2.3.1	Carrega tabelas de encaminhamento.....	281
E.2.4	DAR.....	282
E.2.5	DCR.....	285
E.2.6	RTNR.....	288
E.2.6.1	Bits.....	296
<b>Apêndice F: Saídas da Simulação .....</b>		<b>299</b>
F.1	Ficheiro de Escalares .....	299
F.1.1	Estrutura do ficheiro de saída de escalares .....	299
F.1.2	Exemplo de um ficheiro de saída de escalares.....	302
F.2	Ficheiro de tabelas (ficheiro txt).....	307
F.2.1	Estrutura do ficheiro de saída contendo tabelas .....	307
F.2.2	Exemplo de um ficheiro de saída contendo tabelas .....	308
<b>Apêndice G: Estudo Estatístico da Confiança associada aos Resultados Simulacionais.....</b>		<b>311</b>
G.1	Intervalos de confiança da experiência com o Sistema 1 .....	312
G.2	Intervalos de confiança da experiência com o Sistema 2.....	319
G.3	Intervalos de confiança da experiência com o Sistema 3.....	332

<b>Apêndice H: Outros Resultados da Simulação.....</b>	<b>355</b>
H.1    Experiência computacional com o Sistema 2.....	355
H.1.1    Análise da probabilidade de bloqueio nos feixes .....	355
H.2    Experiência computacional com o Sistema 3.....	357
H.2.1    Análise da probabilidade de bloqueio nos feixes .....	357
 <b>Bibliografia.....</b>	 <b>359</b>



# Lista de Acrónimos e Abreviaturas

## Geral

ARR –	Automatic Rerouting (reencaminhamento automático)
CCS –	CCS Common Channel Signalling (sinalização por canal comum)
DEN –	Distribuição Exponencial Negativa
DMSU –	Digital Main Switch Units
EHCS –	Economic Hundred Call Seconds
EDR –	Event Dependent Routing (encaminhamento dependente do acontecimento)
ESS –	Electronic Switching Systems
FES –	Future Event Set
GNED –	Graphical Network EDitor
GoS –	Grade Of Service (grau de serviço)
ITU –	International Telecommunications Union
ITU-T –	International Telecommunications Union - Telecommunications Sector
LCG –	Linear Congruential Generator (gerador de números aleatórios linear e congruencial)
MB –	Move to Back (move para o fim)
MF –	Move to Front (move para o início)
NED –	NEtwork Description
NEDC –	NEtwork Description Compiler
OD –	(par) Origem / Destino
OOC –	Originating Office Control (controlo na central origem)
PDF –	Probability Density Function (função densidade de probabilidade)
PSTN –	Public Switched Telephone Network (rede telefónica pública comutada)
RCAR –	Residual Capacity Adaptative Routing (encaminhamento baseado na capacidade residual)
RDIS –	Rede Digital com Integração de Serviços (ISDN – Integrated Services Digital Network)
SDOC –	Selective Dynamic Overload Control (controlo selectivo de sobrecarga dinâmico)
SDR –	State Dependent Routing (encaminhamento dependente do estado)
SOC –	Sequential Office Control (controlo nas centrais sequenciais)
SPC –	Stored Program Controlled (controlada por programa armazenado)
TDR –	Time Dependent Routing (encaminhamento dependente do tempo)

- TSM – Trunk Status Map (mapa de estados dos feixes)
- UML – Unified Modelling Language
- VT – Virtual Trunk (circuitos virtuais)

## **Métodos de encaminhamento**

- AAR – Automatic Alternative Routing (encaminhamento alternativo automático)
- ALBA – Aggregated Least Busy Alternative Routing (encaminhamento alternativo no agregado de menor carga)
- ASDR – Adaptative State-Dependent Routing (encaminhamento adaptativo dependente do estado)
- DAR – Dynamic Alternate Routing (encaminhamento dinâmico alternativo)
- DCR – Dynamically Controlled Routing ou Dynamic Call Routing (encaminhamento controlado dinamicamente)
- DNHR – Dynamic Non-Hierarchical Routing (encaminhamento dinâmico não hierárquico)
- DR-5 – Dynamic Routing – 5 minutes (encaminhamento dinâmico com actualizações a cada 5 minutos)
- DTM – Dynamic Traffic Management (gestão dinâmica de tráfego)
- FAR – Fixed Alternate Routing (encaminhamento alternativo fixo)
- FHR – Fixed Hierarchical Routing (encaminhamento hierárquico fixo)
- FLR – Forward-Looking Routing (encaminhamento com previsão)
- FTB – Failure To the Bottom (caminho ocupado para a base)
- HPR – High Performance Routing (encaminhamento de alto desempenho)
- LA – Learning Automata (autómato com aprendizagem)
- LBA – Least Busy Alternative (alternativa menos carregada)
- LLR – Least Loaded Routing (encaminhamento no de menor carga)
- LRR – Learning with Random Routing (encaminhamento aleatório com aprendizagem)
- RTNR – Real Time Network Routing (encaminhamento em tempo-real)
- SDR – State Dependent Routing (encaminhamento dependente do estado)
- SRS – Sticky Random Sequences (sequencias aleatórias persistentes)
- SSDR – Separable State-Dependent Routing (encaminhamento dependente do estado separável)
- STAR – System to Test Adaptative Routing (sistema para testar o encaminhamento adaptativo)
- STR – State- and Time-dependent Routing (encaminhamento dependente do tempo e do estado)
- STT – Success To the Top (caminho de sucesso para o topo)
- TSMR – Trunk Status Map Routing (encaminhamento baseado no mapa de estados dos feixes)

## **Companhias e Redes de telecomunicações**

AT&T – American Telephone and Telegraph Company

Bellcore – Bell Operations Company

BNR/TCN – Bell Northern Research/Telecom Canada Network

BTN – British Telecom Network

CNET – Centro Nacional de Estudos de Telecomunicações (France Telecom)

NT/CTN – Northern Telecom/Canadian Telephone Networks

NTN – Norwegian Telecom Networks

NTT – Nippon Telegraph and Telephone



# Capítulo 1

## Introdução

### 1.1 Apresentação do trabalho

A função de uma rede de telecomunicações é permitir o estabelecimento de caminhos de comunicação entre pares de utilizadores para lhes prestar determinados serviços dentro de certos critérios de qualidade de serviço, ao menor custo para o operador. Hoje em dia, estes serviços vão desde o tradicional serviço telefónico até à transmissão de imagens de elevada resolução. Como é impraticável ligar todos os utilizadores dois a dois, por equipamento dedicado com a disponibilidade de recursos necessária, torna-se inevitável a partilha de recursos. Há portanto que estabelecer um compromisso entre a economia de recursos e a disponibilidade do sistema para prestar os serviços. Os recursos necessários para manter um caminho de comunicação são ocupados quando ocorre o pedido de uma chamada e libertados quando termina a chamada. Como estes recursos são limitados o sistema pode tornar-se impossibilitado de atender um pedido, por falta de recursos disponíveis nesse instante. A rede pode no entanto conseguir encaminhar chamadas entre duas centrais mesmo que não existam recursos na ligação directa entre elas, mediante a utilização de encaminhamento alternativo.

Pretende-se que as situações em que um pedido é rejeitado sejam tão pouco frequentes quanto possível. Torna-se então necessário avaliar o Grau de Serviço (GoS – “Grade of Service”) quantitativamente. O objectivo será oferecer um serviço de qualidade aos utentes com um custo mínimo. Na fase de planeamento/dimensionamento das redes de telecomunicações inter-centrais, as etapas principais são a optimização topológica (o que envolve determinar a localização das centrais e as ligações entre elas), a optimização do encaminhamento do tráfego (o que envolve determinar como estabelecer as chamadas quando elas ocorrem, e que encaminhamento se deve utilizar para cada fluxo de tráfego) e a optimização dos componentes (o que envolve determinar a capacidade dos sistemas de transmissão e comutação de modo a garantir um determinado grau de serviço). No processo de dimensionamento da rede estas etapas

estão interrelacionadas e são normalmente realizadas iteractivamente até ser encontrada uma solução satisfatória. Na fase final de dimensionamento, a estrutura topológica da rede assim como o tipo de encaminhamento usado praticamente não sofrem alterações, pelo contrário a capacidade dos componentes sofre alterações frequentes.

A probabilidade de bloqueio é a medida mais importante de desempenho ou de Grau de Serviço em redes de comutação por circuitos. A probabilidade de bloqueio ou congestão de chamadas (valor médio da fracção de chamadas que são perdidas) é estimada pela razão entre o número de chamadas perdidas e o número de chamadas oferecidas. O bloqueio médio da rede dá-nos uma indicação geral do desempenho da rede. Esta medida não deve no entanto ser analisada por si só pois não é suficiente conseguir que o bloqueio médio da rede tenha um valor abaixo de um determinado limite (objectivo de Grau de Serviço – GoS). A rede deverá também garantir que o bloqueio médio ponto a ponto respeite um determinado nível de GoS sentido pelos utilizadores individuais. Também numa rede multi-serviço, é necessário garantir que um desempenho adequado em termos de bloqueio médio não oculte desempenhos inaceitáveis para determinados serviços menos frequentes.

## 1.2 Objectivos propostos

Com esta dissertação pretende-se fazer uma análise do grau de serviço em redes de telecomunicações inter-centrais, com estrutura topológica pré-determinada, em função do método de encaminhamento de tráfego utilizado. Abordar-se-á esta temática geral tomando como ponto de partida métodos de encaminhamento alternativo em redes multi-serviço de comutação de circuitos, especificamente métodos de encaminhamento dinâmico.

O trabalho será estruturado em quatro fases. A primeira fase passa pela recolha e estudo da literatura relevante correspondente aos métodos essenciais de encaminhamento alternativo dinâmico. Destes métodos será escolhido (de acordo com a relevância, evolução histórica, aplicação prática e principalmente capacidade de caracterizar as diferentes abordagens ao problema de encaminhamento) um subconjunto que será estudado em maior pormenor nas fases seguintes, considerando nomeadamente diferentes implementações dos métodos.

Na segunda fase, pretende-se desenvolver e implementar um simulador de teletráfego, do tipo simulador de tempo-real, que permita avaliar o desempenho de uma rede de estudo em termos de engenharia do teletráfego, nomeadamente no respeitante à análise do grau de serviço, para redes de comutação de circuitos com integração de diferentes classes de serviços. Este simulador deve ser genérico, e independente do método de encaminhamento que se pretenda simular (e capaz inclusivamente de simular redes heterogéneas, com uso de métodos de encaminhamento diferente em diferentes centrais).

Na terceira fase, e para este simulador, serão desenvolvidos módulos capazes de implementar os métodos de encaminhamento escolhidos anteriormente. Estes módulos serão desenvolvidos em geral baseados na descrição do comportamento dos métodos presentes na literatura, mas serão nalguns casos consideradas e propostas capacidades adicionais e parametrizações específicas para cada método, tendo em vista permitir o estudo do desempenho dos métodos com algumas alterações em relação à descrição original.

Finalmente, a quarta fase passa pelo estudo comparativo dos vários métodos implementados, do ponto de vista do desempenho da rede, em situações de carga nominal e em diversas situações de sobrecarga. Os métodos serão estudados no seu comportamento em diferentes redes, construídas através de dimensionamento aproximado a partir de uma rede muito utilizada na literatura [Mitra91a].

## 1.3 Estrutura da dissertação

No capítulo 2 desta dissertação serão revistos os conceitos chave de encaminhamento de tráfego, nomeadamente no respeitante à selecção de caminhos alternativos. Irá ser igualmente definido o conceito de método de encaminhamento. Estudaremos o que constitui um método de encaminhamento, focando os principais princípios de encaminhamento que se podem utilizar, como se efectua o controlo da gestão do tráfego na rede e o controlo do transbordo. De seguida, abordaremos também os casos de redes com entradas e saídas múltiplas e aquelas em que se usam em simultâneo mais do que um método de encaminhamento. Finalmente serão apresentados os diversos parâmetros do desempenho da rede a usar na nossa análise.

No respeitante ao capítulo 3, far-se-á uma resenha dos principais métodos de encaminhamento dinâmico que foram desenvolvidos e implementados em redes de alguns países, bem como de métodos propostos na literatura. Após uma muito breve revisão dos métodos clássicos de encaminhamento, incluindo o encaminhamento hierárquico e alternativo fixo, serão descritos os métodos de encaminhamento dinâmico abordados neste estudo. Podemos dividir os métodos de encaminhamento dinâmico em métodos dependentes do tempo, dependentes do estado, dependentes do acontecimento e híbridos/mistos, a que correspondem respectivamente os princípios de encaminhamento alternativo sequencial dependente explicitamente do tempo, adaptativo dependente do estado, adaptativo dependente do acontecimento e híbrido/misto, definidos no capítulo 2. Como exemplos de métodos inseridos na taxonomia anterior, serão apresentados entre outros os métodos DNHR (dependente do tempo [Ash90]), DCR [Girard90], RTNR [Ash91] e (S)SDR [Krishnan89] (dependentes do estado), DAR (dependente do acontecimento [Mitra91a]) e TSMR (híbrido [Girard90]).

Em relação ao capítulo 4 será apresentada uma representação da rede a analisar. O objectivo desta representação é obter as definições necessárias à análise em termos de engenharia de teletráfego de uma rede inter-centrais em que todas as chamadas exigem serviço em tempo real, é usada comutação por circuitos com estabelecimento da ligação feito a pedido e comunicação ponto a ponto, e em que vão ser considerados diferentes ritmos de transferência simultaneamente. Para tal, após o estabelecimento dos conceitos básicos (nomeadamente a caracterização dos tipos de fluxos de tráfego a tratar, incluindo o clássico tráfego de Poisson, bem como o tráfego gerado por populações finitas ou do tipo tráfego de Engset [Cooper81]) e da nomenclatura usada, serão descritos três casos particulares da rede. Estes casos particulares distinguem-se entre si pelos tipos de fluxo que lhe são oferecidos, pelo que se procede a um dimensionamento da rede de forma a tentar obter comportamentos similares. No primeiro sistema é oferecido à rede apenas uma classe de chamadas, trata-se de um sistema monoclasse com apenas chamadas monocanal (sistema 1). No segundo, os recursos da rede são partilhados por duas classes de chamadas, o tipo de tráfego é de Poisson para ambas as classes e as chamadas dos dois tipos de fluxos requerem um número de circuitos diferente para o seu transporte. Temos neste caso um sistema multiclasse com chamadas monocanal e multicanal (sistema 2). No terceiro sistema será considerada a integração de 4 tipos de fluxo de tráfego. O primeiro tipo de tráfego é de Poisson para duas classes de chamadas e para os dois restantes casos é de Engset generalizado. O requisito de recursos é dependente da classe de chamada. Temos também aqui um sistema multiclasse com chamadas monocanal e multicanal (sistema 3).

No capítulo 5, tendo em conta que as redes inter-centrais com encaminhamento dinâmico são em geral demasiado complexas para serem estudadas analiticamente através de um modelo rigoroso, nomeadamente em termos de desempenho dinâmico, recorreremos a um modelo simulacional para fazer o seu estudo. Neste capítulo far-se-á uma revisão dos princípios e conceitos de simulação que estão na base do desenvolvimento de uma plataforma computacional para a análise de desempenho de redes inter-centrais com comutação por circuitos, com métodos de encaminhamento dinâmico. Por outro lado, apresentam-se as características relevantes da linguagem de simulação utilizada, bem como uma descrição dos aspectos essenciais e módulos do pacote de software desenvolvidos neste âmbito. Assim, após a apresentação de alguns conceitos introdutórios sobre simulação computacional, será apresentada uma súmula da linguagem de simulação OMNeT++ [Varga00] (que pode ser complementada nos apêndices A e B). Com esta linguagem foi desenvolvido um sistema genérico de simulação de redes de comutação por circuitos, que é descrito neste capítulo (e também no apêndice E). Finalmente, será descrita a implementação de diversos métodos de encaminhamento dinâmico nesta plataforma simulacional, (foram implementados os métodos DIRECTO, FAR [Mitra91a], DAR, DCR e RTNR; destes o DIRECTO, o DAR e o RTNR foram implementados de modo a suportar o transporte de várias classes de serviço na rede simultaneamente). A descrição dos métodos é acompanhada, quando

necessário, das modificações ou generalizações implementadas em relação ao descrito no capítulo 3. Por fim, é apresentado o mecanismo de recolha de dados da simulação que foi implementado, bem como uma pequena descrição dos cálculos realizados para determinar os intervalos de confiança dos resultados das corridas do simulador.

No capítulo 6 são apresentados os resultados obtidos por simulação que parecem mais relevantes para os métodos de encaminhamento implementados. São analisados particularmente os métodos DAR e RTNR, servindo o método de encaminhamento DIRECTO como termo de comparação e indicador da dimensão das vantagens relativas conseguidas com cada um dos métodos de encaminhamento, em termos de desempenho da rede. Este capítulo, inicia-se com uma descrição das características comuns a todas as simulações, e está dividido em três partes principais, correspondendo cada uma a um caso particular do sistema em estudo, tal como definido no capítulo 4. Assim, inicialmente é feita a análise para o sistema 1 (apenas tráfego de Poisson, 1 classe), seguida da análise do sistema 2 (apenas tráfego de Poisson, 2 classes com necessidades distintas) e finalmente do sistema 3 (tráfego de Poisson e tráfego Engsetiano, 4 classes com necessidades distintas). Para cada um destes sistemas, é feita a análise da probabilidade de bloqueio global na rede e do bloqueio ponto a ponto. A análise destes valores por classe de tráfego é também efectuada para os sistemas multiclasse.

No capítulo final são recapitulados os resultados principais obtidos com o estudo, apontadas conclusões relevantes sobre a eficiência relativa dos diferentes métodos de encaminhamento dinâmico implementados, em diferentes cenários de carga na rede. Finalmente procurar-se-ão apresentar direcções possíveis de trabalho futuro neste domínio.



# Capítulo 2

## Conceitos sobre Encaminhamento em Redes de Comutação por Circuitos

Neste capítulo serão revistos os conceitos chave do encaminhamento de tráfego. Irá ser igualmente definido o conceito de método de encaminhamento e de acordo com este conceito classificaremos alguns métodos de encaminhamento. A descrição mais pormenorizada dos métodos de encaminhamento aqui referidos será o objecto do próximo capítulo.

### 2.1 Conceitos básicos

Para considerar o problema do encaminhamento do tráfego, vamos definir as características principais da estrutura da rede de comutação por circuitos.

Uma rede de comutação por circuitos inter-centrais pode ser descrita em vários planos, por exemplo o plano da rede de componentes (rede física), o plano da rede funcional (rede lógica), etc. O nosso estudo vai ser feito considerando apenas a rede funcional. A rede funcional considera-se descrita pelos nós (centrais de comutação), pelos ramos ou arcos (feixes) que os interligam, pelo tráfego oferecido central a central ou ponto a ponto (tráfego oferecido inter-centrais) e pelas correspondentes regras de encaminhamento. Cada feixe é constituído por um número de canais paralelos (circuitos).

Um fluxo de tráfego é o fluxo de chamadas geradas entre um determinado par origem/destino. Uma rede funcional à qual são oferecidos vários tipos de fluxos de tráfego, ou seja, à qual são oferecidas várias classes de chamadas, tendo cada uma destas classes as suas características de tráfego, permitirá representar em termos de engenharia de teletráfego, uma rede com integração de serviços.

Uma chamada com origem numa central **A** e destinada a uma central **B** necessita, para o seu sucesso, de um caminho de feixe simples ou múltiplo entre as centrais **A** e **B**, com pelo menos um número de

circuitos livres em cada feixe do caminho, na altura que a chamada é originada, igual ao número de circuitos necessários para essa chamada. Os circuitos são ocupados pela chamada no caminho seleccionado, durante a duração da chamada sucedida, no final da qual ficam disponíveis para estabelecer outras chamadas.

Um caminho de feixe simples é formado apenas pelo feixe que liga directamente a central origem à central destino, sendo por isso designado por caminho directo. O feixe do caminho directo é por vezes também designado por feixe directo. Um caminho de feixes múltiplos é um caminho que usa centrais de trânsito (centrais intermédias) logo é um caminho que liga a central origem à central destino usando dois ou mais feixes.

Em cada central estão armazenados vários conjuntos de caminhos que são usualmente designados por tabelas de encaminhamento. Cada um dos conjuntos de caminhos, armazenados numa dada central, contem os caminhos de possíveis escolha, ordenados ou não, entre esta central, onde a ligação é pedida (central origem), e uma determinada central final (central destino). Em cada central origem, existe uma tabela de encaminhamento por cada central que possa ser destino para chamadas originadas nessa central origem. Uma tabela de encaminhamento juntamente com as regras para seleccionar um caminho (do conjunto de caminhos nela armazenados) para um fluxo de tráfego, formam o que se chama por padrões de encaminhamento.

Tanto no planeamento como na análise de um determinado sistema de teletráfego são utilizadas matrizes de tráfego construídas a partir dos padrões de tráfego. Os padrões de tráfego são definidos em função das previsões do tráfego. As matrizes de tráfego representam de certa forma o tráfego a que se prevê que o sistema seja sujeito. Muitos dos conceitos definidos nesta secção serão definidos formalmente no capítulo 4.

Quando uma chamada tem a possibilidade de ser oferecida a vários caminhos (dois ou mais), de acordo com certas regras, chamados caminhos alternativos, esse modo de encaminhamento designa-se por encaminhamento alternativo.

Os caminhos alternativos devem normalmente ter no máximo 2 ou 3 feixes. Esta restrição deve ser introduzida porque é conhecido que os métodos de encaminhamento não hierárquicos que permitem caminhos com mais feixes podem ser altamente sensíveis nas sobrecargas, resultando na degradação do desempenho, por causa da utilização excessiva de recursos por chamada. Caminhos alternativos com mais de dois feixes podem no entanto ser eficazes quando a carga da rede for suficientemente pequena.

## 2.1.1 Selecção dos caminhos alternativos

Considere-se que existem  $n$  caminhos que podem ser tentados por uma chamada de uma origem particular para um destino particular. O conjunto destes caminhos é denotado por  $T$ . O sucesso ou falha da tentativa de uma chamada em usar algum desses caminhos é registado na central origem. O encaminhamento de uma chamada é controlado pela tabela de encaminhamento, a qual é alguma permutação  $x = (x(1), x(2), \dots, x(n))$  de  $T$ .

Existem dois modos de seleccionar o caminho alternativo, são eles o modo de selecção sequencial e o modo de selecção cíclica.

### 1. Selecção sequencial

O modo sequencial é um sistema de transbordo puro, começará sempre no mesmo caminho e tenta os caminhos sempre na mesma ordem. Considerando a tabela de encaminhamento  $x$ , o algoritmo de encaminhamento sequencial funciona como se segue. O caminho  $x(1)$  é tentado em primeiro lugar. Se algum caminho  $x(k)$ , ( $1 \leq k < n$ ), é tentado e não está livre, então o caminho  $x(k+1)$  é o próximo caminho tentado. Quando algum caminho é tentado com sucesso a chamada é estabelecida nesse caminho e não é tentado mais nenhum caminho por essa chamada. Se o caminho  $x(n)$  é tentado e está ocupado então a chamada é bloqueada e não é tentado mais nenhum caminho por essa chamada.

A principal vantagem desta forma de encaminhamento é a facilidade de implementação.

### 2. Selecção cíclica

Existem dois tipos de selecção cíclica; ambos tentam todos os caminhos do conjunto de caminho possíveis, por uma ordem específica, a diferença entre eles é o modo de escolher o caminho inicial. Num caso, se a chamada foi estabelecida num caminho, a próxima chamada começará por tentar o mesmo caminho. Por exemplo, se a chamada foi estabelecida no caminho 1, ver figura seguinte, a próxima chamada começará por tentar o caminho 1. Este método tira vantagem dos feixes que têm capacidade extra. No outro tipo de selecção cíclica, se uma chamada foi estabelecida num caminho, a próxima chamada começará por tentar o caminho seguinte àquele onde foi estabelecida a última chamada. Utilizando de novo a figura seguinte, se uma chamada foi estabelecida no caminho 1, a próxima chamada começará por tentar o caminho 2. A vantagem deste tipo de selecção é o de tentar distribuir as chamadas pelos diferentes caminhos.

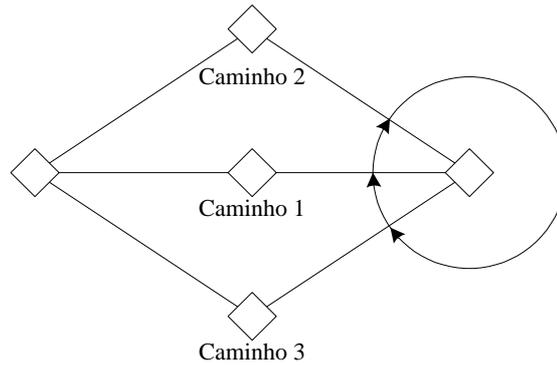


Fig. 2-1 - Selecção cíclica do caminho alternativo. Adaptada de [Ash98].

Considerando a tabela de encaminhamento como definida anteriormente. O caminho  $x(k)$ ,  $(1 \leq k \leq n)$ , está na posição  $k$  da tabela de encaminhamento. A posição 1 é considerada o início ou o topo da tabela de encaminhamento e a posição  $n$  o fim ou a cauda da tabela de encaminhamento.

Na selecção sequencial duas heurísticas possíveis de ordenação da tabela de encaminhamento são [Topkis86]:

1. Mover para o início (“Move to Front” - MF)

Um caminho que é tentado com sucesso passa a ser o primeiro caminho da tabela de encaminhamento e a ordem relativa dos outros caminhos mantém-se inalterada. Sendo assim a tabela de encaminhamento fica ordenada de acordo com o instante do sucesso mais recente para cada caminho.

2. Mover para o fim (“Move to Back” - MB)

Sempre que uma chamada falha num caminho esse caminho é passado para o final da tabela de encaminhamento, a ordem dos outros caminhos na tabela de encaminhamento é mantida inalterada. Sempre que um caminho é tentado com sucesso esse caminho fica no início da tabela de encaminhamento e os caminhos que estavam inicialmente no início passam a ser os sucessores do caminho que estava inicialmente no final da tabela de encaminhamento.

Suponha-se, por exemplo, que uma chamada falha no caminho  $x(1)$  e  $x(2)$  e que a terceira tentativa, no caminho  $x(3)$ , tem sucesso. De acordo com a heurística MF a tabela de encaminhamento será reordenada movendo o caminho com sucesso para o topo da tabela, resultando a nova tabela de encaminhamento  $x = (x(3), x(1), x(2), x(4), \dots, x(n))$ . Se em vez da heurística MF for usada a heurística MB a tabela de encaminhamento será alterada do seguinte modo: os caminhos onde ocorreu falha são movidos para a cauda da tabela de encaminhamento, resultando a nova tabela de encaminhamento  $x = (x(3), \dots, x(n), x(1), x(2))$ .

O MF desloca o caminho onde ocorreu sucesso para o início e desloca os caminhos onde ocorreu insucesso de uma posição, deixando os restantes caminhos nas posições iniciais, o MB também coloca o caminho com sucesso no início, contudo os caminhos onde ocorreu insucesso são colocados no final da tabela e os restantes caminhos são movidos, em direcção ao início da tabela, um número de posições igual ao número de caminhos onde ocorreu insucesso.

As duas heurísticas anteriores podem ser usadas em conjunção com outros processos adaptativos que periodicamente reconstruam o conjunto de caminhos da tabela de encaminhamento. Em [Garzia90] são simulados dois métodos de encaminhamento usando cada um uma das heurísticas anteriores designados por “Success To Top” – STT e por “Failure To Bottom” - FTB.

## 2.2 Definição de método de encaminhamento

As chamadas chegam aleatoriamente e são encaminhadas por caminhos adequados sempre que possível, ficando os circuitos ocupados neste caminho indisponíveis para outras chamadas durante a duração da chamada encaminhada. O tráfego produz assim variações aleatórias na ocupação dos feixes da rede. O objectivo de um método de encaminhamento é em geral tirar partido dessas variações aleatórias na forma da ocupação dos feixes, de modo a minimizar o bloqueio global na rede (a probabilidade de uma chamada que chega não encontrar um caminho adequado livre para a ligação) e portanto maximizar o tráfego total transportado.

Têm sido muitos os métodos de encaminhamento propostos. Na tabela seguinte pode ver-se o local onde foram implementados e/ou desenvolvidos alguns dos métodos de encaminhamento que vão ser estudados no próximo capítulo.

<b>Método</b>	<b>Implementação e/ou desenvolvimento</b>
<b>FHR</b>	Primeiro método de encaminhamento na rede AT&T em 1930, sofreu uma revisão em 1951
<b>DAR</b>	Implementado pela British Telecom Networks (BTN) na rede UK em 1996 Implementado pela Norwegian Telecom Networks (NTN) em 1996
<b>DCR</b>	Desenvolvido pela Bell Northern Research (BNR) Implementado na rede de longa distância Stentor Canada em 1991 Implementado na rede metropolitana Bell Canada Toronto em 1991 Implementado na rede Sprint USA em 1994 Implementado na rede MCI USA em 1995
<b>DNHR</b>	Implementado na rede AT&T de 1984 a 1991 Implementado na rede FTS-2000 em 1987
<b>DR-5</b>	Desenvolvido e testado pela Bellcore
<b>DTM</b>	Implementado pela Canadian Telephone Networks (CTN)
<b>RTNR</b>	Implementado na rede AT&T em 1991
<b>STAR</b>	Desenvolvido pela France Telecom (CNET) Testado na rede Paris
<b>STR</b>	Implementado pela Nippon Telephone and Telegraph no Japão (NTT) em 1992
<b>TSMR</b>	Desenvolvido nos AT&T Bell Laboratories

Tab. 2.1 – Local de desenvolvimento e implementação dos métodos de encaminhamento

De acordo com a nomenclatura usada pelo orientador (José Craveirinha) consideraremos que um método de encaminhamento é especificado por um princípio de encaminhamento, por um princípio de controlo de encaminhamento na rede (por exemplo centralizado ou distribuído), por um algoritmo de encaminhamento e pelo âmbito de aplicação (tipos de serviço/tráfegos). Uma técnica de encaminhamento será ainda especificada pelos requisitos de implementação em termos de :

- Sinalização e controlo (permitindo por exemplo *crankback*, etc.);
- Monitorização (inclui informação necessária em cada central);
- Funções de cálculo e optimização dinâmicas;
- Funções de previsão.

Os tipos de tráfego suportados podem ser: voz, dados ou serviços integrados. A sua aplicação pode ser no nível hierárquico (topo) de rede ou em toda a rede.

O princípio de encaminhamento pode ser hierárquico, fixo ou alternativo dinâmico. Por sua vez o princípio de encaminhamento fixo pode de ser trânsito fixo, alternativo fixo ou com partilha de carga fixo. E o princípio de encaminhamento alternativo dinâmico pode ser dependente do tempo, adaptativo – dependente do estado, adaptativo – dependente do acontecimento ou híbrido/misto. A descrição de cada um destes conceitos é feita na próxima secção e segue de perto a descrição apresentada em [ITU-T97a] e [Oda97].

Os algoritmos de encaminhamento especificam a forma de implementação dos princípios de encaminhamento.

## 2.2.1 Princípios de encaminhamento

### 2.2.1.1 Hierárquico

Neste princípio de encaminhamento as redes encontram-se estruturadas em diferentes níveis, normalmente usados para concentrar tráfego de umas regiões para as outras. Este princípio de encaminhamento foi bastante usado inicialmente devido às limitações tecnológicas das centrais. As redes são estruturadas em diferentes níveis organizacionais. As centrais de alto nível concentram sucessivamente o tráfego das áreas geográficas cada vez maiores, correspondentes a tráfego causado por centrais de níveis hierárquicos inferiores.

### 2.2.1.2 Fixo

O encaminhamento é fixo quando os padrões de encaminhamento são determinados na etapa de projecto e são mantidos inalterados no tempo.

#### 2.2.1.2.1 De trânsito fixo

Neste encaminhamento só há um caminho que pode ser usado entre cada par origem/destino. No entanto esse caminho pode usar centrais de trânsito. Uma rede que usava este princípio de encaminhamento era a rede hierárquica pura.

#### 2.2.1.2.2 Alternativo fixo

Neste encaminhamento existe um conjunto de caminhos alternativos entre cada par origem/destino e um modo de selecção do caminho alternativo (por exemplo modo de selecção sequencial) que são pré determinados e que são mantidos fixos no tempo.

#### 2.2.1.2.3 Partilha de carga fixo

O funcionamento do encaminhamento com partilha de carga fixo, na forma mais simples, é o seguinte, descrito em [Girard90]:

Para o tráfego entre um determinado par origem/destino, um dado conjunto de caminhos  $R_k$  é o único conjunto de caminhos disponíveis, para encaminhar as chamadas entre esse par origem/destino. O tráfego é dividido em  $|R_k|$  subtráfegos, sendo as chamadas do subtráfego  $l$  oferecidas ao caminho  $l$ . Se o caminho não estiver livre a chamada é perdida. O conjunto de proporções, ou seja os coeficientes de partilha, que indicam a proporção de chamadas

encaminhadas no caminho correspondente, são fixos. Por exemplo, um caminho pode ser atribuído, aleatoriamente, a uma chamada recém chegada, de acordo com a probabilidade de selecção desse caminho, que reflecte a proporção de tráfego que é oferecido a esse caminho. O conjunto de caminhos e as proporções são planeadas previamente.

O encaminhamento com partilha de carga pode ser combinado com o encaminhamento alternativo para construir um encaminhamento mais geral.

### **2.2.1.3 Alternativo dinâmico**

No princípio de encaminhamento dinâmico as decisões de encaminhamento são função explícita do tempo e/ou das condições em que a rede se encontra. O modo de encaminhar as chamadas é portanto alterado dinamicamente.

Neste encaminhamento tenta tirar-se vantagem da capacidade extra que existe na rede para lutar contra os vários tipos de sobrecargas e avarias.

Os princípios de encaminhamento dinâmico podem ser divididos em cinco categorias principais: dependentes do tempo, adaptativos - dependentes do estado, adaptativos - dependentes do acontecimento, híbridos/mistos e com partilha de carga.

#### **2.2.1.3.1 Alternativo sequencial dependente explicitamente do tempo**

No princípio de encaminhamento dependente do tempo, os padrões de encaminhamento são alterados em pontos fixos no tempo durante o dia tendo em vista otimizar a capacidade de resposta da rede face às mudanças previsíveis nos pedidos de tráfego. Os padrões de encaminhamento para cada um dos vários períodos de tempo são pré planeados.

Neste princípio de encaminhamento em particular (por exemplo utilizado no método de encaminhamento DNHR) a sequência de caminhos pré determinada é tentada sequencialmente sempre pela mesma ordem, sendo a sequência de caminhos a tentar trocada de tempos a tempos durante o dia de modo pré planeado, de acordo com os padrões de tráfego esperados. A figura 3-2 apresentada no capítulo 3, secção 3.4.1 mostra um exemplo de mudanças nos padrões de encaminhamento num método de encaminhamento alternativo sequencial dependente do tempo.

#### **2.2.1.3.2 Alternativo com partilha de carga dependente do tempo**

Sendo este princípio de encaminhamento também dependente do tempo, os padrões de encaminhamento são também alterados em pontos fixos no tempo durante o dia para adequar a capacidade da rede às mudanças previsíveis nos pedidos de tráfego. E os padrões de encaminhamento para cada um dos vários períodos de tempo são também pré planeados.

Este princípio de encaminhamento funciona como a partilha de carga fixa, mas as proporções a utilizar são substituídas de tempos a tempos, ao longo do dia, de acordo com os padrões de tráfego esperados. As proporções a usar são pré planeadas e mantêm-se fixas em cada período de tempo.

### **2.2.1.3.3 Adaptativo – dependente do estado**

Neste princípio de encaminhamento, os padrões de encaminhamento variam automaticamente de acordo com o estado da rede. Para cada método de encaminhamento que siga este princípio, existe um algoritmo pré planeado que vai ser utilizado durante um longo período de tempo, para determinar os padrões de encaminhamento em resposta às mudanças de estado da rede.

Neste princípio, a intenção é encaminhar as chamadas no “melhor caminho” disponível, que é determinado com base na informação de estado da rede. Podem ser usados algoritmos em que é calculada a capacidade residual dos caminhos, para todos os caminhos do conjunto de caminhos potenciais, e é seleccionado um caminho com base na sua capacidade residual (um exemplo pode ser encontrado no método LLR). Noutros algoritmos de encaminhamento dependentes do estado é calculado um custo para os caminhos, baseado na informação de estado (como no método SSDR). Nestes algoritmos o caminho seleccionado por uma chamada é aquele que tem custo mínimo no conjunto de caminhos potenciais. Nos métodos de encaminhamento que seguem este princípio é normal o caminho directo ser a primeira escolha e apenas no caso de não ser possível encaminhar a chamada nele é que vai ser determinado o melhor caminho alternativo. Se não for possível encontrar um caminho alternativo então a chamada é bloqueada.

Para possibilitar a alteração automática dos padrões de encaminhamento, respondendo a alterações do estado da rede, é necessário armazenar informação apropriada acerca do estado da rede, tal como registo das chamadas bem sucedidas e da ocupação dos grupos de circuitos. A selecção dos caminhos alternativos candidatos é feita em cada central ou num só processador servindo todas as centrais, dependendo do princípio de controlo do encaminhamento a usar. Esta reunião de informação do estado da rede e a selecção do caminho candidato pode ser efectuada periodicamente ou chamada após chamada.

### **2.2.1.3.4 Adaptativo – dependente do acontecimento**

Neste princípio de encaminhamento, usado por exemplo nos métodos DAR e STR, os padrões de encaminhamento são actualizados localmente baseados na informação obtida quando se tenta encaminhar uma chamada num dado caminho escolhido (sucesso ou insucesso). Nos métodos de encaminhamento que usam este princípio as chamadas quando chegam começam sempre por ser oferecidas ao caminho directo se este existir. Se não for possível o encaminhamento directo, a chamada é oferecida ao caminho alternativo corrente. Se a chamada não puder ser encaminhada também neste

caminho alternativo, é bloqueada. Se a chamada for bloqueada na escolha corrente é frequente tentar outro caminho alternativo de entre o conjunto de caminhos alternativos disponíveis.

#### **2.2.1.3.5 Híbrido/Misto**

Neste princípio de encaminhamento é usada uma mistura de dois ou mais princípios de encaminhamento. Um estudo de um método de encaminhamento que usa este princípio foi apresentado por Berezner em [Berezner97].

#### **2.2.1.3.6 Partilha de carga (“Load Share”)**

Na forma mais simples, funciona como o encaminhamento com partilha de carga fixo, descrito anteriormente, mas em que os coeficientes de partilha dependem de medidas do estado da rede, e são determinados periodicamente. Estudos acerca deste princípio de encaminhamento podem ser obtida em [Girard90] e [Bel85].

### **2.2.2 Comparação de alguns princípios de encaminhamento**

A tabela seguinte (tabela 2-2) compara alguns dos princípios de encaminhamento de tráfego relativamente à representação do tráfego, à informação armazenada nas tabelas de encaminhamento e às variações sofridas no encaminhamento do tráfego. Nesta tabela podem ver-se ainda alguns exemplos de métodos de encaminhamento implementados em redes reais.

Para cada princípio podem ser tomadas em conta várias matrizes de tráfego (cada uma das quais reflectirá uma combinação possível de variações de tráfego dos pares origem/destino) para testar o desempenho da rede em oposição às variações de tráfego. Para os princípios de encaminhamento fixos, quando se dimensiona a rede apenas se tem em conta uma matriz de tráfego, logo provavelmente o desempenho só será bom quando a rede estiver sujeita a tráfego com esse padrão, em todos os outros casos o desempenho será em geral inferior ao conseguido pelos outros princípios nos quais o encaminhamento é dinâmico, adaptando-se às características dinâmicas do tráfego. Para concluir os princípios de encaminhamento a seguinte característica deve ser salientada: enquanto que os métodos de encaminhamento que usem o princípio de encaminhamento fixo ou dependente do tempo trabalham com sequências ordenadas de caminhos, os métodos de encaminhamento que usem o princípio de encaminhamento dependente do estado ou dependente do acontecimento trabalham com um conjunto de potenciais caminhos sem nenhuma ordem pré determinada.

Princípio de encaminhamento	Representação do tráfego	Encaminhamento do tráfego	Variações no encaminhamento do tráfego	Métodos implementados em redes reais
Fixo	Várias matrizes de tráfego	Matriz de caminhos ordenados única	Nenhuma	Alternativo fixo em Redes Hierárquicas
Dependente do tempo		Várias matrizes de caminhos ordenados	Pré determinada	DNHR
Dependente do estado		Matriz única de caminhos disponíveis	Determinada em tempo real	DCR RTNR DR5
Dependente do acontecimento		Matriz única de caminhos disponíveis	Determinada em tempo real	DAR

Tab. 2-2 - Comparação dos princípios de encaminhamento do tráfego.

### 2.2.3 Princípios de controlo de gestão do tráfego na rede (“network traffic management control”)

As redes de telecomunicações são planeadas para satisfazer os pedidos de tráfego futuros. Apesar disso, nas operações em tempo real, as flutuações nos pedidos de tráfego podem resultar num desempenho da rede insatisfatório. Então, são necessários mecanismos de controlo de gestão da rede para manter um desempenho óptimo em tempo real.

Inicialmente as capacidades de gestão da rede eram limitadas, e na grande maioria eram controladas manualmente. Recentemente, as capacidades de gestão da rede tem melhorado significativamente, fruto da computadorização. As centrais controladas por programas armazenados e os sistemas de sinalização por canal comum oferecem uma transferência de informação eficiente e capacidades de gestão da rede.

Embora a rede esteja a crescer continuamente (aumento do número do feixes) de modo a satisfazer os pedidos de tráfego crescentes e variáveis, nas operações em tempo real o grau de serviço (“Grade Of Service” - GOS) em alguns pares de centrais O-D da rede podem ser piores do que o planeado devido a várias situações tais como: Os pedidos de tráfego serem maiores do que o previsto; Sobrecargas de tráfego devido a um qualquer acontecimento involuntário; Falhas inesperadas nos elementos da rede. Ao mesmo tempo que isso ocorre a maior parte da rede pode ter capacidade livre.

O objectivo dos mecanismos de controlo de gestão do tráfego na rede é fazer com que o máximo de chamadas sejam estabelecidas numa dada condição dos recursos.

Segundo Mase [Mase90] esses mecanismos podem ser divididos em duas categorias, controlos do tráfego da rede e controlos de configuração da rede (observar a figura 2.2). Vamos descrever apenas os controlos do tráfego da rede, pois são os únicos de interesse para este trabalho.

Os controlos do tráfego da rede, ainda segundo Mase [Mase90], podem ser divididos em controlos do volume do tráfego e controlos do encaminhamento, como pode ser observado na figura seguinte.

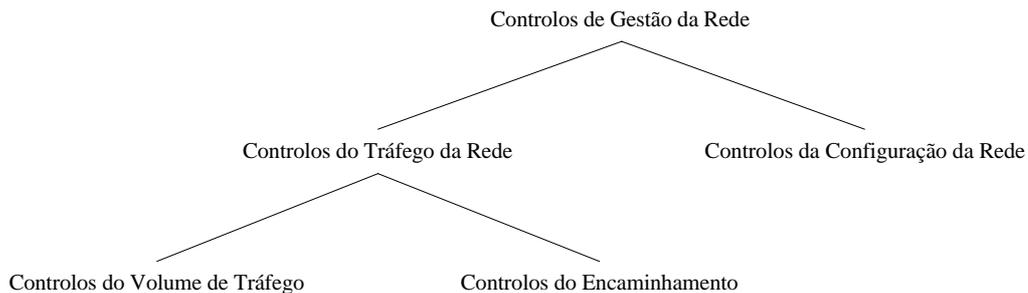


Fig. 2.2 – Controlo de Gestão da Rede. Adaptada de [Mase90].

Os controlos do volume de tráfego são também conhecidos por controlos protectores [Wolf90], [Chemouil90] ou restritivos [Régnier90] e os controlos do encaminhamento por controlos expansivos [Wolf90], [Chemouil90], [Régnier90].

Os controlos protectores são usados para proteger a integridade da rede quando ocorrem situações de tráfego excepcionais ou sobrecarga de elementos da rede. Os controlos expansivos suprimem ou modificam planos de encaminhamento implementados nas centrais para permitir o redireccionamento dos fluxos de tráfego. Em situações de sobrecarga, os controlos expansivos podem não funcionar do modo esperado, levando a rede a um estado de congestão (por exemplo devido a um excesso de encaminhamento alternativo). Por isto os dois tipos de controlo devem ser usados em conjunção. Em situações de sobrecarga os controlos protectores limitam o encaminhamento de algum tráfego. Com a evolução dos sistemas em direcção à gestão em tempo real deu-se a integração dos controlos do volume do tráfego nos métodos de encaminhamento.

Os controlos do tráfego da rede começaram por ser todos manuais sendo hoje a maior parte deles automáticos. Para suportar todas as situações que possam surgir de modo rápido e eficientemente é importante que todos os controlos sejam automatizados. Com a inovação nos métodos de encaminhamento o controlo passou a ser automatizado. Um exemplo destes é o método de encaminhamento DTM, descrito no próximo capítulo, que constrói o encaminhamento automaticamente de acordo com a carga corrente nas centrais e com os padrões de tráfego actuais e restringe o acesso à rede sempre que necessário. Neste método o controlo do tráfego é portanto automático.

Os métodos de encaminhamento adaptativo desempenham muitos dos controlos expansivos que começaram por ser activados manualmente. A introdução destes métodos de encaminhamento, que visam o uso eficiente dos recursos da rede, requer a adição de mecanismos protecção automáticos para evitar a degradação do desempenho da rede.

### **2.2.3.1 Controlos do Volume de Tráfego**

Dois exemplos de Controlos de Volume de Tráfego automáticos são o controlo selectivo de sobrecarga dinâmico (“Selective Dynamic Overload Control” – SDOC) e a reserva selectiva de circuitos (“Selective Trunk Reservation” - STR), ou simplesmente reserva de circuitos. Um outro exemplo de Controlo de Volume de Tráfego é o Código de Bloqueio, que foi inicialmente um controlo manual.

Quando na rede temos integração de serviços, é necessário além dos controlos protectores do tráfego de primeiro encaminhamento, também a existência de mecanismos de protecção dos serviços. Esses controlos protectores dos serviços devem permitir um uso desejado dos recursos pelos vários serviços. Conforme indicado na Recomendação E.529 [ITU-T97a], a Recomendação E.525 descreve os controlos de protecção de serviços disponíveis.

Em [Simões99] pode ser encontrada a descrição de alguns mecanismos de protecção dos serviços (ou estratégias de controlo de acesso). O método de encaminhamento RTNR, que vai ser descrito no capítulo seguinte, utiliza um controlo protector dos serviços designado também por reserva de circuitos, para conseguir uma utilização equilibrada da capacidade do caminho directo, pelas várias classes de tráfego.

#### **2.2.3.1.1 Controlo Selectivo de Sobrecarga Dinâmico**

A função do Controlo Selectivo de Sobrecarga Dinâmico – SDOC, descrito em [Wolf90], [Ash98], é aliviar a congestão das centrais. Quando uma central se torna congestionada envia um sinal específico para as centrais ligadas a ela. Por conseguinte essas centrais reduzem o tráfego para a central congestionada. As centrais que recebem a indicação de congestão activam um temporizador, o qual é actualizado se chegar um novo indicador de congestão. O controlo é removido quando expirar a sinalização de congestão pelo temporizador, passando a ser enviado tráfego normal de todas as centrais.

#### **2.2.3.1.2 Reserva de Circuitos (ou Junções)**

É conhecido que as redes não hierárquicas exibem dois estados de estabilidade, ou biestabilidade, sob congestão e que a rede pode transitar entre estes estados estáveis numa condição de congestão da rede. Este fenómeno de biestabilidade surge porque sob congestão, uma rede não é muitas vezes capaz de completar chamadas no ramo directo. Se o encaminhamento alternativo é permitido, por exemplo caminhos alternativos de dois feixes, então a chamada pode ser completada num caminho de dois feixes seleccionado de entre um largo número de escolhas possíveis de caminhos de dois feixes, apenas um dos quais necessita circuitos livres em ambos os feixes para ser usado para encaminhar a chamada. Porque

esta chamada de dois feixes ocupa recursos que poderiam talvez de outro modo ser usados para completar duas chamadas de um feixe, este é um uso pouco eficiente dos recursos da rede sob congestão. Na situação em que uma larga fracção de chamadas não podem ser completadas no ramo directo mas em vez disso ocupem caminhos de dois feixes, a capacidade de estabelecimento de chamadas é reduzida para metade porque muitas das chamadas ocupam o dobro dos recursos necessários. Este é um dos dois estados estáveis; isto é, muitas das chamadas ou quase todas usam dois feixes. O outro estado estável é aquele em que muitas chamadas ou quase todas ocupam apenas um feixe, o que é a condição desejada numa rede com grande conectividade lógica. Por esta razão, as redes com transbordos mútuos são inerentemente instáveis. Um modo de o evitar é usar a técnica com o nome de reserva de junções (*trunk reservation*). A reserva de junções é a técnica crucial usada nas redes não hierárquicas para prevenir a instabilidade.

A técnica de reserva de circuitos tem sido designada nos métodos de encaminhamento adaptativos baseados na capacidade residual por protecção de estado (*state protection*). Devido ao facto do termo reserva de circuitos poder gerar confusão com a reserva de circuitos físicos (que vai ser descrita mais à frente), o termo protecção de estado é por vezes utilizado em todo o contexto. Apesar disso aqui vamos utilizar o termo reserva de circuitos pois é o utilizado mais frequentemente.

A prática corrente nas redes telefónicas era estabelecer uma chamada sempre que existissem circuitos livres. Uma chamada era bloqueada apenas se todos os circuitos estivessem ocupados. Ao utilizar-se a técnica de reserva de circuitos as chamadas são consideradas de dois tipos, sendo algumas bloqueadas antes de todos os circuitos estarem ocupados. Especificamente, as chamadas que estão a ser oferecidas ao seu primeiro caminho são bloqueadas apenas na condição de todos os circuitos estarem ocupados, enquanto que chamadas de transbordo são bloqueadas sempre que o número de circuitos livres esteja abaixo de um certo valor. Por outras palavras, o tráfego directo num feixe é preferido ao tráfego encaminhado alternativamente, que ocupa dois ou mais feixes, estando este último sujeito a restrições de reserva de circuitos. Deste modo o tráfego encaminhado alternativamente é inibido de seleccionar um caminho de dois feixes quando não existir capacidade livre suficiente em ambos os feixes (quando não existir pelo menos capacidade em ambos os feixes igual à necessária para a chamada mais a reserva necessária para o encaminhamento directo), que será a condição mais provável em situações de congestão.

Esta técnica protege o tráfego de primeira escolha (tráfego que está a tentar o primeiro caminho, também designado por tráfego normal nesse caminho) de ser bloqueado por chamadas que transbordam de outros caminhos. A implementação consiste em reservar um certo número de canais em cada feixe para transportar apenas tráfego encaminhado directamente. Sendo  $N$  o número total de circuitos, é atribuído um limiar  $m < N$ , que pode ser distinto para cada feixe, tal que quando o número de circuitos ocupados no feixe exceder este valor, as chamadas que seriam encaminhadas alternativamente não serão aceites

neste feixe. O termo reserva de circuitos surgiu porque parte da capacidade  $N - m$  é reservada para o tráfego que está a tentar o seu primeiro caminho. Como já referido, esta reserva de circuitos não deve ser confundida com a reserva de circuitos físicos. Na *reserva de circuitos físicos* alguns circuitos só podem ser utilizados por determinadas chamadas, de acordo com a identidade dos circuitos (por exemplo, os primeiros  $k$  circuitos do feixe). Na *reserva de circuitos* agora referida, qualquer chamada pode utilizar qualquer circuito, a restrição é apenas na quantidade, e não na identidade, de circuitos que não podem ser ocupados por chamadas de transbordo. Este mecanismo de controlo tem pouco efeito em condições de carga leve e em condições de carga pesada considera o tráfego a encaminhar directamente como o único a encaminhar.

Se o número de canais reservados for escolhido cuidadosamente o problema da instabilidade desaparece e o desempenho de bloqueio é melhorado [Yum87].

Segundo Mitra [Mitra91a] as melhores escolhas para os parâmetros de reserva de circuitos são relativamente insensíveis à capacidade dos feixes e ao tráfego oferecido; e também, valores pequenos quase sempre resultam bem.

No mecanismo de reserva de circuitos descrito, a reserva é fixa, podendo-o ser para toda a rede ou por feixe. Alguns métodos de encaminhamento mais recentes utilizam já reserva de circuitos dinâmica, existindo diferenças de como e quando é que a reserva é aplicada. Por exemplo a reserva de circuitos pode existir sempre, ou ser accionada dinamicamente apenas sob situações de congestão da rede.

Utilizando reserva de circuitos durante todo o tempo pode conduzir a alguma perda na capacidade de estabelecimento de chamadas em condições normais ou situações de pequena congestão. Esta perda ocorre porque se os circuitos estão reservados para chamadas a ser encaminhadas no caminho directo, mas estas chamadas não surgem, então a capacidade está a ser desnecessariamente reservada pois poderia ser usada para estabelecer chamadas de tráfego encaminhado alternativamente que será deste modo bloqueado. No entanto, sob condições de congestão da rede, é necessário usar a reserva de circuitos para prevenir a instabilidade da rede, como explicado anteriormente. A reserva de circuitos accionada apenas sob situações de congestão da rede protege os recursos da rede em situações de congestão da rede e deixa os recursos disponíveis para serem partilhados quando as condições da rede o permitam.

O método de encaminhamento RTNR é um método de encaminhamento em que a reserva de circuitos é accionada dinamicamente, apenas é accionada a reserva de circuitos sob condições de congestão da rede.

### **2.2.3.1.3 Código de Bloqueio**

O controlo designado por Código de Bloqueio, descrito em [Mase90], proíbe o tráfego para um destino particular ou por um caminho particular. Esta acção deve ser tomada quando se conhece que uma parte

distante da rede está sob congestão. Este controlo é extremamente útil no caso de sobrecargas localizadas, especialmente se as chamadas puderem ser bloqueadas na origem ou próximo, preservando a capacidade da rede que de outro modo seria desperdiçada em tentativas ineficazes.

De modo a ser suportado o Código de Bloqueio Automático deve ser registada informação acerca do estado da rede. Os métodos de Código de Bloqueio Automáticos podem ser agrupados em três categorias de acordo com a informação que utilizam e do local onde são tomadas as decisões de bloqueio. Assim podemos ter:

Métodos Isolados - As decisões de Código de Bloqueio são tomadas em cada central baseadas na informação de estado registada por elas.

Métodos Distribuídos - As decisões de Código de Bloqueio são tomadas em cada central baseadas na informação de estado distribuída pelas outras centrais.

Métodos Centralizados - As decisões de Código de Bloqueio são tomadas num processador central baseadas na informação de estado registada numa base de dados centralizada.

### **Mecanismos de Controlo**

Existem duas classes de métodos para especificar a quantidade de tráfego a ser controlada:

- Controlo Proporcional - Uma certa proporção de tráfego oferecido a cada central é aceite.
- Controlo por limiar - Existe uma intensidade máxima até à qual o tráfego oferecido a cada central é aceite. Exemplos de controlos por limiar são *Call Gapping* e *Control Rate*. *Call Gapping*, descrito em [Ash98], é semelhante a uma porta que permite a passagem de chamadas até uma determinada intensidade; quando a intensidade máxima ocorre a porta é fechada durante um intervalo de tempo especificado. Durante esse intervalo de tempo as chamadas são bloqueadas.

### **2.2.3.2 Controlos do Encaminhamento**

Inicialmente, a gestão do tráfego da rede (controlos do volume de tráfego e controlos do encaminhamento) era activada em situações anormais, enquanto que em condições normais, era assumido que o projecto da rede conseguiria fazer frente às flutuações de tráfego. Os controlos do encaminhamento começaram por ser activados manualmente. Exemplo de controlos do encaminhamento manuais são: cancelar o encaminhamento alternativo, saltar caminhos e encaminhamento alternativo temporário (reencaminhar). Porém, presentemente a fronteira entre normal e anormal tende a desaparecer, e a gestão do tráfego da rede e os métodos de encaminhamento tornam-se mais intrincados. Hoje em dia com a evolução dos métodos de encaminhamento alternativo fixo para encaminhamentos dinâmicos, passaram a utilizar-se controlos de encaminhamento automáticos (os métodos de

encaminhamento dinâmico dependente do estado não são mais do que controlos de encaminhamento automáticos).

O encaminhamento dinâmico pode ser implementado num modo centralizado ou descentralizado. A informação de estado pode estar concentrada num processador central ou espalhada pelas centrais na rede. No modo centralizado é utilizado um processador que monitoriza o estado da rede e recomenda decisões. Este modo foi possível a partir do momento que apareceram os sistemas de comutação electrónicos e a uma grande capacidade de sinalização (canal de sinalização ou rede dedicada) entre as centrais. A troca de informação pode ser efectuada periodicamente ou quando solicitada. Desta forma, o horizonte das decisões de encaminhamento pode variar de vários minutos até segundos, ou mesmo até chamada após chamada. No princípio descentralizado a “inteligência” é distribuída ao longo da rede. O princípio de encaminhamento descentralizado pode ainda dividir-se em princípio de encaminhamento distribuído e princípio de encaminhamento isolado. O princípio isolado tem a vantagem de não precisar de uma rede de controlo, necessária nos princípios centralizado ou distribuído o que faz com que estes sejam mais vulneráveis a falhas.

O princípio de controlo do encaminhamento que pode ser utilizado por cada um dos métodos de encaminhamento dinâmico de tráfego pode ser um dos quatro seguintes: princípios de controlo isolados, princípios de controlo distribuídos, princípios de controlo centralizados e princípios de controlo híbridos/mistos.

Isolados - As decisões são tomadas em cada central baseadas na informação de estado registada por elas. Um programa é executado assincronamente na central origem, numa base chamada após chamada usando a informação disponível localmente, podendo existir uma modificação aleatória no encaminhamento logo que ocorra um bloqueio. Isto é, um novo caminho alternativo pode ser seleccionado na central origem, aleatoriamente, num conjunto de possíveis caminhos alternativos de dois feixes, quando ocorrer bloqueio.

Centralizados – As decisões são tomadas num processador central baseadas na informação registada numa base de dados centralizada. Os dados são transferidos, por uma rede de controlo, para um processador central que periodicamente executa um programa de encaminhamento, após o que envia as recomendações para as centrais para serem usadas durante o próximo período de tempo.

Distribuídos - As decisões são tomadas em cada central baseadas na informação de estado global distribuída pelas outras centrais. As actualizações do encaminhamento podem ser efectuada periodicamente ou numa base chamada após chamada.

Híbridos/Mistos – Quando as decisões de encaminhamento utilizam uma qualquer combinação dos controlos anteriores.

<b>Princípio de Controlo</b>	<b>Informação da rede</b>	<b>Decisão de encaminhamento</b>
<b>Isolado</b>	Apenas local	Central origem
<b>Distribuído</b>	Distribuída para todas as centrais	Central origem
<b>Centralizado</b>	Base de dados centralizada	Processador central

Tab. 2.3 – Classificação dos princípios de controlo do encaminhamento.

Na tabela anterior pode ver-se, para os métodos de encaminhamento dependentes do estado, os princípios de controlo do encaminhamento que podem ser utilizados. Pode ainda ver-se, por cada princípio de controlo, o local onde é tomada a decisão de encaminhamento e que informação do estado da rede é utilizada.

A tabela seguinte indica qual o princípio de controlo do encaminhamento utilizado por vários métodos de encaminhamento.

<b>Método de Encaminhamento</b>	<b>Isolado</b>	<b>Centralizado</b>	<b>Distribuídos</b>	<b>Híbridos</b>
<b>DAR</b>	X			
<b>DCR</b>		X		
<b>DR-5</b>		X	X	
<b>DTM</b>		X		
<b>RTNR</b>			X	
<b>SRS</b>	X			
<b>SSDR</b>		X	X	
<b>STAR</b>		X		
<b>STR</b>				X
<b>TSMR</b>		X		

Tab. 2.4 – Princípios de controlo do encaminhamento em alguns métodos de encaminhamento.

Os métodos de encaminhamento DCR, STAR e DR-5 utilizam o princípio de controlo do encaminhamento centralizado, logo a actualização do encaminhamento é feita periodicamente e as decisões de encaminhamento são tomadas globalmente. O método de encaminhamento DR-5 também pode utilizar o princípio de controlo do encaminhamento distribuído sendo neste caso tomadas localmente as decisões de encaminhamento. O método de encaminhamento DAR utiliza o princípio de controlo do encaminhamento isolado logo a actualização do encaminhamento é feita assincronamente e as decisões de encaminhamento são tomadas localmente.

## 2.2.4 Princípios de controlo do transbordo

Os métodos de controlo de transbordo especificam sob que condições uma chamada será perdida. Dependendo da central que controla o transbordo podemos ter um dos dois casos importantes seguintes, descritos em [Girard90]: controlo na central origem (“Originating Office Control” - OOC) ou controlo nas centrais sequenciais (“Sequential Office Control” - SOC). Com SOC, uma chamada que chega a uma central intermédia num caminho alternativo, e encontra todos os feixes de saída ocupados, é perdida. Sob OOC, um mecanismo de controlo examina cada caminho completamente da origem ao destino de modo a assegurar que todos os feixes estão livres antes de encaminhar a chamada através dele. Aqui o controlo é mantido pela central onde a chamada teve origem, daí o seu nome.

Pode-se dizer que SOC é uma forma de controlo míope, que só tem informação do estado dos feixes adjacentes à central, enquanto OOC usa informação do estado do caminho completo na sua decisão de encaminhamento.

Quando é utilizado SOC uma chamada avança uma central de cada vez. Cada central deve ter, para cada destino, um conjunto ordenado de centrais para poder determinar a próxima central para onde deve ser enviada cada chamada que lá chega. Isto é diferente do controlo OOC pois neste cada central tem um conjunto ordenado de caminhos para utilizar no encaminhamento das chamadas, o transbordo é feito de um caminho para outro. No controlo SOC a lista ordenada de centrais, que cada central possui para encaminhar para um dado destino é designado por bloco de encaminhamento. Neste controlo deve ser construído um bloco de encaminhamento para cada par de centrais. A figura seguinte (figura 2-3) apresenta um exemplo que ilustra o controlo SOC, nesse exemplo pretende-se encaminhar uma chamada de *A* para *Z*. Para encaminhar da central *A* para a central *Z* é consultado o bloco de encaminhamento *A-Z*, que se encontra na central *A*. Esse bloco indica que para uma chamada originada em *A* e destinada à central *Z* a primeira tentativa é ir directamente para a central *Z*. Se o feixe *A-Z* estiver ocupado a próxima tentativa é ir através da central *B*. Se o feixe *A-B* estiver ocupado a próxima tentativa é ir através da central *C*. Se o feixe *A-C* também estiver ocupado a chamada será bloqueada. Quando a chamada chega a uma central intermédia, o bloco de encaminhamento dessa central para a central destino é usado para determinar a próxima central. Por exemplo, se a chamada encontrou o feixe *A-Z* ocupado e portanto chegou à central *B* (assumindo que o feixe *A-B* estava livre), o bloco de encaminhamento *B-Z* é utilizado. Este bloco de encaminhamento diz que a chamada deve ir directamente para a central *Z*, se não for possível (caso em que o feixe *B-Z* esteja ocupado) então a central *C* deve ser tentada. Se o feixe *B-C* também estiver ocupado, a chamada é bloqueada.

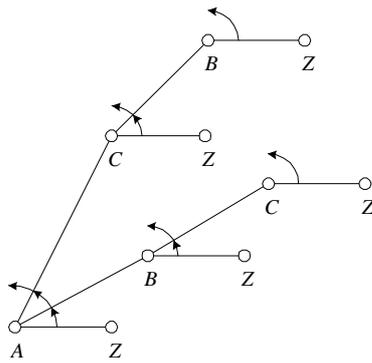


Fig. 2:3 – Encaminhamento de uma chamada de A para Z. Adaptada de [Ash98].

No método de encaminhamento hierárquico fixo é utilizado o controlo SOC.

Quando é utilizado SOC pode ocorrer que uma chamada depois de passar por vários feixes volte a uma central por onde já tinha passado ou pode ocorrer que uma chamada passe de uma central para outra e de novo desta para a anterior. Um modo de evitar estes fenómenos, que podem resultar na ocupação indefinida de circuitos sem que a chamada consiga ser estabelecida, é enviar a história completa (centrais já utilizadas pela chamada) de uma chamada à medida que se está a encaminhar e proibir que uma chamada seja encaminhada para uma central mais do que uma vez. Estes fenómenos não ocorriam no método de encaminhamento hierárquico devido à sua estrutura hierárquica.

Quando é permitido que o controlo seja passado para uma central intermédia seguinte, assumindo que o feixe de ligação a essa central intermédia está livre, esse controlo designa-se por *spill forward*. Se for utilizado OOC com *spill forward* numa determinada central e se o controlo fôr passado para essa central, a partir de então o transbordo apenas pode ocorrer para a frente dessa central e em caminhos que passem por essa central (isto é, em caminhos que tenham essa central como central de trânsito). Se um tal caminho não estiver disponível a chamada é perdida. Por outras palavras quando o controlo é transferido para uma central, não pode regressar à central de onde a transferência tinha sido feita.

No modo denominado por *call crankback*, ou *crankback* simplesmente, o controlo de transbordo, permite que uma chamada bloqueada num ponto intermédio de um caminho possa ainda ser oferecida a outros caminhos na sequência e não seja perdida imediatamente. Ou seja, é permitido que o movimento do transbordo ocorra na ordem inversa.

SOC, também chamado controlo progressivo ou controlo sequencial, não é mais do que OOC com *spill forward* em todas as centrais e sem *crankback*.

Alguns exemplos de métodos de encaminhamento que usam *crankback* são: DNHR, FAR, RTNR, SRS, TSMR e DAR-2. Exemplos de métodos de encaminhamento que não usam *crankback* são: DAR, STR e DCR.

Usualmente, o transbordo simples é adoptado quando se usa encaminhamento adaptativo. Isto elimina a necessidade de *crankback*. Além do mais, a necessidade de transbordos múltiplos não existe, dado que o encaminhamento é modificado consistentemente com as mudanças nas condições da rede.

## 2.3 Mais algumas considerações sobre encaminhamento

### 2.3.1 Entradas e saídas múltiplas (“Multiple ingress/egress”)

O trabalho com redes de comutação por circuitos com encaminhamento dinâmico tem estado concentrado principalmente em redes completamente ligadas. Além disto, cada vez mais, considerações de segurança estão a conduzir a arquitecturas com dois antecessores ou vários antecessores (arquitecturas *dual-* ou *multiparented*), nas quais uma chamada pode entrar na (ou sair da) rede principal em dois (ver figura seguinte) ou mais pontos de acesso.

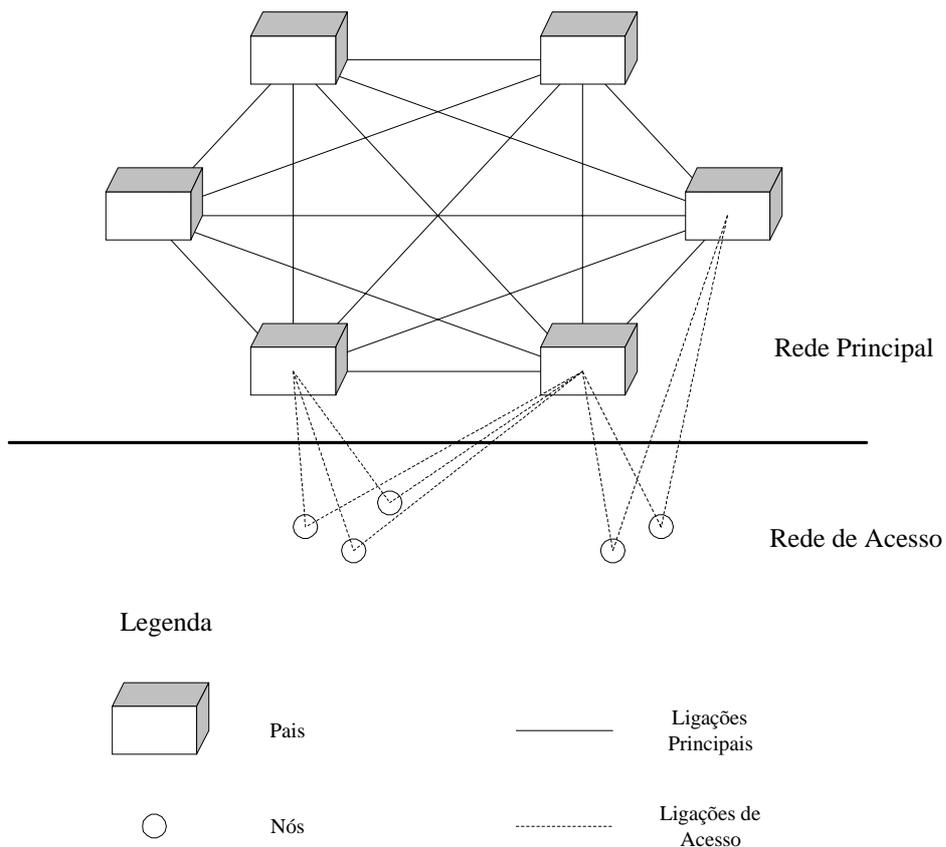


Fig. 2.4 – Estrutura de uma rede com dois antecessores. Adaptada de [Gibbens93].

As redes com um número de antecessores grande aumentam a segurança no acesso à rede principal e podem também reduzir o bloqueio através do aumento do número de caminhos disponíveis.

Por exemplo na rede UK as várias centrais principais, designadas por “Digital Main Switch Units” – DMSU, encontram-se completamente ligadas. Nessa rede existe uma camada inferior de centrais locais, sendo cada uma delas ligada a duas DMSU [Key88].

Gibbens em [Gibbens93] mostrou que métodos de encaminhamento dinâmicos simples com vários pontos de acesso são capazes de alcançar muitas das vantagens permitidas por métodos de encaminhamentos dinâmicos mais complexos com apenas um ponto de acesso.

## **2.3.2 Funcionamento conjunto de vários métodos de encaminhamento dinâmico**

A grande parte do trabalho desenvolvido tem sido feito considerando uma rede homogénea. Nessas redes é apenas usado um tipo de método de encaminhamento. Com o aumento das redes com encaminhamento dinâmico, tornou-se necessário considerar o funcionamento conjunto dos vários tipos de encaminhamento dinâmicos. Isto não só se torna necessário nas redes internacionais como nas redes nacionais, nestas últimas devido à abertura do mercado a novos operadores.

Uma outra razão para a necessidade de considerar o funcionamento conjunto de vários tipos de métodos de encaminhamento é a existência de algumas centrais, numa mesma rede, que não suportam o tipo de método de encaminhamento que se pretende usar nessa rede. Isto devido a limitações de algumas centrais.

Inicialmente, a nível internacional, era usado principalmente o encaminhamento hierárquico, principalmente através de acordos bilaterais. No entanto, ao longo do tempo, tal encaminhamento tem vindo a ser substituído por encaminhamento dinâmico ([Kashper95]), o que levou à necessidade de cooperação entre operadores de forma a poder estender estes métodos às ligações entre todas as redes. Neste sentido, tem sido desenvolvidos estudos (ver [Ash94], [ITU-T97b] e [Oda97]) com objectivo de padronizar as mensagens de sinalização entre as diversas centrais.

### **2.3.2.1 Redes com centrais que não suportam encaminhamento dinâmico**

Numa rede nem todas as centrais podem suportar encaminhamento dinâmico, no entanto pode continuar a ser possível utilizar um método dinâmico, ver por exemplo [Régnier95] onde foi demonstrado que nem todas as centrais precisam de suportar o encaminhamento dependente do estado para permitir a sua utilização. Nesse artigo foi mostrado que, no caso de se querer utilizar o DCR, as centrais que não o suportam podem utilizar uma variedade de métodos de encaminhamento que se adaptem às suas

capacidades. Podendo então ter o DCR a trabalhar em conjunto com outros métodos de encaminhamento para obter muitos dos benefícios que se conseguiam se fosse implementado o DCR ideal, que seria ter todas as centrais a permiti-lo.

## 2.4 Parâmetros do desempenho da rede

O objectivo deste estudo é a comparação do grau de eficácia de diferentes métodos de encaminhamento, dada uma estrutura específica da rede de telecomunicações inter-centrais especificada em termos da rede funcional associada. A análise do grau de eficácia, ou seja do grau de serviço (“Grade Of Service” - GOS), vai ser feita do ponto de vista do desempenho da rede. Numa rede de comutação por circuitos uma das medida de desempenho mais importantes é o cálculo dos bloqueios.

O bloqueio médio da rede dá apenas visão global do desempenho da rede. No entanto, esta medida não deve ser considerada por si só, pois o caso dessa medida ter um valor baixo apenas garante uma probabilidade de bloqueio baixo para os fluxos com maior intensidade de tráfego. É necessário analisar também os bloqueios médios ponto a ponto (onde ponto significa fonte de tráfego), pois só através destes é que podemos obter uma indicação das probabilidades de bloqueio sentidas pelos utilizadores que originam fluxos de tráfego de baixo caudal.



# Capítulo 3

## Resenha sobre os Métodos de Encaminhamento Dinâmico

### 3.1 Introdução

Far-se-á neste capítulo uma resenha dos principais métodos de encaminhamento dinâmico que foram desenvolvidos e implementados em redes de alguns países, bem como de métodos propostos na literatura.

Com as evoluções tecnológicas foram introduzidas centrais electrónicas que permitem o armazenamento de programas de controlo. Com essas centrais tornou-se possível considerar a implementação de algoritmos de encaminhamento de chamadas mais sofisticados. O encaminhamento dinâmico foi por essa razão proposto, a partir do início da década de 1980, como alternativa ao encaminhamento fixo, sendo mais eficiente que o encaminhamento hierárquico usado até então devido à maior capacidade de partilhar recursos disponíveis.

Nos métodos de encaminhamento dinâmico, as chamadas são encaminhadas no “melhor” caminho disponível na rede, de acordo com certos critérios que definem a essência de cada método, tentando utilizar de modo adequado toda a capacidade extra disponível nesta. Com as evoluções tecnológicas é possível implementar métodos de encaminhamento cada vez mais sofisticados, que implicam por vezes cálculos mais complexos. Devem por isto ser cuidadosamente avaliados os compromissos entre os ganhos de desempenho conseguidos e a complexidade dos cálculos a efectuar embora o desenvolvimento de centrais cada vez mais poderosas computacionalmente diminuam de alguma forma estas preocupações.

A necessidade de um elevado GOS originado pela introdução de novos serviços força a que novos métodos de encaminhamento sejam projectados de forma a melhorar a eficiência e robustez da rede

numa grande variedade de situações. O encaminhamento dinâmico tem sofrido grandes alterações desde o seu aparecimento.

O encaminhamento dinâmico pode ser implementado num modo centralizado ou descentralizado. No modo centralizado é utilizado um processador, que monitoriza o estado da rede e recomenda decisões de encaminhamento. Este modo foi possível a partir do momento que apareceram os sistemas de comutação electrónicos e uma grande capacidade de sinalização (canal de sinalização ou rede dedicada) entre as centrais. O horizonte das decisões de encaminhamento pode variar de vários minutos até poucos segundos, ou mesmo até chamada após chamada. Por outro lado, no modo descentralizado a “inteligência” (capacidade de decisão) é distribuída ao longo da rede. Este modo pode ser dividido em isolado ou distribuído, como definido no capítulo 2. O modo isolado tem a vantagem de não precisar de uma rede de controlo, necessária no modo centralizado ou distribuído, o que faz com que estes sejam mais vulneráveis a falhas.

Este capítulo será dedicado, quase exclusivamente, à descrição de vários métodos de encaminhamento dinâmicos encontrados na literatura. Os métodos de encaminhamento dinâmico estão divididos em métodos dependentes do tempo, dependentes do estado, dependentes do acontecimento e híbridos/mistos. A estes correspondem respectivamente os princípios de encaminhamento alternativo sequencial dependente explicitamente do tempo, adaptativo dependente do estado, adaptativo dependente do acontecimento e híbrido/misto, já definidos no capítulo 2.

## 3.2 Encaminhamento Alternativo Hierárquico

Nas primeiras redes hierárquicas, designadas por redes hierárquicas puras, para cada chamada existia apenas um caminho possível onde a chamada podia ser encaminhada. Esse caminho podia no entanto usar centrais de trânsito.

O método de encaminhamento Alternativo Hierárquico, descrito em [Girard90], foi o primeiro método de encaminhamento, e é o mais extensivamente usado nas redes de telecomunicações. As limitações tecnológicas que existiram impunham-no, sendo uma vantagem do método o facto de poder ser implementado com hardware muito simples e com limitada informação e controlo.

Em 1930 foi criado pela AT&T um plano em que ficou estabelecida a hierarquia dos centros de comutação. Inicialmente o encaminhamento na rede era efectuado manualmente por operadores. A primeira revisão do plano da estrutura hierárquica estabeleceu dez regiões e cinco níveis de hierarquia de sistemas de comutação centralizados nas centrais regionais. As centrais (também chamadas classes) de alto nível concentram sucessivamente o tráfego das áreas geográficas cada vez maiores. Isto é, como havia ainda limitações tecnológicas, a requerer garantia de integridade em termos de numeração,

sinalização, encaminhamento e transmissão, as redes foram estruturadas em diferentes níveis hierárquicos. Em 1951 foi introduzida a marcação directa à distância (“Direct Distance Dialing” - DDD) concorrentemente com o encaminhamento alternativo automático.

As redes hierárquicas com encaminhamento alternativo generalizaram-se a partir dos anos 50 com o desenvolvimento dos sistemas de comutação e de controlo que possibilitaram a introdução do conceito de encaminhamento alternativo baseado nas técnicas de transbordo.

Para evitar que uma chamada volte a uma mesma central dum dado caminho - o fenómeno da chamada em ciclo - o transbordo e a selecção de caminhos alternativos foi sujeito a regras hierárquicas. Além disso, devido às capacidades limitadas das centrais e à falta de capacidades de computação, o encaminhamento foi determinado na etapa de projecto e permanece fixo, em condições normais, até à próxima etapa de projecto. O encaminhamento hierárquico fixo foi assim estabelecido, e é ainda usado em numerosas redes quer electromecânicas, quer digitais. Como o seu nome indica é um tipo de encaminhamento alternativo fixo, com restrições adicionais impostas na natureza dos caminhos que podem ser usados por uma chamada. O carácter hierárquico do método vem da partição das centrais em classes.

No exemplo da figura seguinte existem, como se pode ver, cinco classes e dois tipos de circuitos a ligar as centrais. Os dois tipos de circuitos são os grupos de circuitos de grande utilização e os grupos de circuitos finais. Os grupos de circuitos de grande utilização ligam quaisquer duas centrais que tenham suficiente tráfego entre eles para fazer mais económica a atribuição de um ramo directo. E os grupos de circuitos finais são os feixes entre cada central e a sua imediatamente superior na hierarquia, juntamente com os feixes que interligam todas as centrais regionais. Uma central ligada por um feixe final a uma central de classe mais alta diz-se que tem casa (“*home*”) nessa central. Os feixes de grande utilização são dimensionados para manusear só uma porção do tráfego dirigido a eles, e o sistema de comutação redirige tráfego por encaminhamento alternativo automático para um feixe diferente quando todos os circuitos, de um feixe de grande utilização, estão ocupados. Em cada etapa, o plano de encaminhamento alternativo desloca as chamadas que transbordam dos caminhos mais directos para os caminhos finais.

A estrutura hierárquica apresentada na figura 3-1 tem cinco níveis. As classes com um número cada vez menor (nível hierárquico mais alto) concentram tráfego de áreas geográficas cada vez maiores. As centrais da classe 5, as chamadas centrais finais, são aquelas às quais as linhas dos assinantes se ligam directamente. Subindo na hierarquia são encontradas, as centrais locais, primárias, seccionais e regionais. Cada central tem uma e uma só central que é a sua “*casa*” à qual está ligada por um feixe final. Claro que, a central pode ser ligada a outras centrais na rede pelos feixes de grande utilização. Assim sendo, dado qualquer par de centrais, existe um único caminho, formado exclusivamente por

feixes finais, que as liga. Esse caminho costuma designar-se por caminho hierárquico, ou algumas vezes por caminho normal. A distância hierárquica entre duas centrais é definida como o número de grupos (grupos de feixes) necessários para ir de uma para a outra sob o caminho hierárquico. Por exemplo, numa hierárquica de nível cinco completa, a distância hierárquica entre duas centrais finais é 9, e a distância hierárquica entre uma central regional e uma central final é 5 (ou 4 se for a central da sua casa).

Os grupos finais são projectados para manusear o seu próprio tráfego directo e o tráfego de transbordo dos grupos de grande utilização com pequeno bloqueio. O facto de um circuito pertencer a um feixe final ou um feixe de grande utilização é uma propriedade da topologia da rede, ou seja, é independente do fluxo de tráfego particular considerado. Nos métodos de encaminhamento não hierárquicos todos os circuitos são do mesmo tipo, sendo isto que faz a grande distinção entre estes métodos e os hierárquicos. O método de encaminhamento hierárquico evita, como já foi dito, o ciclo de chamadas e o uso de um número excessivo de circuitos por uma chamada.

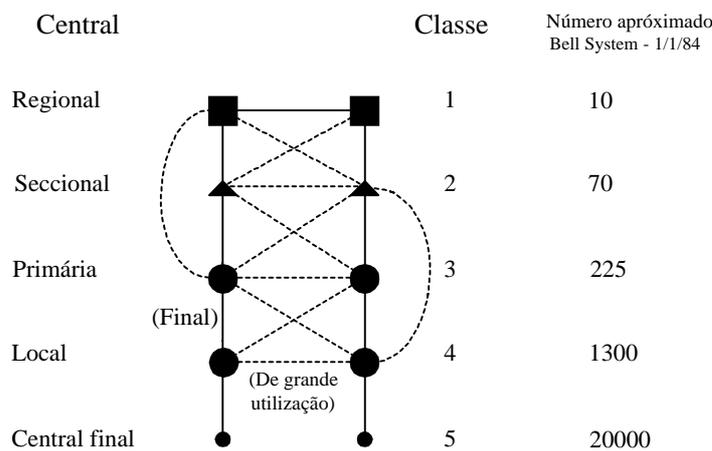


Fig. 3.1 - Encaminhamento hierárquico. Adaptada de [Ash95].

No método de encaminhamento hierárquico existe uma hierarquia de centrais e um conjunto de regras simples que podem facilmente ser implementadas com hardware limitado (a razão pela qual este método foi usado em primeiro lugar). O conjunto de regras é o seguinte [Girard90]:

1. - Para uma chamada que chega a uma central, é suficiente conhecer qual é a central destino para fazer a decisão de encaminhamento. A origem da chamada não é importante.
2. - Para uma chamada que chega a uma central, deve encaminhar-se no primeiro caminho disponível na sequência de caminhos alternativos possíveis.
3. - A sequência de caminhos alternativos é definida em função da central final. A regra é tentar o grupo que tem a extremidade mais próxima, no sentido de distância hierárquica, à central destino da chamada.

4. - A última escolha na lista é sempre o grupo hierárquico (feixe final) da central. Se a chamada é bloqueada neste grupo, é perdida; e por esta razão é chamado feixe final.

Se o conjunto de regras não for este será uma simples variação deste.

O método de encaminhamento hierárquico usado no sistema apresentado na figura 3-1:

- Usa caminhos com 9 feixes no máximo (com a estrutura hierárquica apresentada na figura 3-1);
- Não é alterado com o tempo;
- O controlo é do tipo sequencial (progressivo em direcção ao destino);
- O ramo directo com transbordo é tentado em primeiro lugar. Se não for possível encaminhar neste caminho, a chamada transborda para um caminho alternativo. O caminho alternativo seleccionado para encaminhar uma chamada será o descoberto se for usado o processo referido nos parágrafos anteriores. O último caminho alternativo tentado será o caminho formado exclusivamente por grupos de circuitos finais.

Com o surgimento das centrais Controladas por Programa Armazenado (“Stored Program Controlled” - SPC), o aparecimento de Sistemas de Sinalização por Canal Comum (“Common Channel Signaling Systems - CCS), o desenvolvimento de redes de dados e as novas capacidades de processamento permitiram a evolução do encaminhamento de tráfego hierárquico fixo para o encaminhamento flexível não hierárquico, para eliminar problemas que estão bem identificados:

- Como o encaminhamento do tráfego está profundamente relacionado com o projecto da rede nas redes hierárquicas com encaminhamento fixo, tornou-se claro que usando um padrão de encaminhamento simples - determinado de acordo com o tráfego na hora mais carregada, entre cada par (ordenado) de centrais - não se permite um escoamento eficiente do tráfego em todas as situações. Por outro lado, análises de desempenho de tráfego mostram que, em redes com encaminhamento fixo, existe alguma inconsistência entre a capacidade da rede e os pedidos de tráfego, o que conduz a um grau de serviço mais degradado em algumas partes da rede e a excessivas assimetrias nos bloqueios marginais de diferentes fluxos de tráfego. Isto é particularmente óbvio quando novos perfis de tráfego correspondendo à evolução sócio-económica são identificados. Em muitos casos, isto resulta num ineficiente sobreaprovisionamento da rede. Por outro lado, no caso de um aumento súbito do tráfego ou dum período de interrupção de partes da rede devido a avarias, a vantagem de reencaminhar o tráfego é claramente identificada como parte das acções de Gestão da Rede.
- A robustez da rede não é assegurada, também por causa das características hierárquicas do encaminhamento. Pouca adaptabilidade na sequência de transbordo torna difícil acomodar com

situações fora do normal. Partes da rede com capacidade disponível não podem ser usadas enquanto outras partes sofrem (sobre)congestão. O reencaminhamento manual, activado como uma parte de gestão da rede, é muito difícil de implantar e raramente implementado. Em vez disso, o tráfego de entrada é estrangulado, resultando num uso ineficiente da capacidade da rede.

O desenvolvimento das estruturas da rede que não utilizam predominantemente encaminhamento alternativo hierárquico conduziu a um aumento no número de circuitos ligados a uma central e a um aumento da sua capacidade de transporte de tráfego; o número de centrais é então reduzido. Também, devido à introdução de facilidades de transmissão de alta largura-de-banda, as centrais são ligadas através de feixes contendo um número elevado de circuitos. Consequentemente, a necessidade de concentrar tráfego em níveis organizacionais mais elevados é reduzida, e o número de níveis organizacionais tende a diminuir enquanto a conectividade aumenta. A nova estrutura da rede fornece um largo número de caminhos curtos possíveis que podem consistentemente ser usados para encaminhamento flexível. A rede torna-se então mais vulnerável no caso de um período de interrupção tal como avarias de centrais ou uma falha nas facilidades de transmissão, uma vez que afecta mais recursos da rede.

### **3.3 Método de encaminhamento sequencial alternativo fixo**

#### **3.3.1 Encaminhamento Alternativo Fixo (“Fixed Alternate Routing” - FAR)**

É um algoritmo determinístico pois o modo como se tenta transportar as chamadas obedece a uma ordem específica precalculada. FAR (descrito em [Mitra91a]) é um subconjunto do DNHR (descrito na secção seguinte) em termos de características, contudo os projectos não são relacionados. Ao considerar este método, Mitra foi influenciado por algumas evoluções, especialmente aquelas relacionadas com o conceito do custo de transportar uma chamada, e da aproximação do ponto fixo.

Neste método, o tráfego só é encaminhado alternativamente, em caminhos com dois feixes no máximo, se não for possível transportá-lo directamente. Implementa o mecanismo de controlo de reserva de circuitos, de modo que, uma chamada a ser encaminhada alternativamente só é transportada num caminho alternativo se esse critério é satisfeito em ambos os feixes do caminho.

No método de encaminhamento FAR é assumido serem conhecidas as intensidades de tráfego oferecido para todos os pares de centrais e estas intensidades serem constantes durante o período em que as sequências de encaminhamento são mantidas constantes.

A implementação é baseada numa tabela de encaminhamento que dá, para cada par de centrais, uma sequência ordenada de caminhos alternativos  $r(1), r(2), \dots, r(M)$ . Uma chamada oferecida a um par de centrais é oferecida primeiro a  $r(1)$ , depois a  $r(2)$ , etc.; a chamada é bloqueada e perdida se não pode ser transportada no caminho  $r(M)$ .  $r(1)$  é o caminho directo enquanto que  $r(2), \dots, r(M)$  são caminhos alternativos e  $M$  depende do par de centrais considerado.

## 3.4 Métodos de encaminhamento dependentes do tempo

### 3.4.1 Encaminhamento Dinâmico Não Hierárquico (“Dynamic Non-Hierarchical Routing” - DNHR)

Até 1984, o encaminhamento alternativo hierárquico foi o método de encaminhamento universal nas redes telefónicas em todas as partes do mundo. Nessa altura, uma grande mudança foi introduzida no encaminhamento da rede de longa distância AT&T nos Estados Unidos, o que representou uma quebra radical com métodos hierárquicos clássicos. Este método é chamado encaminhamento dinâmico não hierárquico.

Os velhos comutadores mecânicos (usados nas redes hierárquicas) foram substituídos pelos sistemas de comutação electrónicos controlados por computador (*Electronic Switching Systems* - ESS, no encaminhamento DNHR na rede AT&T - 4ESS), mais flexíveis, em que não é necessário manter a estrutura hierárquica das redes para utilizar o encaminhamento alternativo. Os sistemas de comutação electrónicos encontram-se interligados por feixes de Sinalização por Canal Comum (CCS).

O método DNHR é descrito em [Ash90], [Ash95], [Ash98], [Girard90] e [Watanabe90]. Todas as centrais numa rede DNHR são do mesmo nível funcional e não existe qualquer relação hierárquica entre elas. Este encaminhamento é ainda do tipo alternativo fixo, pois as sequências de caminhos são constantes, com a condição de que os caminhos alternativos são limitados a dois feixes, quando muito. Esta restrição foi introduzida porque é conhecido que os métodos de encaminhamento não hierárquico que permitem caminhos longos podem ser mais sensíveis a sobrecargas, além de aumentar a carga nos sistemas de sinalização e controlo.

A central origem mantém o controlo sobre o encaminhamento dinâmico da chamada até que seja completada ou bloqueada. Uma chamada bloqueada no segundo feixe de uma ligação com dois feixes, regressa à central origem para tentar outro caminho alternativo, se possível. O controlo é devolvido pelo envio de um sinal de retorno (*crankback*) por um canal de sinalização comum para a central origem. A técnica de *crankback* é então usada para identificar o bloqueio na segunda etapa de cada caminho alternativo de dois feixes, e uma chamada bloqueada numa central de trânsito será reencaminhada.

A operação de gestão da rede tem-se tornado mais centralizada, fornecendo a possibilidade de recolher informação global do estado da rede, necessária na etapa de projecto. Durante a fase de planeamento a oferta de tráfego é representada por um número de matrizes de tráfego, cada uma correspondendo a um período do dia. Das muitas alternativas possíveis temos como resultado do projecto a produção de uma sequência de caminhos quase óptima para cada um desses períodos, e para cada fluxo de tráfego. Cada uma das várias sequências de encaminhamento usa um subconjunto de caminhos numa ordem diferente. Terminando esta fase de projecto, estas tabelas de encaminhamento são armazenadas nas centrais, sendo seleccionada a tabela apropriada em cada período de tempo, sem ter em consideração o estado da rede. Podem ser armazenados até 14 caminhos nas centrais 4ESS por cada um dos 15 períodos de tempo. Os 14 caminhos armazenados para cada um dos períodos de tempo incluem os caminhos nominais (pré-determinados) e os caminhos de utilização em tempo real (à frente apresenta-se a respectiva definição). Adicionalmente o sistema de gestão em tempo real da rede pode juntar 7 caminhos aos 14 caminhos numa base de tempo real.

O DNHR é um método de encaminhamento dinâmico, pois muda as sequências de encaminhamento em intervalos de tempo fixo. O padrão de sequências de encaminhamento variáveis no tempo (ver figura 3-2) é preplaneado *off-line* de acordo com a não coincidência dos picos de tráfego previstos, para diferentes horas do dia. A eficiência do projecto desses padrões de encaminhamento depende da certeza com que se pode prever a carga na rede. Porque estas previsões são de natureza incerta, os encaminhamentos determinados não são necessariamente os melhores para as condições de tráfego encontradas na rede. As falhas e sobrecargas causam ineficiência, devido à natureza não adaptativa do método. O algoritmo de encaminhamento deve então ser capaz de reagir em tempo real, procurando capacidade livre numa base chamada após chamada, se necessário. Esta capacidade é oferecida pelos caminhos em tempo real. Quando os caminhos são determinados, os primeiros  $k$  melhores caminhos são destinados para o encaminhamento ordinário. Destes,  $m < k$  são disponíveis para encaminhamento, designados por caminhos “nominais”, os restantes  $k - m$  são reservados, designados por caminhos de utilização em tempo real, e são usados sempre que o desempenho da rede pareça estar a degradar-se. Os caminhos em tempo real oferecem então possibilidades adicionais para completar chamadas que de outro modo seriam bloqueadas. O uso destes caminhos em tempo real não

deve, no entanto, causar um aumento da perda do tráfego para o qual esses caminhos foram originalmente projectados. Isto é, estes caminhos só são usados quando capacidade livre adicional está disponível. Para conseguir isso, é aplicado aos caminhos em tempo real, uma técnica de reserva de junções baseada num nível de reserva fixo, de modo que estes caminhos não sejam usados a menos que a ocupação, na altura em que eles sejam necessários, esteja abaixo desse nível. Isto garante uma interferência mínima com os caminhos nominais, evitando que as chamadas que normalmente usam um feixe sejam trocadas pelas chamadas em tempo real. Sobrecargas profundas e certas avarias podem resultar em tráfego que transborda os caminhos em tempo real. Quando isto ocorre, caminhos adicionais podem ser implementados automaticamente ou manualmente como um resultado da gestão da rede.

A gestão em tempo real da rede DNHR é controlada por um sistema centralizado simples, o qual usa medidas de tráfego obtidas em cada central 4ESS e analisa estes dados todos os cinco minutos para monitorizar e controlar a disponibilidade da rede. Padrões de encaminhamento especialmente adaptados para maximizar a eficiência nos dias de pico - tais como o Natal ou o dia da Mãe - podem ser colocados na rede para uso nesses dias. O encaminhamento dinâmico não hierárquico usa portanto um mecanismo de selecção de caminho híbrido variável no tempo e em tempo real (implementação simples do encaminhamento adaptativo) para responder às variações de carga da rede.

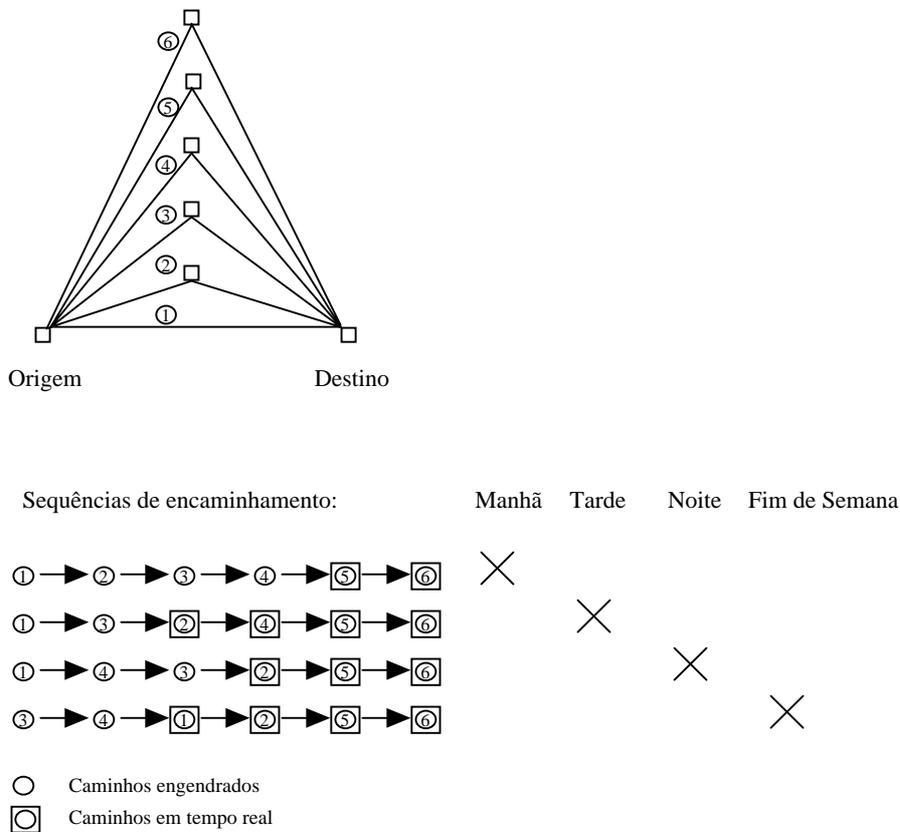


Fig. 3.2 - Método de encaminhamento dinâmico não hierárquico. Adaptada de [Ash90].

A reserva de circuitos, neste método, só é aplicada aos caminhos em tempo real.

## **3.5 Métodos de encaminhamento dependentes do estado**

A generalizada introdução de centrais digitais (5ESS ou DMS100) e de sistemas de transmissão digital combinada com a sinalização por canal comum tornou possível métodos de encaminhamento mais sofisticados. Exemplos de métodos de encaminhamento adaptativos em tempo real baseados na informação de estado da rede são: “Dynamically Controlled Routing” (ou “Dynamic Call Routing”), “Least Loaded Routing”, “Aggregated Least Busy Alternative Routing”, “Real Time Network Routing”, “Separable State-Dependent Routing”. São estes os métodos que são descritos nesta secção fazendo-se uma divisão em dois grupos. Um dos grupos considerado é o grupo dos métodos baseados na capacidade residual e o outro grupo é o dos métodos baseados no processo de decisão de Markov. Vamos começar a descrição pelos métodos baseados na capacidade residual.

### **3.5.1 Métodos Baseados na Capacidade Residual (“Residual-Capacity Adaptative Routing” - RCAR)**

Nos métodos de encaminhamento do tipo Encaminhamento Adaptativo de Capacidade Residual (Residual Capacity Adaptative Routing - RCAR) é usada informação do estado de todos os grupos de circuitos da rede, informação esta actualizada periodicamente através de medidas feitas pelas centrais. As chamadas que não possam ser encaminhadas no caminho directo são oferecidas a um caminho de transbordo, escolhido, entre os caminhos potenciais, com base na capacidade residual desses caminhos. A capacidade residual de um feixe é definida pelo número total de circuitos desse feixe menos o número de circuitos ocupados, menos o número de circuitos reservados. A capacidade residual de um caminho de transbordo é o valor mínimo entre a capacidade residual dos feixes constituintes. Na altura do encaminhamento de uma chamada, os métodos determinam a capacidade residual do caminho para saber se ele pode ser usado para encaminhar a chamada. Para escolher o caminho alternativo a usar, alguns encaminhamentos usam uma estimativa da capacidade residual, enquanto outros usam o valor da capacidade residual determinado em tempo real. O DCR é um exemplo dos primeiros e o RTNR é um exemplo dos segundos. Isto é, no RTNR a determinação do caminho alternativo é feita em tempo real enquanto que no DCR o caminho alternativo é determinado apenas periodicamente.

### **3.5.1.1 Encaminhamento no caminho de menor carga (“Least Loaded Routing” - LLR)**

O LLR (descrito em [Vargas96] e [Chung93]) é um método de encaminhamento adaptativo o qual tenta encaminhar as novas chamadas na parte da rede menos ocupada.

Nesta estratégia é calculada a capacidade residual dos caminhos, para todos os caminhos do conjunto de caminhos potenciais, e é seleccionado, para a chamada que acaba de chegar, o caminho com a máxima capacidade residual. Se não existe um caminho disponível com circuitos livres, então a chamada é bloqueada. Esta regra de selecção só deve ser aplicada ao conjunto de caminhos alternativos, ou seja para as chamadas que são bloqueadas na sua primeira escolha.

Neste método de encaminhamento quando chega uma chamada para o par OD (par Origem-Destino), a chamada começa por tentar o caminho directo. Se houver circuitos disponíveis, então a chamada é aceite e encaminhada nesse caminho. Senão, então o caminho alternativo com menor carga (isto é, aquele com maior capacidade residual) é considerado.

Supondo que o caminho alternativo com menor carga é o caminho  $z$ , então uma chamada será aceite e encaminhada nele se a capacidade residual no mesmo for maior do que  $T$  (parâmetro de reserva de circuitos); caso contrário a chamada é bloqueada.

No caso de existirem dois ou mais caminhos alternativos de menor carga, o empate é quebrado por uma sequência predeterminada na qual o conjunto de caminhos alternativos é ordenado.

### **3.5.1.2 Encaminhamento Alternativo no Agregado de Menor Carga (“Aggregated Least Busy Alternative Routing” - ALBA)**

O método de encaminhamento ALBA (descrito em [Mitra91b], [Mitra93] e [Vargas96]) tenta encaminhar as chamadas novas nas partes da rede menos ocupadas, como o faz o LLR, mas utilizando um conjunto reduzido de estados de ocupação dos feixes.

No método de encaminhamento ALBA o caminho para uma chamada é determinado pela informação local do estado dos feixes dos caminhos possíveis na altura em que a chamada ocorre. ALBA é então uma estratégia de encaminhamento distribuída, dinâmica e dependente do estado.

Sendo ALBA um encaminhamento dependente do estado, o estado da rede na altura em que a chamada ocorre determina se a chamada é aceite e, se o for, determina também qual o caminho para o seu estabelecimento. Note-se que a formulação apresentada nesta secção apenas se aplica a redes completamente emalhadas e simétricas (isto é, com o mesmo número de circuitos em cada feixe).

Esta estratégia de encaminhamento, é baseada no conhecimento dos estados de agregação de cada feixe. A agregação refere-se à característica de não se usar informação detalhada; em vez disso usa-se uma representação grosseira. A implementação torna-se mais simples quanto maior for a agregação.

Podem-se considerar dois casos particulares de ALBA os quais representam dois extremos de agregação. O caso particular de ALBA no qual não existe agregação é designado por Least-Busy-Alternative (LBA) e o outro extremo de agregação no qual o número de estados agregados é 2 e é designado por ALBA(2).

No encaminhamento ALBA os caminhos são restringidos a ter quando muito 2 feixes. E a informação usada para tomar as decisões de encaminhamento baseia-se nos estados agregados dos feixes. ALBA( $K$ ) indica a versão na qual o número de estados agregados para cada feixe é  $K$  (igual para todos os feixes). Os  $(C + 1)$  estados, representando o número de circuitos ocupados em cada feixe, com  $C$  canais, são agrupados no ALBA( $K$ ) em  $K$  agregados  $\{A_0, A_1, A_2, \dots, A_{K-1}\}$  por ordem crescente do nível de ocupação. Quando o número de circuitos ocupados num feixe é  $i$  e  $i \in A_l$ , então o feixe é dito estar no estado agregado  $l$ .

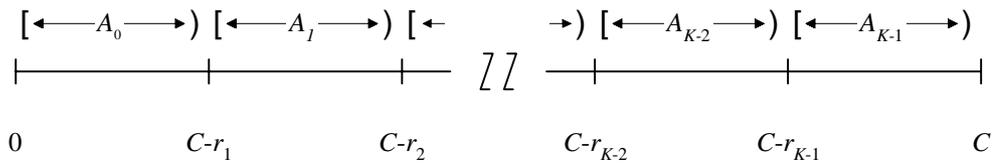


Fig. 3-3 –  $K$  estados agregados num feixe com  $C$  circuitos. Adaptada de [Mitra93].

Como ilustrado na figura anterior, os conjuntos  $\{A_l\}$  são definidos como segue:

$$A_0 = \{0, 1, \dots, C - r_1 - 1\}$$

$$A_1 = \{C - r_1, C - r_1 + 1, \dots, C - r_2 - 1\}$$

...

$$A_{K-1} = \{C - r_{K-1}, C - r_{K-1} + 1, \dots, C\}$$

Onde:

$$C > r_1 > r_2 > \dots > r_{K-1} > 0$$

Então à medida que a agregação aumenta, isto é, à medida que o  $K$  diminui, a informação de estado torna-se cada vez mais grosseira.

No ALBA( $K$ ) o estado agregado  $A_{K-1}$  é especial: o último agregado  $A_{K-1}$  é o conjunto de estados com  $r_{K-1}$  ou menos circuitos livres. Quando um feixe está neste estado agregado é-lhe oferecido

apenas tráfego directo, não lhe é oferecido nenhum tráfego de transbordo; por esta razão, o feixe é considerado estar no estado reservado. Por isso a  $r_{K-1}$  é chamado parâmetro de reserva de circuitos, e também será denotado por  $r$ , simplesmente.

No encaminhamento ALBA, quando uma chamada ocorre, o primeiro caminho a ser tentado é o caminho directo, e a chamada é transportada nesse caminho se existir um circuito livre. Caso contrário, a chamada tenta o caminho agregado menos ocupado de dois feixes. Um caminho agregado menos ocupado para o par origem-destino  $i, j$  é um que minimiza o  $\max(I_{ik}, I_{kj})$  onde  $k$  é uma central intermédia e  $I_{ik}$  e  $I_{kj}$  são os estados agregados dos feixes  $(i, k)$  e  $(k, j)$ , respectivamente. Qualquer empate no caminho agregado menos ocupado é quebrado aleatoriamente. A chamada é aceite no caminho de dois feixes se satisfaz o critério de reserva de circuitos, nomeadamente, a aceitação da chamada não deixa nenhum dos dois feixes no estado agregado  $A_{K-1}$ . Caso contrário a chamada é bloqueada. O parâmetro de reserva de circuitos deve ser cuidadosamente determinado, assim como os limites dos estados agregados.

Como já referido ALBA(2) e LBA representam os dois extremos de agregação. Os dois estados agregados de ALBA(2) são o conjunto de estados reservados e o conjunto de estados não reservados. Em contraste, LBA é obtido se todos menos o último agregado forem compostos por um estado único, isto é, nas decisões de encaminhamento é usado o conhecimento exacto da ocupação de todos os feixes em todos os caminhos alternativos.

Fazendo as decisões de encaminhamento baseadas apenas na informação sobre os estados agregados dos feixes nos caminhos alternativos, consegue-se uma simplificação na implementação, e uma redução no tráfego de sinalização. Mitra et al [Mitra93] argumentam que a perda de desempenho pela agregação é pequena com um projecto apropriado de agregação, e que o uso de informação de estado grosseira para tomar as decisões de encaminhamento é atractivo.

Repare que a única diferença entre o LLR e o LBA tem a ver com a ordenação dos caminhos. No LBA não existe nenhuma ordem nos caminhos para cada par de centrais ao contrário do que se passa no LLR. Esta ordenação é utilizada no LLR na determinação do caminho de menor carga no caso de empate dos caminhos alternativos de menor carga.

### **ALBA Geral**

Anteriormente ALBA foi formulado apenas para redes simétricas. De seguida, iremos apresentar uma generalização de ALBA, que cobre o caso de redes assimétricas. O ALBA generalizado (descrito em [Vargas96]) inclui o LLR e o ALBA como casos especiais. O ALBA generalizado é denotado por  $ALBA(K_{ij})$ , onde o conjunto de estados para o feixe  $(i, j)$  é dividido em  $K_{ij}$  subconjuntos cada um

dos quais é então tratado como um estado (agregado) para esse feixe. Cada feixe  $(i, j)$  tem então  $K_{ij}$  estados agregados denotados por  $A_0^{ij}, A_1^{ij}, \dots, A_{K_{ij}-1}^{ij}$ . Para  $m = 0, 1, 2, \dots, K_{ij} - 2$ , o  $m$  ésimo estado agregado para o feixe  $(i, j)$ ,  $A_m^{ij}$ , é dado pelo conjunto de estados  $\{C_{ij} - r_m^{ij}, C_{ij} - r_m^{ij} + 1, \dots, C_{ij} - r_{m+1}^{ij} - 1\}$ , onde  $r_m^{ij}$  é um limiar para o feixe  $(i, j)$  é onde  $r_0^{ij} = C_{ij} > r_1^{ij} > r_2^{ij} > \dots > r_{K_{ij}-1}^{ij} > r_{ij}^{ij} = 0$ . O estado agregado  $A_{K_{ij}-1}^{ij}$  é dado pelo conjunto de estados  $\{C_{ij} - r_{K_{ij}-1}^{ij}, \dots, C_{ij}\}$ . O estado agregado  $A_{K_{ij}-1}^{ij}$  consiste nos estados reservados para o tráfego directo do par  $[i, j]$ .

Se os dois feixes de um caminho alternativo,  $(i, j)$  e  $(j, k)$ , estão no estado agregado  $A_l$  e  $A_m$ , respectivamente, então o caminho alternativo é dito estar no estado  $A_n$  onde  $n = \max\{l, m\}$ .

A política de encaminhamento para o ALBA( $K_{ij}$ ) é a seguinte: quando uma chamada ocorre no par origem-destino  $[i, j]$ , a chamada tenta primeiro o ramo directo  $(i, j)$ . Se existe algum circuito livre, então a chamada é aceite e encaminhada no caminho directo. Caso contrário é considerado o caminho alternativo com o estado agregado com menor número. Supondo que este é o caminho alternativo  $(i, v)$ ,  $(v, j)$ , a chamada é transportada nesse caminho se os feixes  $(i, v)$  e  $(v, j)$  não estão no estado agregado  $A_{K_{iv}-1}$  e  $A_{K_{vj}-1}$ , respectivamente. Caso contrário a chamada é rejeitada. Os empates são desfeitos por uma sequência predeterminada na qual o conjunto de caminhos alternativos é ordenado.

O LLR pode ser obtido do ALBA( $K_{ij}$ ) considerando cada estado agregado como um conjunto de um único elemento com a excepção do estado agregado  $A_{K_{ij}-1}^{ij}$  o qual compreende todos os estados os quais estão reservados para tráfego directo. Igualmente, ALBA( $K$ ) para redes simétricas pode ser obtido escolhendo todos os  $K_{ij}$  iguais a  $K$ . Para redes assimétricas para definir o ALBA( $K$ ), todos os  $K_{ij}$  devem ser iguais a  $K$ . Aqui deve-se ter cuidado na escolha dos limiares dos estados agregados, caso contrário podem obter-se algoritmos complicados. Uma escolha possível é, por exemplo, escolher os valores dos limiares de modo que os estados agregados num feixe reflectam uma percentagem da capacidade do feixe. Assim sendo, por exemplo quando  $K = 4$  cada estado agregado num feixe representa o estado de ocupação de 25% dos circuitos presentes no feixe.

Note que, o ALBA generalizado inclui o LLR porque se definiu como tendo os caminhos ordenados.

### 3.5.1.3 Encaminhamento em Tempo Real (“Real Time Network Routing” - RTNR)

O RTNR (descrito em [Ash91], [Ash93], [Ash95] e [Ash98]) é um método de encaminhamento adaptativo que veio substituir o encaminhamento DNHR na rede AT&T, tendo a substituição começado em 1991. O RTNR foi introduzido para estender o encaminhamento dinâmico a todos os serviços, quer aos serviços novos quer aos serviços existentes, e para aumentar a robustez da rede. Com o RTNR, as centrais têm um modo simples de trocar a informação do estado dos feixes pela utilização de *bit maps*, deste modo determinando as condições de disponibilidade e carga de todos os caminhos de dois feixes até ao destino.

O estado de ocupação - disponibilidade dos feixes é trocado entre as centrais da rede usando a troca de dados *bit map* através da rede CCS, e as chamadas são estabelecidas onde exista mais capacidade livre na rede.

Ash [Ash93] desenvolveu um modelo analítico para estas redes. Até então as análises das redes RTNR tinham sido só feitas com modelos simulacionais.

O RTNR é uma estratégia descentralizada, logo não precisa do desenvolvimento de um controlador central. Evita assim os riscos associados com um processador central que controle o encaminhamento de toda a rede. Por isso, o RTNR é uma estratégia prometedora na qual as classes de serviço dinâmicas da rede AT&T têm sido completamente implementadas desde 1991.

As centrais, ao usar o RTNR, seleccionam primeiro o ramo directo entre a central origem, designada aqui por  $OS_j$  (Originating Switch), e a central final, designada por  $TS_k$  (Terminating Switch). Quando não existe capacidade directa disponível, a central origem testa as condições de disponibilidade e de carga de todos os caminhos de dois feixes até à central final, numa base chamada após chamada. Se qualquer destes caminhos de dois feixes está disponível, a chamada é estabelecida sobre o caminho de dois feixes com menor carga. As cargas de tráfego são equilibradas dinamicamente através dos feixes ao longo da rede para maximizar o estabelecimento de chamadas.

O RTNR tem a característica importante de permitir a partilha da rede por múltiplos serviços heterogéneos e de permitir a adaptação de parâmetros básicos da estratégia de encaminhamento em tempo real (reserva de circuitos, profundidade de selecção do caminho, etc.).

Numa rede com integração de serviços (múltiplos serviços), uma chamada para um serviço particular  $i$  é considerada consumir uma largura de banda média igual a  $r^i$ . Sendo usada uma unidade de capacidade única denotada por circuito virtual VT (Virtual Trunk) utilizada para designar a largura de banda média consumida por uma classe de serviço. Por exemplo numa chamada de voz cada VT terá

$r^i$  igual a 64 Kbps de largura de banda, numa chamada de um serviço digital comutado de 384 Kbps cada VT terá  $r^i$  igual a 384 Kbps de largura de banda, etc.

Devido à complexidade do método, vamos começar por analisar o caso de uma rede com um único serviço, só depois passaremos ao caso de uma rede com vários serviços.

### 3.5.1.3.1 Encaminhamento com apenas um serviço

#### Troca dos Bit Maps do estado da rede

Na figura seguinte pode ser observado o processo de determinação dos caminhos candidatos a caminho alternativo.

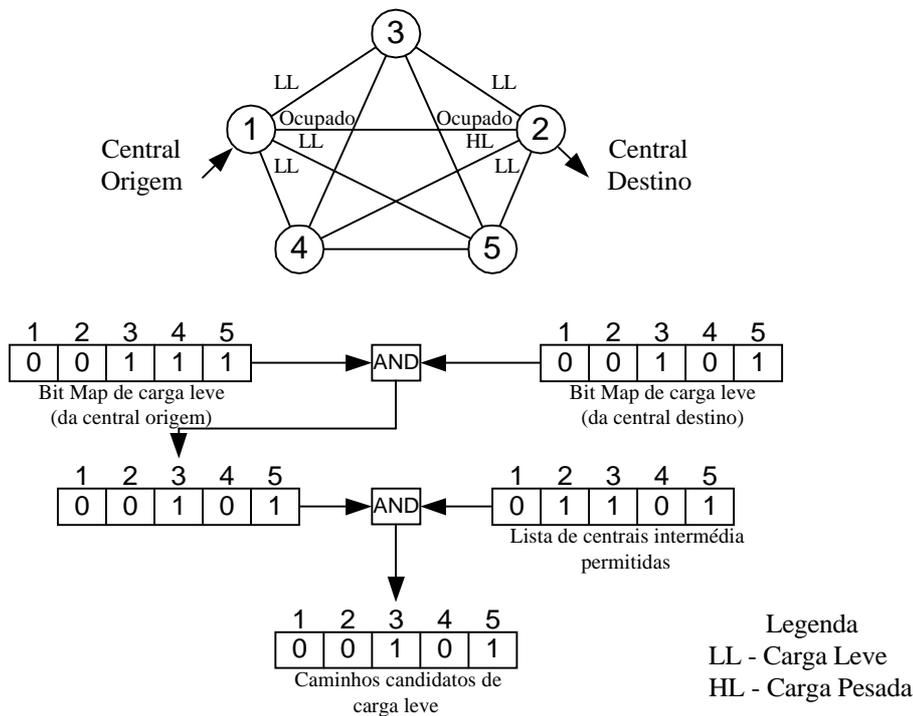


Fig. 3-4 - Determinação do caminho com menor carga. Adaptada de [Ash93].

Um caminho disponível de dois feixes de  $OS_j$  para  $TS_k$  é um caminho que tem como central intermédia uma central para a qual ambas as centrais,  $OS_j$  e  $TS_k$ , têm capacidade livre, isto é, nenhum dos dois feixes está ocupado. Um caminho de dois feixes disponível é considerado com carga leve se o número de circuitos virtuais livres em ambos os feixes exceder um limiar. Para determinar todas as centrais na rede que satisfazem este critério, a central origem envia uma mensagem ao  $TS_k$  sob a rede CCS, pedindo-lhe para enviar uma lista das centrais para as quais têm feixes com carga leve. Ao receber esta lista de centrais do  $TS_k$ , a central origem compara esta lista com a sua própria lista de centrais para as quais têm feixes com carga leve. Qualquer central que aparece em ambas as

listas tem actualmente feixes com cargas leves para  $OS_j$  e  $TS_k$ , e portanto pode ser usada como uma central intermédia para uma ligação de dois feixes para essa chamada. Na figura anterior há dois caminhos com cargas leves entre  $OS_j$  e  $TS_k$ , que são os caminhos que têm como central intermédia a central 3 ou a central 5.

Os identificadores das centrais usados na lista de centrais enviada pelo  $TS_k$  devem ser reconhecidos pela central origem, por isso a cada central na rede é atribuído um número único de central na rede NSN (NSN – Network Switch Number); estes NSNs são usados como identificadores das centrais. No exemplo da figura, há cinco centrais na rede aos quais foram atribuídos NSNs arbitrários. Com esses NSNs atribuídos, uma lista de centrais pode ser representada por um *bit map* que tem uma entrada de um *bit* para cada NSN na rede. Na figura anterior, no *bit map* correspondente a  $TS_k$ , uma entrada é colocada a “1” para cada NSN que tenha um feixe com carga leve para  $TS_k$ .  $OS_j$  também mantém o seu próprio *bit map* listando cada NSN que tem um feixe com carga leve para si ( $OS_j$ ). Tendo os *bit maps* feitos é muito fácil e eficiente para a central origem encontrar todos os caminhos de dois feixes com carga leve. A central origem simplesmente faz a conjunção lógica (AND) do *bit map* que recebe de  $TS_k$ , o qual lista todos os feixes de saída de  $TS_k$  com carga leve, com o seu próprio *bit map* para produzir um novo *bit map* que identifica todas as centrais intermédias com feixes com carga leve para ambas as centrais,  $OS_j$  e  $TS_k$ . Os *bit maps* são também um modo muito compacto de armazenar a informação das ligações de uma central a todas as outras centrais da rede. Isto é uma consideração importante dado a lista ser enviada numa mensagem CCS. No caso mais simples, são necessários apenas 16 *bytes* de dados para codificar um dado estado de carga de uma rede com 128 centrais.

Dado que alguns dos caminhos de dois feixes podem não fornecer boa qualidade de transmissão de voz, a administração da rede pode restringir a selecção dos caminhos de dois feixes aos que garantem boa qualidade de transmissão. Isso é conseguido através do uso de outro *bit map*, que lista as centrais intermédias permitidas, e que especifica quais são as centrais intermédias aceitáveis de  $OS_j$  para  $TS_k$ . Como se pode também observar na figura anterior, fazendo o AND desse *bit map* com o *bit map* contendo as centrais intermédias de todos os caminhos de dois feixes disponíveis, removem-se as centrais intermédias em caminhos com qualidade de transmissão inaceitável.

Quando dois ou mais caminhos disponíveis têm o mesmo estado de carga, como na figura onde foram encontrados dois caminhos com cargas leves, a central origem selecciona um desses caminhos para ser usado pela chamada. Para seleccionar um caminho para essa chamada particular para  $TS_k$ , a central origem começa uma pesquisa circular através da lista *bit map* dos caminhos com o mesmo estado de

carga, começando com a entrada imediatamente a seguir à central intermédia por último usada para uma chamada para  $TS_k$ .

### Determinação do estado de carga dos feixes.

Para equilibrar dinamicamente as cargas de tráfego através da rede, é necessário conhecer a condição de carga actual dos feixes ao longo da rede, e usar os caminhos de dois feixes para os quais os feixes tenham a menor carga. O RTNR usa 6 estados de carga discretos para os feixes, mas é possível também usar apenas 3 ou 4. No exemplo da figura 3-4 foram considerados apenas 4 estados. Se forem usados seis estados eles serão: LL1 – carga leve máxima (LL1 maximum **L**ightly **L**oaded), LL2 – carga leve média (LL2 medium **L**ightly **L**oaded), LL3 – carga leve mínima (LL3 minimum **L**ightly **L**oaded), HL – carga pesada (HL **H**eavily **L**oaded), R – Reservado (R **R**eserved) e B – Ocupado (B **B**usy). O número de circuitos virtuais livres num feixe é comparado com os limiares do estado de carga para o feixe para determinar as condições de carga desse feixe. Esta determinação é feita todas as vezes que um circuito virtual é libertado ou ocupado num feixe. A central final identifica todas as centrais para as quais tem capacidade livre, e a condição de carga dos feixes para cada uma dessas centrais, e envia uma mensagem de resposta para a central origem. A central final faz isto enviando múltiplas listas *bit map* nessa mensagem de resposta; uma lista *bit map* das centrais para as quais tem um feixe LL1, uma lista *bit map* das centrais para os quais tem um feixe LL2 (ou com menor carga), etc. A central origem mantém listas *bit map* similares, das centrais para as quais tem capacidade livre, ordenadas pelas condições de carga dos feixes para essas centrais. A central origem primeiro compara a lista de centrais para as quais tem um feixe LL1 com a lista de centrais para as quais a central final tem um feixe LL1 para determinar se alguma central aparece em ambas as listas. Caso contrário, a central origem compara sucessivamente as outras listas do estado de carga para encontrar uma central intermédia com os feixes disponíveis e com carga o menor possível. No RTNR a agregação é também uma questão central.

Cada central origem  $OS_j$  mantém, para cada destino  $TS_k$ , as estimativas correntes de quatro quantidades, para com elas determinar os limiares do estado de carga de um feixe particular (feixe de  $OS_j$  para  $TS_k$ ). Essas quantidades são:

1. o número corrente de chamadas em progresso,  $CIP_k$  ( $CIP_k$  **C**alls **I**n **P**rogress), o qual é o número de chamadas activas de  $OS_j$  para  $TS_k$ . As chamadas em progresso são medidas, à medida que ocorrem, pelas centrais e incluem as chamadas estabelecidas nas ligações directas e de dois feixes;
2. o nível de bloqueio corrente ponto a ponto da central  $OS_j$  para a central  $TS_k$ ,  $NN_k$  ( $NN_k$  **N**ode-**N**ode blocking), o qual é a taxa de bloqueio para chamadas de  $OS_j$  para  $TS_k$ . Aqui

$NN_k = OV_k / PC_k$  onde  $OV_k$  ( $OV_k$  **O**Verflow count) e  $PC_k$  ( $PC_k$  **P**eg Count) são contadores de chamadas bloqueadas e de chamadas oferecidas, respectivamente, para o destino sobre o intervalo de actualização periódico de 3 minutos.  $NN_k$  é estimado em tempo real pela central a cada 3 minutos;

3. a carga de tráfego oferecida,  $TL_k$  ( $TL_k$  offered **T**raffic **L**oad), para cada uma das outras centrais na rede. Esta aproximação é baseada no número de chamadas em progresso e no bloqueio de chamadas para cada central;

4. e  $VTtraf_k$  que é o número de circuitos virtuais necessários para atingir o grau de serviço pretendido para a carga de tráfego oferecida corrente de  $OS_j$  para  $TS_k$ . Considera-se que:

$$VTtraf_k = 1.1 \times TL_k,^1$$

$VTtraf_k$  é estimado em tempo real pela central a cada 3 minutos e pode exceder o número total de circuitos virtuais directos entre  $OS_j$  para  $TS_k$ ; esta condição significa que são necessários os circuitos virtuais de caminhos alternativos de dois feixes para atingir o grau de serviço pretendido para a corrente carga de tráfego oferecida.

O valor do limiar entre dois quaisquer estados de carga de um feixe é tanto maior quanto maior for  $VTtraf_k$ . Os limiares do estado de carga altos reduzem a oportunidade do feixe ser usado na ligação de chamadas em caminhos com dois feixes; isto faz com que o feixe transporte mais tráfego directo conseguindo assim tratar melhor a carga de chamadas entre as centrais ligadas pelo feixe. O limiar de estado reservado é baseado no nível de reserva  $R_k$  ( $R_k$  **R**eservation level) calculado em cada feixe, o qual por sua vez é baseado no nível de bloqueio  $NN_k$ , como apresentado na tabela seguinte.

$NN_k$ - Nível de Bloqueio	Nível de Reserva	$R_k$ (VTs)
[0, .01]	0	0
(.01, .05]	1	$.05 \times VTtraf_k$
(.05, .15]	2	$.1 \times VTtraf_k$
(.15, .5]	3	$.15 \times VTtraf_k$
(.5, 1]	4	$.2 \times VTtraf_k$

Tab. 3-1 - Nível de reserva de circuitos. Adaptada de [Ash93].

<sup>1</sup> O valor 1.1 foi obtido pelos autores após análises simulacionais extensivas.

Existem vários níveis de reserva de modo que, se o bloqueio  $NN_k$  exceder ou se tornar inferior a um determinado limiar,  $R_k$  é alterado imediatamente o que melhora a resposta das centrais e da rede a variações de carga e falhas.

O número de circuitos virtuais reservados  $Rtraf_k$  é a diferença entre o  $VTtraf_k$  estimado e o número corrente de chamadas em progresso  $CIP_k$ , com um limite superior igual a  $R_k$ , como segue:

$$Rtraf_k = \min[R_k, \max(0, VTtraf_k - CIP_k)] \quad (3.1)$$

A figura seguinte ilustra o funcionamento deste mecanismo de reserva de circuitos.

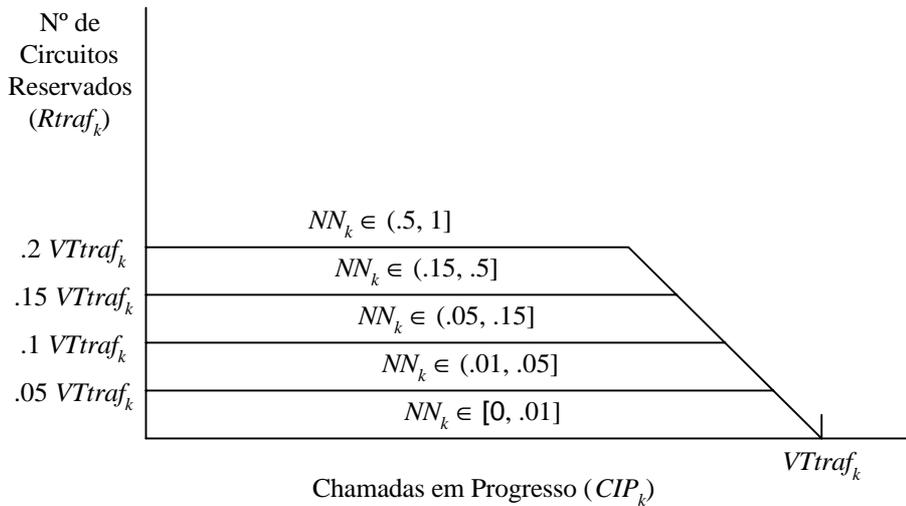


Fig. 3-5 – Operação da reserva de circuitos no RTNR. Adaptada de [Ash93].

$VTtraf_k - CIP_k$  é o número de chamadas adicionais da central origem para  $TS_k$  que precisam ser estabelecidas para atingir o valor estimado de  $VTtraf_k$ .

Cada central testa continuamente a taxa de bloqueio de chamadas para cada uma das outras centrais na rede. Quanto maior o bloqueio ponto a ponto, maior é o número de circuitos virtuais reservados  $Rtraf_k$  para o tráfego directo. A central origem ajusta  $Rtraf_k$  à medida que o número de chamadas estabelecidas para  $TS_k$  ( $CIP_k$ ) muda. Todas as vezes que a central origem estabelece uma nova chamada para  $TS_k$ , através de um ramo directo ou de um caminho de dois feixes, a central origem testa a ver se o limiar do estado reservado do feixe para  $TS_k$  deve ser decrementado. Do mesmo modo, quando uma chamada para  $TS_k$  termina, a central origem testa a ver se o limiar  $Rtraf_k$  deve ser incrementado. Seguindo estas acções, a central origem altera o controlo de reserva deste feixe para

activo ou inactivo, à medida que o número de chamadas em progresso para  $TS_k$  oscila à volta do valor de  $VTraf_k$  estimado.

À medida que a carga de tráfego aumenta, o número de circuitos reservado aumenta, e à medida que a carga de tráfego diminui, o número de circuitos livres necessários para ter feixes com carga leve diminui. Em condições normais, sem bloqueio na rede, todo o tráfego partilha completamente toda a capacidade disponível na rede.

Quando o mecanismo de reserva de circuitos é activado deixa de ser possível encaminhar tráfego alternativamente nesses feixes. À medida que se tornam livres circuitos virtuais nos feixes (chamadas antigas são desligadas), a capacidade é reservada para novas chamadas que surjam entre a central origem e  $TS_k$ . Uma vez que a central origem descubra que o seu grau de serviço pretendido para as chamadas para  $TS_k$  está de novo a ser satisfeito, o limiar de estado reservado para o feixe entre essas duas centrais é colocado a zero.

O RTNR também permite que exista um número  $n$  de circuitos reservados constantemente, devendo para isso ser somada a quantidade  $n$  ao lado direito da expressão anterior ( equação 3-1).

Na tabela seguinte temos valores dos limiares para determinar os diferentes estados:

Nome do limiar	Nível do limiar (VTs)
$TK1_k$	$.05 \times VTraf_k$
$TK2_k$	$.1 \times VTraf_k$
$TK3_k$	$.2 \times VTraf_k$

Tab. 3-2 – Limiares do estado de carga dos feixes. Adaptada de [Ash93].

Sendo  $ILVT_k$  o número de circuitos virtuais livres de  $OS_j$  para  $TS_k$  ( $ILVT_k$  **IdLe Virtual Trunks**), então o estado de carga dos feixes é dado na tabela seguinte:

Circuitos virtuais livres no feixe	Estado de carga
$ILVT_k = 0$	Ocupado
$1 \leq ILVT_k \leq Rtraf_k$	Reservado
$Rtraf_k < ILVT_k \leq TK1_k + Rtraf_k$	HL
$TK1_k + Rtraf_k < ILVT_k \leq TK2_k + Rtraf_k$	LL3
$TK2_k + Rtraf_k < ILVT_k \leq TK3_k + Rtraf_k$	LL2
$TK3_k + Rtraf_k < ILVT_k$	LL1

Tab. 3-3 – Definição do estado de carga dos feixes. Adaptada de [Ash93].

A determinação do estado de carga dos feixes é realizada todas as vezes que um feixe é libertado ou ocupado. Se a chamada foi estabelecida no caminho directo a determinação do estado de carga é efectuada para o ramo directo (em ambas as direcções). Senão, se a chamada foi estabelecida num caminho de dois feixes a determinação do estado de carga é efectuada no ramo directo e nos feixes do caminho usado (em ambas as direcções).

**Alocação de tráfego para o caminho directo e para os caminhos de dois feixes**

Vamos neste ponto descrever mais pormenorizadamente o processo de selecção do caminho mais desejável na rede de entre todos os caminhos candidatos, baseado na informação de estado da rede como apresentado anteriormente.

Como foi dito inicialmente, quando uma chamada chega esta tenta o caminho directo; se este não tiver capacidade disponível suficiente para estabelecer a chamada, então um caminho alternativo de dois feixes vai ser pesquisado. Se for possível estabelecer a chamada num caminho alternativo sê-lo-à no caminho alternativo disponível que tiver carga mais leve. Na figura seguinte podemos ver as regras de selecção do caminho directo e de dois feixes para uma chamada.

Estado do Caminho	Ordem de selecção
Directo	1
Através de LL1	2
Através de LL2	3
Através de LL3	4
Através de HL	5 (de utilização controlada)
Através de Reservados	6 (de utilização controlada)

Tab. 3-4 – Ordem de selecção do caminho. Adaptada de [Ash93].

A sequência de caminhos consiste no caminho directo, se existir, no caminho de carga leve máxima, no caminho de carga leve média, no caminho de carga leve mínima, no caminho de carga pesada e no caminho reservado.

Vamos analisar o processo de selecção de capacidade num caminho alternativo. Existem dois limiares associados com a profundidade de selecção do caminho alternativo. Estes limiares permitem um uso controlado do caminho associado. Enquanto um indica se as chamadas para  $TS_k$  podem ser encaminhadas usando caminhos alternativos com carga pesada, o outro indica se podem ser encaminhadas usando caminhos alternativos reservados, sendo eles  $Dhl_k$  e  $Dr_k$  respectivamente.

Se  $CIP_k < Dhl_k \times VTtraf_k$  então a chamada pode usar um caminho com carga pesada.

Se  $CIP_k < Dr_k \times VTtraf_k$  então a chamada pode usar caminhos reservados.

Os factores de profundidade  $Dhl_k$ ,  $Dr_k$  são controlados pelos limiares de bloqueio ponto a ponto  $NN_k$  e de bloqueio total para a central  $j$ ,  $TO_j$ , pela existência de capacidade no ramo directo e pelo nível de tráfego oferecido ponto a ponto, como ilustrado na tabela seguinte:

Item	Condições do tráfego				Factor de profundidade	
	$VTtraf_k$ (VTs)	Cir. Dir. (VTs)	Bloq. Tot. da Central ( $TO_j$ )	Bloq. Ponto a Ponto ( $NN_k$ )	$Dhl_k$	$Dr_k$
A	>15	>0	[0, .03]	[0, .01] (.01, 1]	0 1	0 0
B	$\leq 15$	>0	[0, .03]	[0, .01] (.01, 1]	1 1	0 1
C	qualquer	>0	(.03, .1]	[0, .5] (.5, 1]	0 1	0 0
D	qualquer	>0	(.1, 1]	[0, .5] (.5, 1]	0 0	0 0
E	qualquer	=0	[0, .03]	[0, 1]	8	8
F	qualquer	=0	(.03, 1]	[0, 1]	8	0

Tab. 3-5 – Profundidade de selecção do caminho. Adaptada de [Ash93].

Em geral, a profundidade de selecção permitida é maior se existe bloqueio para  $TS_k$ , dado que quantas mais escolhas de caminhos alternativos mais se reduz o bloqueio para  $TS_k$ . Esta grande profundidade de selecção é inibida se o bloqueio total alcançar um nível alto, indicando que existe uma condição de transbordo geral. Nesse caso é mais vantajoso reduzir o encaminhamento alternativo. Também, se não existir ramo directo, ou o tráfego ponto a ponto for pequeno, a profundidade de selecção deve então ser aumentada dado que a reserva de circuitos se tornará ineficiente ou mesmo impossível, e é necessária grande dependência nos caminhos alternativos para satisfazer os objectivos de bloqueio da rede.

Estudos simulacionais têm mostrado que os valores dos parâmetros, ilustrados na tabela anterior, oferecem uma boa alocação da capacidade disponível para pares de centrais com bloqueio [Ash93].

Uma vez ter sido determinada a sequência de caminhos, a central origem faz um pedido à central destino para que esta lhe envie os *bit maps* do estado de carga. Uma vez recebidos os *bit maps*, da central destino, a central origem faz a função *AND* sobre a lista dos *bit maps* para determinar os caminhos disponíveis. A central origem identifica os caminhos com o estado de carga mais leve. O próximo passo é calcular a soma, de acordo com a tabela seguinte, do estado de ambos os feixes no caminho, para cada um dos caminhos identificados.

Estado	Valor
Ocupado	0
Reservado	1
HL	2
LL3	3
LL2	4
LL1	5

Tab. 3-6 – Valor correspondente a cada estado. Adaptada de [Ash93].

O caminho com o maior valor para a soma, que é o caminho com a capacidade livre máxima, é então escolhido. Se existirem vários caminhos com o mesmo máximo valor para a soma, o caminho a ser escolhido é o próximo caminho em rotação circular a partir do caminho previamente seleccionado numa ligação dessa origem para esse destino. Vai-se de seguida apresentar um exemplo para ilustrar a operação dos *bit maps* para seleccionar um caminho alternativo de dois feixes.

Considere a figura seguinte:

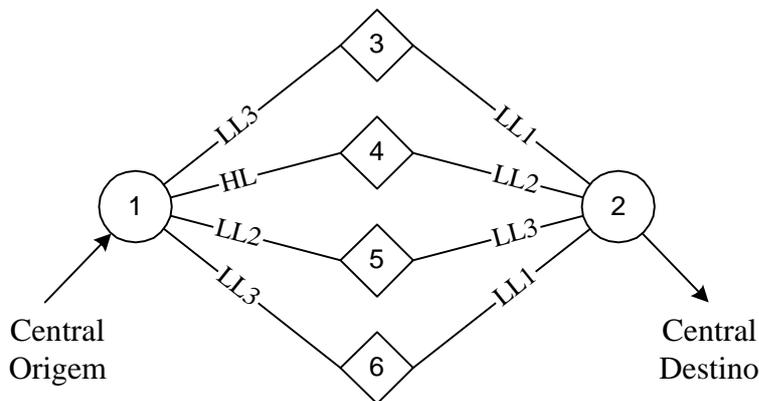


Fig. 3-6 – Exemplo de selecção de um caminho alternativo no RTNR. Adaptada [Ash98].

Considerando que a rede tem apenas as seis centrais representadas na figura, os *bit maps* da central origem seriam:

	1	2	3	4	5	6
LL1	0	0	0	0	0	0
LL2	0	0	0	0	1	0
LL3	0	0	1	0	1	1
HL	0	0	1	1	1	1
Reservado	0	0	1	1	1	1

Fig. 3-7 – *Bit maps* da central origem.

E os *bit maps* da central destino seriam:

	1	2	3	4	5	6
LL1	0	0	1	0	0	1
LL2	0	0	1	1	0	1
LL3	0	0	1	1	1	1
HL	0	0	1	1	1	1
Reservado	0	0	1	1	1	1

Fig. 3-8 – *Bit maps* da central destino.

Note que em cada *bit map* uma entrada a "1" significa que o feixe que liga a central correspondente a essa entrada à central possuidora desses *bit maps* tem carga igual ou menor que o estado de carga representado por esse *bit map*. Por exemplo o *bit map* LL3 da central origem tem três entradas a "1". As entradas correspondentes à central 3 e à central 6 estão a "1" porque o feixe da central origem para a central 3 e o feixe da central origem para a central 6 estão no estado de carga LL3, enquanto que a entrada a "1" correspondente à estação cinco é porque o feixe da central origem para a central cinco está no estado de carga LL2.

Ao efectuar a conjunção lógica (*AND*) dos *bit maps* correspondentes o resultado seria:

	1	2	3	4	5	6
LL1	0	0	0	0	0	0
LL2	0	0	0	0	0	0
LL3	0	0	1	0	1	1
HL	0	0	1	1	1	1
Reservado	0	0	1	1	1	1

Fig. 3-9 – *Bit maps* dos caminhos candidatos.

A central origem identifica os caminhos com estado de carga LL3 (caminhos candidatos com estado de carga mais leve). Esses caminhos são aqueles que usam a central 3, 5 ou 6 como central intermédia. Para saber qual destes é o que vai ser usado para estabelecer a chamada, vamos determinar a soma do estado dos feixes para cada um desses caminhos. De acordo com a tabela anterior vem que:

Central intermédia	Estado do 1º feixe		Estado do 2º feixe		Soma
3	3	+	5	=	8
5	4	+	3	=	7
6	3	+	5	=	8

Tab. 3-7 – Exemplo de selecção de um caminho. Adaptada de [Ash98].

Nesta tabela os números da segunda e da terceira coluna representam os estados conforme tabela 3-6. Verifica-se que o maior valor para a soma é obtido em 2 caminhos,. São eles o caminho alternativo com central intermédia 3 e o caminho alternativo com central intermédia 6. O caminho onde a

chamada deve ser estabelecida depende da última central intermédia que foi usada numa ligação desta origem para este destino. Supondo que a última central intermédia usada foi a central 4, a central que seria agora usada seria a 6. Se tivesse sido, por exemplo, a central 6 agora seria usada a central 3.

### Redução do atraso no estabelecimento de uma chamada

Quando é necessário um caminho de dois feixes para estabelecer a chamada, a central origem tem que enviar um pedido à central destino, esperar pela resposta ao pedido e analisar a resposta recebida antes de poder encaminhar a chamada. De modo a reduzir a quantidade de chamadas com este atraso, a central origem armazena a última informação de estado que recebeu dessa central destino (ou apenas a última central intermédia que foi utilizada numa ligação entre essa origem/destino). Enquanto espera a nova informação de estado, da central destino, tenta estabelecer a chamada com a informação que já possui. Apenas se a chamada não puder ser estabelecida com essa informação é que vai tentar estabelecer-se com a informação actualizada.

Neste método as centrais têm que ter a possibilidade de *crankback*, pois só assim é que uma chamada que tenha sido bloqueada no segundo feixe, do primeiro caminho alternativo tentado, pode tentar encaminhar-se de novo, mas agora com a informação de estado actualizada.

### Actualizações periódicas

Todos os 3 minutos é executado o processo de actualização para determinar o nível de bloqueio ponto a ponto,  $NN_k$ , o nível de reserva ponto a ponto,  $R_k$ , a carga de tráfego estimada ponto a ponto,  $TL_k$ , os circuitos virtuais necessários,  $VTraf_k$ , a sequência de caminhos ponto a ponto com os valores  $Dhl_k$  e  $Dr_k$  (limiares associados com a profundidade de selecção do caminho), e o nível de bloqueio total  $TO_j$ . A estimativa da carga de tráfego oferecido (em número de chamadas) no período  $n$  é dada por:

$$TL_k^n = 0.5 \times TL_k^{n-1} + 0.5 \times (CIP_k^n + OV_k) \quad (3-2)$$

Como já foi dito  $VTraf_k$  é estimado por:

$$VTraf_k = 1.1 \times TL_k \quad (3-3)$$

Nesta expressão o factor 1.1 é indicado como uma boa escolha através de análises simulacionais. É este o valor utilizado na implementação do RTNR na rede AT&T. [Ash93]

E  $R_k$ ,  $Dhl_k$  e  $Dr_k$  são determinados como indicado nas tabelas 3.1 e 3.5.

O bloqueio total para a central  $j$  obtém-se do seguinte modo:

$$TO_j = \sum_k OV_k / \sum_k PC_k \quad (3.4)$$

onde o somatório é para todas as centrais  $k$  para as quais esta central envia tráfego.

### 3.5.1.3.2 Encaminhamento por classe de serviço no RTNR

O RTNR oferece uma plataforma para implementar uma rede dinâmica de serviços integrados, e portanto pode ser usado para todos os serviços de voz e dados na rede de comutação. As várias classes de serviço partilham a largura de banda nos feixes da rede. A rede dinâmica de múltiplas classes de serviços é projectada para transportar a carga combinada prevista de chamadas para todas as classes de serviço. O controlo de encaminhamento atribui um certo número de circuitos directos entre  $OS_j$  e  $TS_k$  para cada classe de serviço. Esse número é a largura de banda mínima garantida para a classe de serviço, mas se a classe de serviço tem o seu objectivo de bloqueio satisfeito, as outras classes de serviço podem partilhar essa largura de banda. A soma do número de circuitos directos entre  $OS_j$  e  $TS_k$  para cada classe de serviço é igual à largura de banda total no feixe. Esta é uma questão específica do método para redes com integração de serviços.

Define-se uma rede virtual  $i$  associada a cada classe de serviço  $i$ . Cada uma destas redes virtuais é tratada de forma análoga a uma rede com apenas uma classe de serviço. As diferenças estão apenas no modo como se determinam os limiares do estado de carga e na existência, agora, de uma largura de banda reservada mínima no feixe de  $OS_j$  para  $TS_k$  para a rede virtual  $i$ . Por isto vai explicar-se, de forma geral, como funciona o método para redes com integração de serviços, só entrando em pormenor nas questões específicas para estas redes.

Também neste caso a central origem tenta primeiro completar a chamada no ramo directo, se existir. Se o circuito directo não estiver disponível, a central origem tenta encontrar um caminho de dois feixes. Vão-se considerar seis estados de carga para reflectir o estado de carga dos feixes, como fizemos no caso de apenas um serviço.

Cada central origem  $OS_j$  mantém para cada central destino  $TS_k$  também quatro quantidades, como no caso monoclasse, mas agora por cada classe de serviço  $i$ . Essas quantidades são usadas para determinar o limiar do estado de carga para o feixe de  $OS_j$  para  $TS_k$  para a rede virtual  $i$ . São elas  $CIP_k^i$ ,  $NN_k^i$ ,  $TL_k^i(n)$  e  $VTraf_k^i$  definidas como no caso monoclasse mas agora referentes à classe de serviço particular  $i$ .

O nível de reserva calculado para cada feixe é também agora determinado por classe de serviço, sendo designado por  $R_k^i$ . A determinação do  $R_k^i$  é baseada no nível de bloqueio ponto a ponto  $NN_k^i$  na rede

virtual  $i$ , de modo análogo ao utilizado no caso monoclasa (ver tabela 3.1). Se ocorrer bloqueio ponto a ponto para uma rede virtual, no intervalo de actualização, a reserva de circuitos é accionada e o nível de reserva  $R_k^i$  é activado para a rede virtual  $i$  de acordo com o bloqueio ponto a ponto. Deste modo, o tráfego que está a tentar ser encaminhado num caminho alternativo que contenha o ramo directo que liga o par de centrais com bloqueio, está sujeito a reserva de circuitos, de modo a favorecer o tráfego directo entre essas centrais.

O número de circuitos virtuais reservados para a rede virtual  $i$ ,  $Rtraf_k^i$  é:

$$Rtraf_k^i = n^i + \min[R_k^i, \max(0, VTtraf_k^i - CIP_k^i)] \quad (3.5)$$

onde  $n^i$  é o número de circuitos virtuais continuamente reservados para a rede virtual  $i$ .

Sendo a largura de banda total reservada  $RBWtraf_k$  ( $RBWtraf_k$  total **R**eserved **B**and**W**idth capacity) nos feixes da rede integrada dada por:

$$RBWtraf_k = \sum_{i=VN_1}^{VN_N} Rtraf_k^i \times r_i \quad (3.6)$$

O somatório vai da classe de serviço de menor necessidade de largura de banda até à classe de serviço de maior necessidade de largura de banda. Para permitir maior oportunidade ao tráfego que necessite de maior largura de banda deve alterar-se o somatório anterior para abranger apenas os valores que vão desde a classe de serviço que se está a considerar, na determinação do estado de carga do feixe, até à classe de serviço existente que necessite de maior largura de banda.

Os valores dos limiares dos circuitos virtuais para os diferentes estados estão indicados na tabela seguinte:

Nome do limiar	Nível do limiar (VTs)
$TK1_k$	$.05 \times VTtraf_k^i$
$TK2_k$	$.1 \times VTtraf_k^i$
$TK3_k$	$.2 \times VTtraf_k^i$

Tab. 3-8 – Limiares do estado de carga dos feixes para a rede virtual  $i$ . Adaptada de [Ash98].

Com estes limiares, com a largura de banda livre no feixe para  $TS_k$ ,  $ILBW_k$  ( $ILBW_k$  **I**dle **L**ink **B**and**W**idth) e com a largura de banda total reservada  $RBWtraf_k$  define-se o estado de carga do feixe como ilustrado na tabela seguinte

Nível de largura de banda livre no feixe	Estado de carga
$0 \leq ILBW_k < r^i$	Ocupado
$r^i \leq ILBW_k \leq RBWtraf_k$	Reservado
$RBWtraf_k < ILBW_k \leq TK1_k \times r^i + RBWtraf_k$	HL
$TK1_k \times r^i + RBWtraf_k < ILBW_k \leq TK2_k \times r^i + RBWtraf_k$	LL3
$TK2_k \times r^i + RBWtraf_k < ILBW_k \leq TK3_k \times r^i + RBWtraf_k$	LL2
$TK3_k \times r^i + RBWtraf_k < ILBW_k$ e $ILBW_k > r_i$	LL1

Tab. 3-9 – Definição do estado de carga dos feixes para a rede virtual  $i$ . Adaptada de [Ash98].

Como  $VTtraf_k^i$  depende da rede virtual, o estado de carga de um feixe também depende da rede virtual.

### Seleção de capacidade no feixe directo

A largura de banda mínima no feixe de  $OS_j$  para  $TS_k$  para a rede virtual  $i$  designada por  $VTeng_k^i$  é a largura de banda mínima garantida para a rede virtual  $i$  quando existe bloqueio. Se a rede virtual  $i$  tem o seu objectivo de bloqueio satisfeito, as outras redes virtuais são livres de partilhar a largura de banda alocada  $VTeng_k^i$  para essa rede virtual. O processo de partilha do ramo directo pelas várias classes de serviços é implementado do seguinte modo:

- Se as chamadas em progresso para essa rede virtual  $i$  ( $CIP_k^i$ ) estiverem abaixo do nível  $VTeng_k^i$  um serviço (chamada) nessa rede virtual vai tentar ocupar sempre um circuito virtual no ramo directo.
- Se  $CIP_k^i$  é igual ou maior do que  $VTeng_k^i$  a chamada pode ocupar um circuito virtual no ramo directo apenas enquanto a largura de banda livre no feixe seja superior à largura de banda reservada pelas outras redes virtuais que não têm o seu objectivo de bloqueio satisfeito. Isto é se:

$$CIP_k^i \geq VTeng_k^i \quad (3.7)$$

e

$$ILBW_k \geq r^i + RBWeng_k \quad (3.8)$$

onde

$$RBWeng_k = \sum_{i=VN_1}^{VN_N} \left\{ r^i + \min \left[ R_k^i, \max \left( 0, VTeng_k^i - CIP_k^i \right) \right] \right\} \times r^i \quad (3.9)$$

então pode seleccionar-se um circuito virtual no ramo directo. A segunda expressão significa que uma chamada, na rede virtual  $i$ , que está acima da sua capacidade projectada pode ser encaminhada no ramo directo se existir capacidade em excesso, isto é, além do que está reservado para as outras redes virtuais. A largura de banda reservada no ramo directo para qualquer rede virtual é quando muito  $VTeng_k^i - CIP_k^i$ , e esta capacidade é reservada apenas se a reserva for accionada para  $TS_k$  para a rede virtual  $i$ . Assim que o número de chamadas em progresso alcance o nível  $VTeng_k^i$  as chamadas só podem ser encaminhadas no ramo directo quando a largura de banda não estiver reservada para o encaminhamento directo de outras redes virtuais, as quais podem estar abaixo dos seus valores  $VTeng_k^i$  e nas quais ocorra bloqueio ponto a ponto. Aqui o somatório também deve abranger apenas os valores que vão desde a classe de serviço seguinte à que se está a considerar, até à classe de serviço existente que necessite de maior largura de banda. Sendo a classe de serviço que se está a considerar a correspondente à chamada que se está a tentar encaminhar no ramo directo.

As quantidades  $VTeng_k^i$  são escolhidas no processo de projecto da rede tendo o cuidado de a sua soma, para todos os serviços que partilham a largura de banda do ramo directo, ser igual à largura de banda total do feixe.

Se não há largura de banda para a chamada no ramo directo, a chamada vai tentar encaminhar-se num caminho alternativo de dois feixes.

O processo de selecção de capacidade num caminho alternativo é executado de modo análogo ao utilizado no caso de apenas uma classe de serviço. Aqui a profundidade de selecção do caminho é determinada, para cada rede virtual, do seguinte modo:

Se  $CIP_k^i < Dhl_k^i \times VTtraf_k^i$  então a chamada pode usar um caminho com carga pesada.

Se  $CIP_k^i < Dr_k^i \times VTtraf_k^i$  então a chamada pode usar caminhos reservados.

Sendo  $Dhl_k^i$  (associado com a selecção de um caminho alternativo com carga HL na rede virtual  $i$ ) e  $Dr_k^i$  (associado com a selecção de um caminho alternativo reservado na rede virtual  $i$ ) os factores de profundidade de selecção do caminho alternativo.

Item	Condições do tráfego				Factor de profundidade	
	$VTtraf_k^i$ (VTs)	$VTeng_k^i$ (VTs)	Bloq. Tot. da Central ( $TO_j^i$ )	Bloq. Ponto a Ponto ( $NN_k^i$ )	$Dhl_k^i$	$Dr_k^i$
A	>15	>0	[0, .03]	[0, .01] (.01, 1]	0 1	0 0
B	≤ 15	>0	[0, .03]	[0, .01] (.01, 1]	1 1	0 1
C	qualquer	>0	(.03, .1]	[0, .5] (.5, 1]	0 1	0 0
D	qualquer	>0	(.1, 1]	[0, .5] (.5, 1]	0 0	0 0
E	qualquer	=0	[0, .03]	[0, 1]	8	8
F	qualquer	=0	(.03, 1]	[0, 1]	8	0

Tab. 3-10 – Profundidade de selecção do caminho por classe de serviço. Adaptada de [Ash98].

Resumindo, nas condições normais de funcionamento da rede, a capacidade disponível é partilhada completamente por todos os serviços. Quando ocorre bloqueio na rede virtual  $i$ , a reserva de circuitos actua para proibir que o tráfego encaminhado alternativamente e o tráfego de outras redes virtuais ocupem capacidade do ramo directo projectada para o serviço  $i$ . Isto é, se existe bloqueio na rede virtual  $i$ , é proibido encaminhar tráfego de outras redes virtuais no ramo directo para além do  $VTeng_k^i$  para elas definido, enquanto as chamadas em progresso para a rede virtual  $i$  estejam abaixo do nível  $VTeng_k^i$  e é proibido também encaminhar tráfego alternativamente de todas as redes virtuais no ramo directo enquanto as chamadas em progresso para a rede virtual  $i$  estejam abaixo de  $VTtraf_k^i$ . Se as chamadas em progresso para a rede virtual  $i$  excederem  $VTeng_k^i$ , a capacidade alocada já está a ser usada e outros serviços são livres de usar a capacidade do ramo directo. Se as chamadas em progresso para a rede virtual  $i$  excederem  $VTtraf_k^i$ , já não é necessária qualquer reserva para atingir o objectivo do grau de serviço, e todos os circuitos directos podem ser partilhados por todo o tráfego. Também aqui é executado o processo de actualização em cada período para determinar o nível de bloqueio ponto a ponto  $NN_k^i$ , nível de reserva ponto a ponto  $R_k^i$ , carga de tráfego estimada ponto a ponto  $TL_k^i(n)$ , os circuitos virtuais necessários ponto a ponto  $VTtraf_k^i$ , os valores de  $Dhl_k^i$  e de  $Dr_k^i$ , e o nível de bloqueio total.

A estimativa da carga de tráfego oferecida no período  $n$  é dada por:

$$TL_k^i(n) = 0.5 \times TL_k^i(n-1) + 0.5 \times (CIP_k^i(n) + OV_k^i) \quad (3-10)$$

E  $VTtraf_k^i$  é estimado por:

$$VTtraf_k^i = 1.1 \times TL_k^i(n) \quad (3-11)$$

Todas as quantidades anteriores são determinadas, para cada rede virtual, pela central  $OS_j$  para a central  $TS_k$  segundo Ash [Ash91] e [Ash93]; no entanto no livro [Ash98] parece indicar que estes valores são calculados para pares de centrais. Sendo assim haveria necessidade de troca de outra informação entre as centrais além dos referidos *bit maps*, o que não é indicado e que acarretaria maior carga nas comunicações entre as centrais através da rede CCS. Deve ser dito, no entanto, que se estas quantidades fossem calculadas para pares de centrais (feixes), por exemplo, a reserva de circuitos funcionaria de modo justo, o que não acontece se assim não for. Pode ocorrer por exemplo que de  $OS_j$  para  $TS_k$  esteja um determinado número de circuitos reservado para tráfego directo e no sentido inverso não haja qualquer necessidade de reserva de circuitos, o que permite então gastar todos os circuitos (pois não existe reserva) enquanto que eles deviam ficar reservados para o tráfego directo de  $OS_j$  para  $TS_k$ .

**Notação para o RTNR monoclasse**

$OS_j$  - central origem ( $OS_j$  **O**riginating **S**witch)

$TS_k$  - central destino ( $TS_k$  **T**erminating **S**witch)

VT – circuito virtual (VT **V**irtual **T**runk)

NSN – número da central na rede (NSN **N**etwork **S**witch **N**umber)

$CIP_k$  - número de chamadas em progresso de  $OS_j$  para  $TS_k$  ( $CIP_k$  **C**alls **I**n **P**rogress)

$NN_k$  - bloqueio ponto a ponto da central  $OS_j$  para a central  $TS_k$  ( $NN_k$  **N**ode-to-**N**ode blocking)

$OV_k$  - contador de chamadas transbordadas de  $OS_j$  para  $TS_k$  ( $OV_k$  **O**verflow count)

$PC_k$  - contador de chamadas oferecidas de  $OS_j$  para  $TS_k$  ( $PC_k$  **P**eg **C**ount)

$TL_k$  - carga de tráfego oferecida (em número de chamadas) à central  $TS_k$  pela central  $OS_j$  ( $TL_k$  **O**ffered **T**raffic **L**oad)

$VTraf_k$  - número de circuitos virtuais necessários para satisfazer o grau de serviço pretendido para a carga de tráfego oferecida corrente  $TL_k$  de  $OS_j$  para  $TS_k$  ( $VTraf_k$  number of **V**irtual **T**runks)

$R_k$  - nível de reserva- ( $R_k$  **R**eservation level)

$Rtraf_k$  - circuitos virtuais reservados ( $Rtraf_k$  **R**eserved virtual trunks)

$ILVT_k$  - número de circuitos virtuais livres de  $OS_j$  para  $TS_k$  ( $ILVT_k$  **I**dLe **V**irtual **T**runks)

LL1 – com carga leve máxima (LL1 maximum **L**ightly **L**oaded)

LL2 – com carga leve média (LL2 medium **L**ightly **L**oaded)

LL3 – com carga leve mínima (LL3 minimum **L**ightly **L**oaded)

HL – com carga pesada (HL **H**eavily **L**oaded)

R – Reservado (R **R**eserved)

B – Ocupado (B **B**usy)

$Dhl_k$  - limiar associado com a selecção de um caminho alternativo com carga HL

$Dr_k$  - limiar associado com a selecção de um caminho alternativo reservado

$TO_j$  - bloqueio total para a central  $j$  ( $TO_j$  **T**otal **O**ffice blocking)

### Notação para o RTNR multiclasse

$OS_j$  - central origem ( $OS_j$  **O**riginating **S**witch)

$TS_k$  - central destino ( $TS_k$  **T**erminating **S**witch)

$CIP_k^i$  - número de chamadas em progresso de  $OS_j$  para  $TS_k$  na rede virtual  $i$  ( $CIP_k^i$  **C**alls **I**n **P**rogress)

$NN_k^i$  - bloqueio ponto a ponto da central  $OS_j$  para a central  $TS_k$  na rede virtual  $i$  ( $NN_k^i$  **N**ode-to-**N**ode blocking)

$TL_k^i(n)$  - carga de tráfego oferecida à central  $TS_k$  pela central  $OS_j$  no período  $n$  pela classe de serviço  $i$  ( $TL_k^i(n)$  offered **T**raffic **L**oad)

$VTraf_k^i$  - número de circuitos virtuais necessários para satisfazer o grau de serviço pretendido para a carga de tráfego oferecida corrente  $TL_k^i(n)$  de  $OS_j$  para  $TS_k$  na rede virtual  $i$  ( $VTraf_k^i$  number of **V**irtual **T**runks)

$R_k^i$  - nível de reserva para a rede virtual  $i$  ( $R_k^i$  **R**eservation level)

$Rtraf_k^i$  - circuitos virtuais reservados para a rede virtual  $i$  ( $Rtraf_k^i$  **R**eserved virtual trunks)

$RBWtraf_k$  - largura de banda total reservada de  $OS_j$  para  $TS_k$  ( $RBWtraf_k$  total **R**eserved **B**and**W**idth capacity)

$ILBW_k$  - largura de banda livre do feixe  $OS_j$  para  $TS_k$  ( $ILBW_k$  **I**dle **L**ink **B**and**W**idth)

$VTeng_k^i$  - largura de banda mínima no feixe de  $OS_j$  para  $TS_k$  para a rede virtual  $i$

$Dhl_k^i$  - limiar associado com a selecção de um caminho alternativo com carga HL na rede virtual  $i$

$Dr_k^i$  - limiar associado com a selecção de um caminho alternativo reservado na rede virtual  $i$

$TO_j^i$  - bloqueio total para a central  $j$  na rede virtual  $i$  ( $TO_j^i$  **T**otal **O**ffice blocking)

### 3.5.1.4 Encaminhamento Controlado Dinamicamente (“Dynamically Controlled Routing” ou “Dynamic Call Routing” - DCR)

O DCR (descrito em [Girard90] e [Watanabe90]), foi inicialmente designado por High Performance Routing (HPR), foi desenvolvido no Canadá, por Bell-Northern Research para ser usado nas redes da Telecom Canada. Usa um processador central para encontrar o melhor caminho alternativo. Para tomar as decisões de encaminhamento usa a informação da ocupação em todos os grupos de circuitos da rede, informação essa actualizada periodicamente através de medidas obtidas pelas centrais. Neste método existe uma transmissão de dados periódica das centrais inteligentes para um computador central o qual calcula as recomendações e as envia de volta para as centrais.

Este método opera enviando as chamadas pelos caminhos com o número esperado de circuitos livres mais alto e trabalha com um caminho de transbordo único calculado periodicamente para cada par origem-destino. Cada chamada é primeiro oferecida ao caminho directo (assume-se que existe um ramo directo entre quaisquer pares de centrais que tenham tráfego entre si). Se a chamada é bloqueada, é oferecida a um caminho alternativo de dois feixes, onde elas são perdidas se bloqueadas de novo. O caminho alternativo é seleccionado aleatoriamente, com probabilidade de selecção proporcional à capacidade residual estimada de todos os caminhos potenciais. Qualquer central, excepto a origem e o destino, pode ser central de trânsito.

O método utiliza o mecanismo de reserva de circuitos para contornar situações de sobrecarga de tráfego. O número de circuitos reservados pode ter valores distintos nos vários feixes e pode ser fixo ou variável.

O DCR foi o primeiro método de encaminhamento, do tipo RCAR (*Residual Capacity Adaptive Routing*), implementado numa rede telefónica pública.

No DCR todos os  $\Delta$  segundos (intervalo de actualização), o centro de controlo da rede pergunta às centrais qual a ocupação dos seus feixes de saída. A informação é processada, e são enviadas novas tabelas de encaminhamento para as centrais para serem usadas nos  $\Delta$  segundos seguintes. Como já foi referido anteriormente, o caminho alternativo para qualquer par ordenado de centrais é único. Vamos ver de seguida como é feita a determinação desse caminho.

Seja  $\mu_s$  o tempo de ocupação médio por chamada no feixe  $s$  (na prática o seu valor é considerado independente de  $s$ ) e sejam  $N_s$ ,  $n_s(t)$  e  $m_s(t)$  o número total de circuitos, o número de circuitos ocupados no instante  $t$  e o número de circuitos reservados no instante  $t$  num feixe  $s$  respectivamente. A estimativa da capacidade residual no feixe  $s$  (número médio de circuitos livres esperado no feixe  $s$ ) no instante  $t + \tau$  é dada por:

$$R_s(t + \tau) = N_s - n_s(t) - \tau[A_s(t) - n_s(t)/\mu_s] - m_s(t) \quad (3.12)$$

Pois o número de circuitos livres no instante  $t$  é igual a  $N_s - n_s(t) - m_s(t)$ , o número médio (estimado) de chegadas em  $]t, t + \tau]$  é igual a  $\tau A_s(t)$  e o número médio (estimado) de terminações em  $]t, t + \tau]$  é igual a  $\tau n_s(t)/\mu_s$  (sendo  $1/\mu_s$  a taxa de terminação por chamada em curso).

Nesta fórmula,  $A_s(t)$  é a intensidade de chegadas de chamadas estimada no feixe  $s$  no instante  $t$  sendo a actualização nesse instante feita por:

$$A_s(t) = \theta A_s(t - \Delta) + (1 - \theta) \frac{M_s(\Delta)}{\Delta} \quad (3.13)$$

O valor actualizado de  $A_s(t)$  é obtido a partir do seu valor no último intervalo de actualização,  $A_s(t - \Delta)$ , e do número de chamadas oferecidas ao feixe  $s$ ,  $M_s(\Delta)$ , durante o último intervalo de tempo  $\Delta$ . O valor de  $M_s(\Delta)$  vai sendo incrementado à medida que são oferecidas chamadas ao feixe  $s$  e é reinicializado a zero no início de cada intervalo de actualização. O coeficiente  $\theta$  é usado na determinação da intensidade de chegada actualizada, de modo a dar mais peso ao seu valor no último intervalo de actualização, no caso de  $\theta > 0.5$ , ou ao número de chamadas oferecidas ao feixe  $s$  durante o último intervalo de tempo  $\Delta$ , no caso de  $\theta < 0.5$ .

A selecção de uma central intermédia é feita aleatoriamente. A probabilidade de selecção da central intermédia  $k$  para uma chamada entre  $i$  e  $j$ ,  $\alpha_k^{i,j}$ , é determinada como segue:

$$\alpha_k^{i,j} = \frac{\bar{\alpha}_k^{i,j}}{\sum_k \bar{\alpha}_k^{i,j}}, \quad (3.14)$$

onde

$$\bar{\alpha}_k^{i,j} = \max_k [0, \min\{R_{i,k}, R_{k,j}\}] \quad (3.15)$$

Os valores das estimativas das capacidades residuais,  $R_{i,k}$  e  $R_{k,j}$  usados são os obtidos no instante de extrapolação  $t + \tau$ .

O desempenho do algoritmo pode ser influenciado pelos seguintes parâmetros:

$\tau$  - intervalo de extrapolação.

$\Delta$  - intervalo de actualização.

$\theta$  - peso usado para determinação da intensidade de chegada actualizada

$m_s(t)$  – reserva de circuitos no feixe  $s$ . Pode tomar um valor fixo ou ser variável.

Ao fazer a simulação devem-se encontrar valores apropriados para estes parâmetros, afim de conseguir um bom desempenho.

A partir do momento em que o nível de ocupação atinge  $m_s(t)$  as chamadas de transbordo nesse feixe serão bloqueadas. Pode-se tornar adaptativa esta reserva de circuitos usando a fórmula seguinte:

$$m_s(t) = g \times a'_s \quad (3.16)$$

onde  $a'_s$  é a quantidade corrente de tráfego, oferecido pela primeira vez, que transborda do feixe  $s$  e  $g$  é um factor de escala, sendo normalmente usado o valor 1. A justificação da fórmula, para o cálculo da reserva adaptativa, pode ser encontrada em [Cameron86] como indicado por [Girard90].

### Notação para o DCR

$N_s$  - número de circuitos no feixe  $s$ .

$R_s(t)$  – número esperado de circuitos livres no feixe  $s$  no instante  $t$ .

$n_s(t)$  – número de circuitos ocupados no feixe  $s$  no instante  $t$ .

$A_s(t)$  – intensidade estimada de chegada de chamadas no feixe  $s$  no instante  $t$ .

$\mu_s$  – tempo de ocupação médio por chamada no feixe  $s$ .

$m_s(t)$  – reserva de circuitos no instante  $t$  no feixe  $s$ .

$M_s(\tau)$  – número de chamadas oferecidas actualmente ao feixe  $s$  durante um intervalo de tempo  $\tau$ .

$\alpha_k^{i,j}$  - probabilidade de selecção da central intermédia  $k$  para uma chamada entre  $i$  e  $j$ .

$\Delta$  - intervalo de actualização.

$\tau$  - intervalo de extrapolação.

$\theta$  - peso usado para determinação da intensidade de chegada actualizada.

### 3.5.1.5 Sistema para Testar o Encaminhamento Adaptativo (“System to Test Adaptative Routing” - STAR)

STAR é um projecto desenvolvido pelo Centro Nacional de Estudos de Telecomunicações (CNET), o centro de pesquisa da Telecom francesa, para avaliar o uso do encaminhamento adaptativo na rede francesa [Watanabe90]. Semelhante ao DCR, os dados de ocupação do feixe são periodicamente armazenados num processador central. A capacidade residual de um caminho é definida como a capacidade residual mínima de todos os grupos de circuitos pertencentes ao caminho. A sequência de selecção do caminho é reordenada por ordem decrescente da capacidade residual do caminho. Novas

chamadas são oferecidas, em primeiro lugar, ao caminho directo. Se este estiver bloqueado, as chamadas transbordam para os caminhos alternativos.

### 3.5.1.6 Gestão Dinâmica de tráfego (“Dynamic Traffic Management” – DTM)

O método gestão dinâmica de tráfego tem mostrado ser um meio de reduzir o bloqueio do tráfego oferecido em condições de sobrecarga da rede normais e anormais, entre outras coisas [Régnier90].

Nesta estratégia as funções de encaminhamento e gestão do tráfego são combinadas numa única função.

#### Arquitectura do Sistema e Fluxo de Dados

No método de encaminhamento DTM (“Dynamic Traffic Management”) (descrito em [Régnier90]) usa-se informação global da rede, obtida por todas as centrais da rede.

Existe um processador central que coleciona as medidas de tráfego enviadas pelas centrais e devolve a estas recomendações.

A colecção dos dados, aplicação e selecção do controlo são completamente automatizadas dentro de um ciclo de tempo fixo referido como ciclo de actualização, com valor recomendado de 10s.

Cada central  $i$  envia as seguintes medidas de tráfego ao processador:

$I_{ij}$  - número de circuitos livres no feixe para a central  $j$ , para todas as centrais  $i$  na rede DTM;

$CPU_i$  - a ocupação do CPU da central  $i$ ;

$O_{ij}$  - Uma medida do tráfego enviado por  $i$  para  $j$  e que transbordou do caminho directo.

Estas medidas são usadas no encaminhamento e controlo da congestão. Elas permitem ao processador obter um conhecimento global da capacidade livre na rede conseguindo assim definir o encaminhamento, permitindo também, nas decisões de controlo, proteger elementos sobrecarregados conseguindo o controlo da congestão.

Correspondendo às funções de encaminhamento e controlo da congestão, o processador envia dois tipos de controlos para cada central  $i$  :

$r_{ij}$  - para cada central  $j$ , é o nome de uma central através da qual  $i$  reencaminhará as chamadas para  $j$  quando o caminho directo está ocupado;  $r_{ij}$  pode também ser uma recomendação de bloqueio, e nesse caso a central  $i$  bloqueia a chamada para  $j$  se esta não pode ser encaminhada no caminho directo;

$IAT_{ij}$  - para cada recurso  $j$ , um intervalo de tempo mínimo entre chamadas admitidas para  $j$ .

### Encaminhamento

Considere-se que chega uma chamada à central  $i$  destinada à central  $j$ . No método de encaminhamento DTM, a central  $i$  tenta sempre primeiro encaminhar a chamada no ramo directo para  $j$ . Se o ramo directo está ocupado, a central  $i$  tenta encaminhar a chamada no caminho alternativo de dois feixes utilizando  $r_{ij}$  como central de trânsito, tal como foi recomendado pelo processador central. Este é o único caminho alternativo que a chamada pode utilizar. Se  $r_{ij}$  é uma recomendação de bloqueio ou se o feixe  $(i, r_{ij})$  está ocupado, a central  $i$  não tenta qualquer outro caminho, mas bloqueia a chamada imediatamente. A recomendação do caminho alternativo  $r_{ij}$  aplica-se só a chamadas originadas na central  $i$ . Chamadas que chegam a  $i$  de alguma outra central (chamadas de trânsito) para o qual  $i$  foi a central de trânsito recomendada, só podem tentar o caminho directo para  $j$ . Caso contrário, esta chamada utilizaria 3 ou mais feixes, o que seria ineficiente.

O processo de selecção de  $r_{ij}$  no processador central depende da existência ou não do ramo directo entre  $i$  e  $j$ . Se o ramo directo existe, que é o caso para a maior parte das chamadas,  $r_{ij}$  é determinado como sendo a central  $t$  que maximiza:

$$\text{Max}\{A_t, \min[I_{it} - PA_{it}, I_{tj} - PA_{tj}]\} \text{ e } t \neq i, j \quad (3.17)$$

se o valor máximo é positivo, caso contrário,  $r_{ij}$  é uma recomendação de bloqueio.  $I_{ij}$  já foi definido,  $PA_{ij}$  é uma *protective allowance* (funciona como reserva de circuitos) para o tráfego directo no feixe  $(i, j)$ . *Protective allowance* faz com que o tráfego que seria encaminhado no caminho alternativo não o seja quando o feixe está quase completamente ocupado. Isto reduz a possibilidade de o tráfego de transbordo causar um uso ineficiente dos recursos ocupando circuitos que provavelmente serão necessários para o tráfego directo. O valor de  $PA_{ij}$  pode ser calculado dinamicamente para obter uma

maior adaptação à carga de tráfego corrente. No entanto numa rede bem dimensionada a perda de desempenho pode ser pequena ao considerar um valor fixo [Régnier90].  $A_t$  é um parâmetro com valores no intervalo [0,1] que reflecte a disponibilidade da central  $t$ . É 1 se a central funciona normalmente, mas inferior quando a central  $t$  está sobrecarregada. Este é o modo de fazer com que os caminhos alternativos utilizando centrais de trânsito sobrecarregadas sejam menos atractivos e por isso com menor probabilidade de ser escolhidos pelo processador central. Vai ser ilustrado o processo de selecção de  $r_{ij}$  definido pela equação anterior utilizando a figura seguinte.

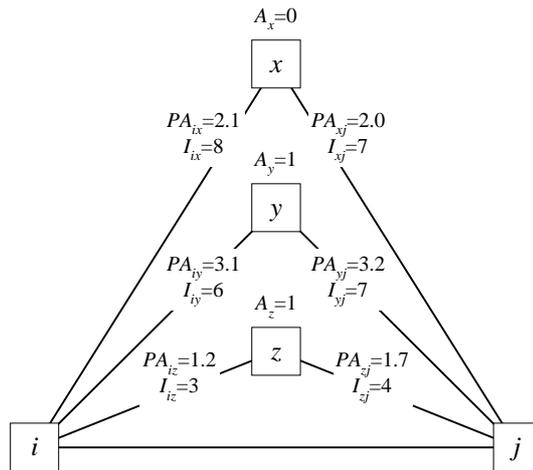


Fig. 3-10 - Selecção do caminho alternativo no método de encaminhamento DTM. Adaptada de [Régnier90].

Considerando a equação anterior, o caminho alternativo recomendado na figura é o caminho  $i - y - j$ . Embora o caminho  $i - x - j$  tenha mais capacidade livre não é recomendado porque a central  $x$  está sobrecarregada.

Se de  $i$  para  $j$  não existe um ramo directo,  $r_{ij}$  é determinado como sendo a central  $t$  que maximiza:

$$Max\{A_t, \min\{I_{it}, I_{tj}\}\} \text{ e } t \neq i, j \tag{3-18}$$

caso o valor máximo seja positivo.

Caso contrário,  $r_{ij}$  é uma recomendação de bloqueio. Esta equação é semelhante à equação anterior, excepto que as *protective allowances* não são consideradas. Isto significa que, se o tráfego de  $i$  para  $j$  não tem ramo directo, então este tráfego não dá prioridade ao tráfego directo nos feixes do seu caminho alternativo. Usar esta equação em vez da anterior faz com que o bloqueio para esse tráfego seja menor, mas ainda superior em média ao caso em que existe ramo directo. Como as parcelas de tráfego que não têm caminho directo são uma pequena fracção do tráfego total, este tratamento preferencial tem um impacto desprezável no bloqueio total.

O DTM toma as decisões de encaminhamento não numa base chamada após chamada, mas sim decisões para grupos de chamadas (nomeadamente numa base de 10 s).

### **Controlo da congestão**

Quando o processador declarar que um recurso está sob congestão, envia um controlo para todas as centrais para limitar o número de tentativas que eles podem deixar prosseguir na rede. Este controlo denotado  $IAT_{ij}$  para a central  $i$  e para o recurso  $j$ , consiste num intervalo de tempo mínimo. É implementado na central  $i$  aceitando uma chamada para  $j$  apenas se um temporizador exceder o valor de  $IAT_{ij}$ , colocando o temporizador a zero sempre que uma chamada para  $j$  seja aceite.

## **3.5.2 Métodos baseados nos processos de decisão de Markov**

Os estudos sobre o SDR (“State Dependent Routing”) começaram na Bellcore com o objectivo de construir uma teoria para responder à questão “Qual é a melhor regra de encaminhamento que pode ser aconselhada, assumindo informação de estado completa e actual”.

Krishnan et al tem proposto métodos de encaminhamento com base nas teorias dos processos de decisão de Markov. Nesses métodos de encaminhamento a escolha de um caminho para uma chamada é feita tendo como base os custos associados com os estados dos caminhos admissíveis na altura em que a chamada ocorre. Os custos são agora formulados no contexto de processos de decisão de Markov em vez dos valores dos caminhos usados nos métodos anteriores que eram definidos essencialmente pela capacidade residual do caminho. São esses os métodos que vão ser analisados de seguida.

### **3.5.2.1 Encaminhamento Dependente do Estado Separável (“Separable State Dependent Routing” - SDDR)**

**Base Teóricas para o SDDR** (descrito em [Krishnan89], [Krishnan90] e [Krishnan91])

Este método de encaminhamento trata o problema de encaminhamento como um processo de decisão de Markov. Assumindo independência estatística dos feixes da rede, o estado da rede pode ser definido simplesmente como o conjunto de ocupações dos feixes. Esta aproximação é equivalente a considerar que uma chamada estabelecida, que usa vários feixes, pode ser vista como várias chamadas independentes nesses feixes, com partidas independentes; isto é razoável sempre que o número de chamadas comuns aos dois feixes for uma pequena fracção do número total de chamadas estabelecidas em ambos os feixes – esta condição será verificada se existirem muitas chamadas encaminhadas directamente ou se cada feixe for usado por chamadas de pares de centrais muito diferentes.

Se  $n$  for o número de feixes, e  $x_i$  a ocupação do feixe  $i$ , então o estado da rede é dado pelo vector  $x = (x_1, x_2, \dots, x_n)$ . As transições de estado são muito simples:

A partida de uma chamada que estava a ocupar o feixe  $i$  muda o estado para  $(x_1, x_2, \dots, x_i - 1, \dots, x_n)$ .

A chegada de uma chamada pode produzir um novo estado. Se a chamada for transportada, é produzido novo estado, com a ocupação de cada feixe do caminho escolhido aumentada de uma unidade. Se a chamada for bloqueada o estado do sistema mantém-se inalterado.

Seja a regra de encaminhamento descrita por uma função  $R(x, k)$  que especifica o estado da rede obtido como resultado das decisões de encaminhamento quando uma chamada de um par de centrais  $k$  chega no estado  $x$ . A função  $V_R(x)$ , conhecida como função dos custos relativos de Howard [Howard60], permite-nos associar um custo a cada estado, de modo que  $[V_R(y) - V_R(x)]$  é a diferença entre o número médio total de chamadas perdidas na rede, usando a regra de encaminhamento  $R$ , partindo do estado inicial  $y$  ou partindo do estado inicial  $x$ .

A regra de encaminhamento SSDR óptimo é descrita como segue, em termos dos  $\{V_*(x)\}$  custos relativos óptimos correspondentes.

Quando uma chamada chega no estado  $x$ , tomemos o estado  $y$ , entre aqueles que podem ser alcançados por escolhas de caminhos admissíveis, com o menor custo. Depois:

$$\begin{cases} \text{Se } [V_*(y) - V_*(x)] < 1 & \text{escolher o caminho que leva ao estado } y \\ \text{Se } [V_*(y) - V_*(x)] \geq 1 & \text{então deve-se rejeitar a chamada} \end{cases} \quad (3.19)$$

A segunda parte da equação anterior é o mecanismo de controlo da congestão do SSDR, uma forma de reserva de circuitos, uma característica que, como se sabe, é essencial à estabilidade de qualquer método de encaminhamento não hierárquico.

A determinação dos custos relativos  $\{V(x)\}$  para o encaminhamento óptimo é uma tarefa sem solução para redes práticas, dada a dimensão do problema a resolver. No entanto, se se conseguirem determinar os custos  $\{V_R(x)\}$  para algum encaminhamento simples  $R$ , e se forem usados na equação anterior para construir um novo esquema  $R'$ , então é conhecido [Howard60] que  $R'$  é superior a  $R$  (a menos, claro, que  $R$  seja óptimo, e então  $R = R'$ ).

Logo, se conseguirmos determinar os custos relativos para o caso do encaminhamento directo, então a equação (3.19) pode ser usada para melhorar esse encaminhamento simples.

Em [Krishnan90] é apresentada uma tal expressão para a determinação dos custos relativos dos estados adjacentes  $(k + 1)$  e  $k$  para o caso do encaminhamento directo que é a seguinte:

$$D_k = \frac{B(s, \lambda)}{B(k, \lambda)}, \quad 0 \leq k < s \quad (3.20)$$

$D_k$  representa o número médio de chamadas perdidas adicionalmente, a longo prazo, quando o sistema parte do estado  $k + 1$  em vez do estado  $k$  ( $k \leq s - 1$ ), por virtude de se ter aceite uma nova chamada (chamada de teste).

Sendo  $s$  o número total de circuitos no feixe,  $\lambda$  a intensidade do fluxo de Poisson oferecida ao feixe e  $k$  o número de circuitos ocupados no feixe.  $B(s, \lambda)$  é a formula de Erlang-B para o bloqueio de um grupo de  $s$  circuitos ao qual é oferecido um tráfego de Poisson de  $\lambda$  Erlangs.

Krishnan, em [Krishnan90], diz que se pode conseguir um melhoramento no desempenho do esquema de encaminhamento se o esquema nominal de encaminhamento directo for substituído por um esquema nominal de encaminhamento não alternativo, do qual o encaminhamento directo é um caso especial. No esquema de encaminhamento não alternativo cada chamada que chega é oferecida precisamente a um caminho admissível, escolhido de acordo com uma distribuição probabilística sobre os seus caminhos admissíveis; se o caminho seleccionado está ocupado então nenhum outro caminho é tentado e a chamada é bloqueada. Dentro deste enquadramento de encaminhamento não alternativo, é obtido, a partir de um programa não linear, um conjunto óptimo de probabilidades de alocação do caminho para chamadas dos vários pares de centrais (para minimizar o bloqueio na rede), o qual requer conhecimento de todas as intensidades de tráfego, ponto a ponto, na rede. As cargas de tráfego obtidas  $y_k$ , resultantes deste esquema de encaminhamento nominal não alternativo, são tratadas como equivalentes às cargas nominais do encaminhamento directo, e são então vistas como cargas de Poisson oferecidas aos feixes, encaminhadas directamente e independentes; isto é equivalente a desacoplar a rede num conjunto de feixes independentes. Um passo no cálculo iterativo do encaminhamento é então obtido para este conjunto nominal de fluxos nos feixes com o uso da fórmula da equação (3.20) que vamos reescrever do seguinte modo:

$$\Delta(k, j) = \frac{B(s_k, y_k)}{B(j, y_k)}, \quad 0 \leq j < s_k \quad (3.21)$$

onde  $k$  é o feixe que tem  $s_k$  circuitos,  $j$  dos quais ocupados e uma carga nominal oferecida de Poisson de  $y_k$  erlangs.

Como os feixes são vistos como independentes, o custo de adicionar uma chamada que ocupe os feixes  $k_1, \dots, k_m$ , nos estados  $j_1, \dots, j_m$  respectivamente, é então dada por [Krishnan89]:

$$\text{custo do caminho} = \sum_{i=1}^m \Delta(k_i, j_i) \quad (3.22)$$

Então, o custo de um caminho é ‘separável’ nos custos dos feixes constituintes.

Em conclusão, o método de encaminhamento SDDR é obtido quando é usado um processo iterativo para melhorar um esquema nominal de encaminhamento directo. O processo iterativo é baseado no cálculo dos custos de Howard, de adicionar uma chamada a um caminho. Para o cálculo dos custos, as intensidades de tráfego para os feixes são obtidas de um esquema de encaminhamento fictício não alternativo independente do estado.

**O funcionamento do SDDR** (descrito em [Koussoulas93], [Krishnan90] e [Krishnan91])

O conjunto de caminhos permitidos (admissíveis), para um dado fluxo, contém o ramo directo (se existe) e um número de caminhos alternativos. Os caminhos são distinguidos uns dos outros através dos respectivos custos.

Quando uma chamada chega, para tomar a decisão de encaminhamento, é calculado o custo de cada caminho admissível que está livre (isto é, tem pelo menos um circuito livre em cada feixe do caminho), no estado corrente da rede, usando as fórmulas (3.21) e (3.22).

Depois dos custos de todos os caminhos permitidos (admissíveis) terem sido calculados, se o custo do caminho de custo mínimo iguala ou excede 1, então a chamada é rejeitada, mesmo quando haja circuitos disponíveis; caso contrário, a chamada é encaminhada no caminho de custo mínimo. De notar que uma chamada nunca será rejeitada se o conjunto de caminhos permitidos incluir o caminho directo e se este estiver livre (tiver pelo menos um circuito livre) pois analisando a fórmula da equação (3.20) ou a reescrita (3.21) verifica-se facilmente que o custo nesse caminho é sempre menor que 1, nessas condições .

As decisões de encaminhamento podem ser tomadas numa base, chamada após chamada, como proposto no método de encaminhamento original, ou com intervalos maiores. No primeiro caso são utilizadas medidas instantâneas enquanto que no segundo são utilizadas medidas das ocupações médias do último intervalo.

Na expressão de cálculo dos custos é utilizado o valor das cargas nominais oferecidas aos feixes num esquema de encaminhamento auxiliar [Krishnan90], descrito anteriormente. A reacção em tempo real às condições variáveis no tráfego não é possível numa implementação realista. Assim o intervalo de actualização dos valores das cargas nominais oferecidas aos feixes, deve ser meia hora ou superior.

A versão geral do algoritmo consiste, então, em duas fases que funcionam em escalas de tempo diferentes: A fase lenta, ou a longo prazo, toma como entrada a carga ponto a ponto estimada ou previsível e gera um conjunto de cargas oferecidas ao feixe que são ótimas para um esquema de encaminhamento muito simples. A fase rápida, ou a curto prazo, calcula o custo para cada caminho tomando em conta a carga nominal oferecida anterior e a ocupação mais recente dos feixes. Portanto a parte rápida reage às mudanças rápidas (tentativa de chamada), enquanto a parte lenta reage às mudanças lentas (padrões de carga). A grandeza destas escalas pode ser decidida pelo projectista ou imposta por restrições práticas.

Então, o SSDR usa estatísticas e medidas de tráfego para produzir as suas decisões de encaminhamento.

Uma característica interessante de SSDR é a capacidade de funcionar no modo distribuído ou centralizado, dando ampla flexibilidade para o projecto da arquitectura da rede.

Em [Koussoulas93] é apresentado o modelo matemático que pode ser usado para prever o desempenho macroscópico de uma rede em que o encaminhamento SSDR seja utilizado.

#### **3.5.2.1.1 Encaminhamento Dinâmico - 5 minutos (“Dynamic Routing 5 minutes” – DR-5)**

No método de encaminhamento SSDR original, a informação de estado está disponível instantaneamente na chegada de uma chamada. Na prática, nas redes metropolitanas dos Estados Unidos, a informação de ocupação do feixe é obtida apenas em intervalos de 5 minutos por muitas centrais. Aí, o método de encaminhamento SSDR foi adaptado para uma forma prática na qual os períodos de actualização da informação de estado são usados para definir os padrões de encaminhamento da rede que permanecem em uso até à próxima actualização. Esta adaptação tem sido designada por Dynamic Routing – 5 minutes (DR-5) (descrito em [Krishnan90]).

#### **3.5.2.2 Encaminhamento com previsão (“Forward Looking Routing” - FLR)**

Os autores do SSDR, Krishnan e Ott, constataram que este tem a tendência para o encaminhamento excessivo em caminhos de dois feixes [Krishnan89].

Estudos de investigação, levados a cabo pelos autores, resultaram no desenvolvimento de uma modificação do SSDR chamada FLR [Krishnan89], a qual, segundo eles, oferece melhorias significativas sobre o SSDR.

O FLR assim como o SSDR é um método de encaminhamento que tenta encaminhar as chamadas de modo a minimizar efeitos negativos sobre o desempenho das chamadas futuras (daqui vem o nome

FLR). Como vamos ver a implementação do método FLR não é mais difícil do que SSDR, necessita apenas do cálculo adicional, *off-line*, de um número modesto de parâmetros.

A fórmula (3.21), a qual equaciona o bloqueio no feixe com chamada perdida, não tem em consideração as várias oportunidades (vários caminhos admissíveis) que a chamada pode ter. O FLR tenta tomar em conta essas outras oportunidades das chamadas. Isto vai conduzir a que o custo de encaminhar uma chamada directamente seja menor, no FLR, do que o custo de encaminhar uma chamada directamente no SSDR.

A ideia foi considerar nas decisões de encaminhamento um par de centrais de cada vez. Quando se considera um par de centrais específico, assume-se que todos os outros pares de centrais estão restringidos a seguir um esquema de encaminhamento independente do estado, especificado para eles no esquema de encaminhamento nominal não alternativo. Ao par de centrais particular sob consideração é permitido seguir uma regra de encaminhamento dependente do estado.

Com este ponto de vista, quando consideramos, por exemplo, o encaminhamento de chamadas do par de centrais  $AB$ , todo o restante tráfego na rede é composto por fluxos de Poisson independentes encaminhados directamente (resultantes da aproximação feita na análise do encaminhamento nominal não alternativo). Por isso a rede, a ser examinada para as decisões de encaminhamento para o par de centrais  $AB$ , reduz-se aos feixes constituindo os caminhos admissíveis para o par de centrais  $AB$ . Nessa rede, as chamadas do par de centrais  $AB$  podem usar o ramo directo  $AB$  ou qualquer um dos caminhos de dois feixes  $AV_iB$ ,  $i = 1, \dots, n$ ; o restante tráfego consiste em fluxos de Poisson independentes os quais são restringidos a ser encaminhados directamente nos seus respectivos feixes. Por isso, as decisões de encaminhamento são necessárias apenas nessa subrede para as chamadas do par de centrais  $AB$ , como função dos estados dos feixes nos caminhos admissíveis para as chamadas  $AB$ . Mesmo para este problema reduzido, uma solução exacta é impraticável, pois o espaço de estados da subrede pode ser ainda muito grande.

Portanto, aqueles autores desenvolveram uma solução aproximada, fazendo uso do fluxo nos feixes do esquema de encaminhamento nominal não alternativo. Em particular, trabalhando com a carga oferecida aos feixes  $\{y_k\}$  determinada por esse esquema de encaminhamento, assume-se que sobre cada feixe  $k$ , todos os componentes do fluxo de  $y_k$ , com excepção do componente do fluxo  $AB$ , são restringidos a ser encaminhados directamente no feixe, enquanto que ao componente  $AB$  de  $y_k$  é também permitido considerar os outros caminhos  $AB$ , isto é, é permitida uma segunda oportunidade.

A fórmula (3.21) para os custos relativos dos estados adjacentes vai ser modificada tendo isto em consideração. Suponhamos que o par de centrais  $AB$  tenha sido numerado par de centrais  $m$ , e

supondo que, em cada feixe  $k$  na subrede dos caminhos  $AB$  uma fracção  $\phi_{mk}$  da carga  $y_k$  oferecida ao feixe pertence ao fluxo  $AB$  e a fracção restante  $(1 - \phi_{mk})$  é constituída por outros fluxos alocados ao feixe  $k$  sob o encaminhamento nominal não alternativo. De acordo com as suposições anteriores, veremos  $(1 - \phi_{mk})$  como a fracção de tráfego restringido a ser encaminhado directamente no feixe  $k$ , enquanto que  $\phi_{mk}$  é a fracção de tráfego que pode ser encaminhado em qualquer caminho admissível para o par de centrais  $m$ .

Então, considerando o carácter distinto das chamadas destes dois fluxos quando bloqueadas no feixe  $k$ , é usada a seguinte expressão para o custo modificado de colocar uma chamada do par de centrais  $m$  no feixe  $k$  quando  $j$  dos seus circuitos já estão ocupados [Krishnan89]:

$$\Delta_m(k, j) = \frac{B(s_k, y_k)}{B(j, y_k)} * [1 - \phi_{mk} + \phi_{mk} h_{mk}] \quad (3.23)$$

onde  $h_{mk}$  é por definição o produto dos bloqueios (sob o encaminhamento não alternativo) de todos os caminhos para o par de centrais  $m$  excepto aqueles que contém (ou consistem no) o feixe  $k$ .

Então

$$\Delta_m(k, j) = \Delta(k, j) * g_{mk} \quad (3.24)$$

onde

$$g_{mk} \doteq 1 - \phi_{mk} (1 - h_{mk}) \quad (3.25)$$

A modificação proposta por Krishnan em [Krishnan89] ao SDDR é usar  $\Delta_m(k, j)$  em vez de  $\Delta(k, j)$  como o custo de transportar uma chamada do par de centrais  $m$  no feixe  $k$  quando  $j$  dos seus circuitos estão ocupados.

Para implementar o FLR, para cada feixe  $k$ , são necessários, além dos custos de estado utilizados no SDDR, os parâmetros  $g_{mk}$  para cada par de centrais  $m$ , para os quais as chamadas possam usar o feixe  $k$ .

### **Análise da modificação proposta**

Se  $m = k$ , está-se a considerar o custo do encaminhamento de uma chamada no ramo directo. Nesse caso, no esquema de encaminhamento não alternativo  $\phi_{mm} \approx 1$  e então, pela equação (3.25),  $g_{mm} \approx h_{mm}$  é igual ao produto dos bloqueios de todos os caminhos admissíveis para o par de centrais  $m$ , excepto o ramo directo  $m$ . Sempre que entre um par de centrais existe mais do que um caminho,

$h_{mm}$  será significativamente pequeno. Então o custo de uma chamada encaminhada directamente no FLR será, em muitos casos, uma pequena fracção do custo que lhe era atribuído no SDR.

Se  $m \neq k$ , está-se a considerar a utilização do feixe  $k$  por uma chamada que está a usar dois feixes. No esquema de encaminhamento não alternativo  $\phi_{mk} \ll 1$  (a menos que o caminho directo não exista), e então  $g_{mk} \approx 1$ . Então, o custo de usar um feixe para uma chamada num caminho indirecto é aproximadamente o mesmo que o custo que lhe era atribuído no SDR.

Então, no método de encaminhamento separável modificado, existe uma preferência superior pelo encaminhamento directo face ao encaminhamento indirecto do que existia no método de encaminhamento separável original.

### 3.5.2.3 Encaminhamento Adaptativo Dependente do Estado ("Adaptative State Dependent Routing" - ASDR)

Nos dois métodos de encaminhamento anteriores, propostos por Krishnan e Ott, a escolha do caminho para uma chamada é feita com base nos 'custos', associados com os estados dos caminhos admissíveis, na altura que a chamada ocorre.

O custo de um caminho é uma estimativa do aumento esperado no bloqueio de chamadas futuras que resultarão da aceitação de uma chamada adicional no caminho, no seu estado actual. Estes custos são determinados a partir da solução *off-line* de um grande programa não linear, o qual requer o conhecimento prévio de todas as intensidades de tráfego oferecidas à rede. Como tal informação não é mais do que uma previsão, é então necessário considerar ainda os erros resultantes da mesma. Será portanto vantajoso poder substituir, a necessidade do conhecimento da informação à priori, por informação obtida das medições feitas na rede quando está em funcionamento. O ASDR, descrito em [Krishnan91], é um método de encaminhamento que tem essa vantagem, sendo que aqui cada um dos feixes da rede usa as suas medidas de tráfego para actualizar os seus custos de estado em intervalos regulares.

O método de encaminhamento ASDR não precisa das estimativas à priori das cargas oferecidas à rede. Usa, em vez disso, as medidas de tráfego actuais obtidas dos feixes da rede durante a sua operação.

A hipótese da independência dos fluxos nos feixes é razoável para o encaminhamento não alternativo e é usada na derivação do SDR. No entanto, no SDR, o fluxo nos feixes torna-se correlacionado pela escolha (dos caminhos) dependente do estado. Contudo, para conseguir um tratamento analítico manejável, é-se mais ou menos obrigado a continuar a ficção de que, mesmo no encaminhamento dependente do estado, os fluxos resultantes nos feixes podem ser tratados como fluxos independentes. Então, considerando que o fluxo induzido nos feixes pelo encaminhamento dependente do estado é

aproximado por fluxos independentes, e se, além disso, cada fluxo no feixe for aproximado por uma carga ‘equivalente’ de Poisson oferecida ao feixe, então a determinação das condições da rede, produzidas pelo encaminhamento dependente do estado, é equivalente a determinar as cargas ‘equivalentes’ de Poisson oferecidas aos feixes. Na prática, deixa-se a rede ‘calcular’ estes fluxos, durante a sua operação real sob o método de encaminhamento, e pode-se então medir, em vez de calcular, os resultados necessários, sujeitos às considerações da independência das cargas nos feixes modeladas como fluxos de Poisson.

Deve ser notado que, sob o encaminhamento dependente do estado, a noção de carga ‘oferecida’ a um feixe específico não é bem definida, visto que, quando uma chamada tem mais do que um caminho admissível, a carga nunca é oferecida a um feixe quando é conhecido que não pode ser transportada nele. Por outro lado, a carga transportada num feixe (número médio de circuitos ocupados) é um conceito bem definido independente das regras de encaminhamento e é, de facto o que pode ser medido directamente. Para uma determinação aproximada de cada fluxo no feixe, define-se a carga oferecida ‘equivalente’ como sendo a carga oferecida de Poisson a qual produziria a mesma carga transportada como o dado fluxo no feixe. Então, em princípio, uma carga de Poisson oferecida a um feixe pode ser determinada a partir de uma medição da carga transportada. Na prática, é estimada por combinação de medidas separadas da carga transportada e do bloqueio, pela fórmula seguinte:

$$\text{carga oferecida} = \frac{\text{carga transportada}}{(1 - \text{bloqueio})} \quad (3.26)$$

Também, o ‘bloqueio’ (razão entre os ‘transbordos’ e as ‘tentativas’) num grupo de circuitos não é bem definido nos métodos de encaminhamento dependentes do estado, pois a ideia de uma chamada sendo oferecida a um grupo e ‘transbordar’ dele, não é aplicável à escolha dependente do estado de um caminho de entre vários alternativos. Contudo, considerando a hipótese de uma carga oferecida de Poisson, a ideia de bloqueio como a proporção de chamadas ‘bloqueadas’ é equivalente à ideia de bloqueio como a proporção de tempo passado pelo grupo de circuitos no estado ‘todos os circuitos ocupados’. Esta última interpretação permite obter medições directas do bloqueio, a partir do mesmo conjunto de amostras (da utilização dos circuitos) que são usadas para medir a carga transportada num grupo de circuitos. A carga transportada é medida examinando o estado do grupo de circuitos em intervalos regulares (o intervalo standard é de 100 segundos) e guardando o número de circuitos ocupados observado; para medir o ‘bloqueio’ é necessário simplesmente manter a contagem das ocasiões em que se encontram todos os circuitos ocupados. Pode usar-se, então, a expressão anterior para determinar a carga oferecida de Poisson ‘equivalente’ a um grupo de circuitos, das medidas da carga transportada e da fracção de tempo no estado de bloqueio.

Então, assumindo a aproximação de que os fluxos nos feixes podem ser representados como originados a partir de cargas independentes de Poisson oferecidas aos feixes, é possível substituir o passo da determinação dos valores  $(y_k)$  por medições feitas na rede.

### O funcionamento do ASDR

1. Assumir um conjunto inicial de cargas nominais no feixe  $\{y_k\}$  para usar na equação (3-21), e implementar o SSDR de acordo com as regras dadas. (Uma sugestão para a escolha inicial é considerar  $y_k = s_k$ , o número de circuitos no feixe  $k$ .)
2. Em intervalos regulares, de duração suficiente para a rede alcançar o equilíbrio estatístico sob o método de encaminhamento corrente, usar as medições feitas nos feixes (a carga transportada e a fracção de tempo passada no estado ocupado) para reestimar a carga oferecida ‘equivalente’ para cada feixe, pela fórmula (3-26).
3. Usar as estimativas revistas das cargas nos feixes na equação (3-21) para determinar os novos parâmetros  $\{\Delta(k, j)\}$ , e implementar o SSDR correspondente até regressar ao passo 2, no final do intervalo corrente.

O método ASDR não precisa do conhecimento prévio das cargas oferecidas à rede e também não necessita da solução prévia de um problema de optimização não linear. O ASDR, em vez disso, precisa aprender as condições correntes da rede que resultam da política de encaminhamento. Apesar dos bons resultados que foram conseguidos com o método por Krishnan ainda não existem provas da sua convergência [Krishnan91]. Esta implementação adaptativa foi obtida a partir do SSDR, estando ainda em estudo uma implementação desenvolvida a partir do FLR [Krishnan91].

## 3.6 Métodos de encaminhamento dependentes do acontecimento

### 3.6.1 Autómato com Aprendizagem (“Learning Automata” – LA)

Este é um método de encaminhamento por aprendizagem, descrito em [Girard90] e [Key90]. Não são feitas medidas directamente do estado da rede, mas em vez disso o autómato ganha informação acerca do estado da rede indirectamente. Sempre que o autómato toma uma acção recebe uma resposta indirecta do estado da rede. O autómato baseia-se no seguinte conhecimento: “esta acção é melhor do que aquela porque é recebida mais frequentemente uma boa resposta na primeira do que na segunda”. As chamadas são oferecidas aos caminhos possíveis de acordo com uma distribuição de probabilidade,

que é actualizada em intervalos de tempo discretos de acordo com a informação de realimentação (chamada aceite ou chamada rejeitada). A vantagem deste método é a simplicidade, mas com o inconveniente correspondente de que a aprendizagem pode ser muito lenta. Este método de encaminhamento não foi implementado em nenhuma rede prática apenas foram feitos estudos simulacionais.

### 3.6.2 Encaminhamento Dinâmico Alternativo (“Dynamic Alternate Routing” - DAR)

Encaminhamento alternativo dinâmico (DAR) (descrito em [Mitra91a], [Girard90], [Watanabe90] e [Key90]), foi desenvolvido pela British Telecom, e é uma implementação isolada simplificada do método de encaminhamento *Learning Automata*. É baseado numa forma simples de um esquema de aprendizagem, dado que jamais é efectuada qualquer medida do estado, e a realimentação ocorre só através do sucesso ou falha da ligação no caminho alternativo corrente. A única informação (informação local) que o método utiliza é o caminho alternativo correntemente seleccionado e o limiar de reserva de circuitos nos caminhos (a cada feixe é atribuído um parâmetro de reserva de circuitos).

O diagrama de fluxo seguinte ilustra o modo de funcionamento do método de encaminhamento DAR.

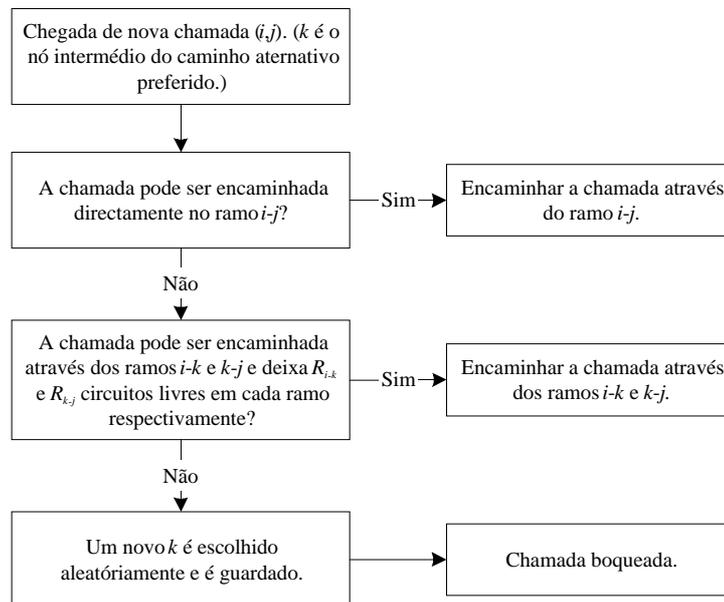


Fig. 3-11 - Diagrama de fluxo de DAR-1. Adaptada de [Mitra91a].

Nas redes DAR, cada chamada é oferecida primeiro ao caminho directo, e se estiver bloqueado, transborda para o caminho alternativo de dois feixes actualmente seleccionado, isto é, usa um caminho alternativo preferido para o tráfego que não pode ser transportado directamente. Então, para cada par de centrais  $(i, j)$ , existe uma central pré seleccionada  $k$  para ser utilizada como central intermédia no

único caminho alternativo permitido. A identidade dessa central intermédia corrente  $k$  está armazenada na central origem.

Se a chamada for transportada no caminho directo a central intermédia permanece inalterada. Se a chamada é oferecida ao caminho alternativo e puder ser transportada nele, a chamada é completada e o caminho alternativo preferido permanece inalterado. Caso contrário, se a chamada é bloqueada no caminho alternativo, dado não poder ser transportada num dos feixes  $i - k$  ou  $k - j$ , a chamada é perdida, e então um novo caminho alternativo (central intermédia) é seleccionado aleatoriamente, de entre todos os caminhos possíveis de dois feixes, para chamadas subsequentes.

### 3.6.2.1 Extensões do DAR

Algumas extensões foram sugeridas ao método DAR [Girard90]:

- Usar uma ordem cíclica na selecção das centrais intermédias.
- Usar uma distribuição não uniforme na selecção das centrais intermédias, com o objectivo de dar preferência ou excluir algumas centrais. Duas possíveis aplicações disto são: o caso de uma falha numa central ou feixe, e o caso de se pretender influenciar o reencaminhamento local.
- Introduzir um segundo parâmetro, com valor superior ao parâmetro de reserva de circuitos. Quando uma chamada está a tentar o caminho alternativo, se o número de circuitos livres se encontra entre o parâmetro de reserva de circuitos e o segundo parâmetro, então a chamada é transportada mas é seleccionado aleatoriamente uma nova central intermédia. Noutras palavras, a resselecção de uma central intermédia pode ocorrer antes da chamada ser bloqueada no caminho.
- Usar caminhos com mais de dois feixes, ou usar um caminho falso como primeiro caminho, no caso da rede não ser completamente ligada.
- Permitir eleger um segundo caminho alternativo, veja-se o diagrama de fluxo apresentado na figura seguinte. Nesta variação de DAR designada por DAR-2 [Mitra91a], tal como no DAR uma chamada que não pode ser encaminhada directamente tenta o caminho alternativo preferido. A diferença é que a chamada não é bloqueada se não puder ser encaminhada neste caminho. Em vez disso, é usado *crankback*, e um novo caminho alternativo preferido é escolhido (como no DAR), e a chamada tenta imediatamente este novo caminho  $i - k'$  e  $k' - j$ . Se houve sucesso, o caminho é mantido como caminho alternativo preferido. Caso contrário, a chamada é bloqueada e é seleccionado aleatoriamente outro caminho alternativo.

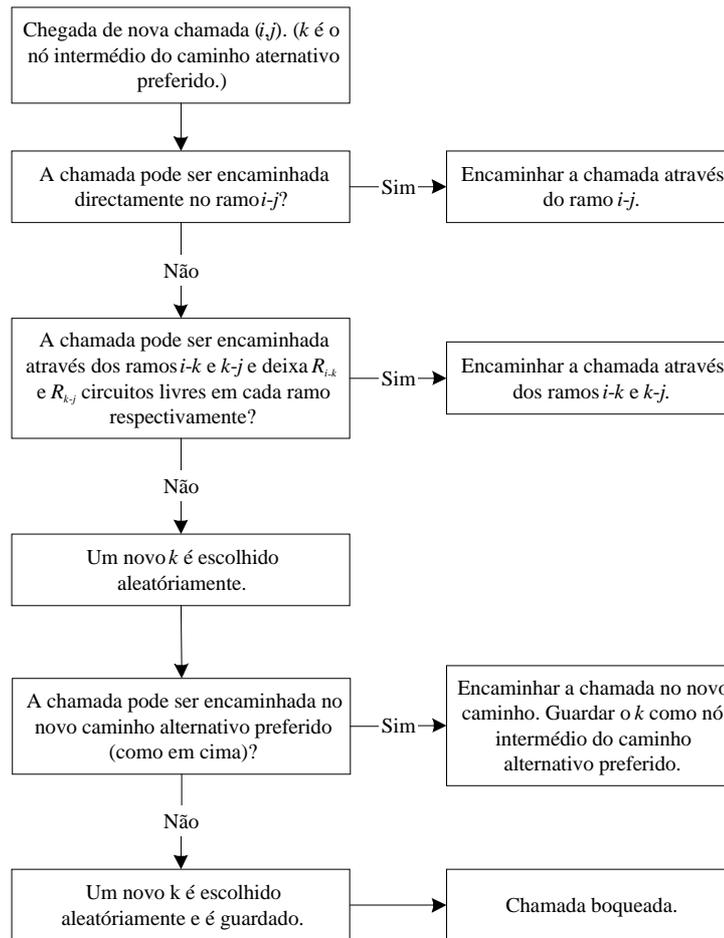


Fig. 3-12 - Diagrama de fluxo de DAR-2. Adaptada de [Mitra91a].

### 3.6.3 Encaminhamento Aleatório com Aprendizagem (“Learning with Random Routing” - LRR)

Este método de encaminhamento é análogo ao método de encaminhamento DAR.

No método de encaminhamento LRR (descrito em [Vargas96]) é seleccionado um novo caminho alternativo apenas quando uma chamada é bloqueada. O LRR é um método de encaminhamento isolado, com actualizações chamada após chamada baseadas, no encaminhamento aleatório. Usa um método de aprendizagem isolado para atingir um encaminhamento flexível.

O ramo directo é usado se disponível, e um caminho alternativo fixo é usado até que ocorra bloqueio. Nesse caso um novo caminho alternativo é seleccionado aleatoriamente para caminho alternativo para as próximas chamadas que transbordarem o ramo directo. Não é usado *crankback*, logo se uma chamada for bloqueada numa central intermédia será perdida. A reserva de circuitos é activada dinamicamente na condição de bloqueio de chamadas.

### 3.6.3.1 Caminho de sucesso para o topo (“Success To the Top” - STT)

É uma extensão do LRR, sendo aqui permitido fazer *crankback* quando se verifica que o caminho alternativo está bloqueado e a chamada se encontra na central intermédia, sendo a chamada oferecida a um novo caminho escolhido aleatoriamente. A capacidade de *crankback* evita a perda de uma chamada num caminho alternativo se existir outro com capacidade livre. Esta característica é que faz com que o seu desempenho seja significativamente superior ao método anterior. No caso limite do STT podem ser tentados todos os caminhos de escolha possíveis por uma chamada antes de ela ser bloqueada, no caso de ir encontrando sempre alternativos sem capacidade disponível. Na selecção do caminho em tempo real, é usado encaminhamento cíclico, e se uma chamada é estabelecida num caminho, a próxima chamada começará por tentar o mesmo caminho (o caminho com sucesso mantém-se no topo da lista de caminhos, dando o nome ao método). Note que este método não é o mesmo que o método STT descrito no capítulo 2, onde era usado encaminhamento sequencial e não cíclico.

### 3.6.4 Sequências Aleatórias Persistentes (“Sticky Random Sequences” – SRS)

O método de encaminhamento SRS (descrito em [Mitra91a]) também usa um algoritmo de escolha aleatória. É similar ao método de encaminhamento DAR, excepto que no SRS é gerado um par ordenado  $(k, k')$  como um par de caminhos alternativos preferidos. Veja-se o diagrama de fluxo seguinte.

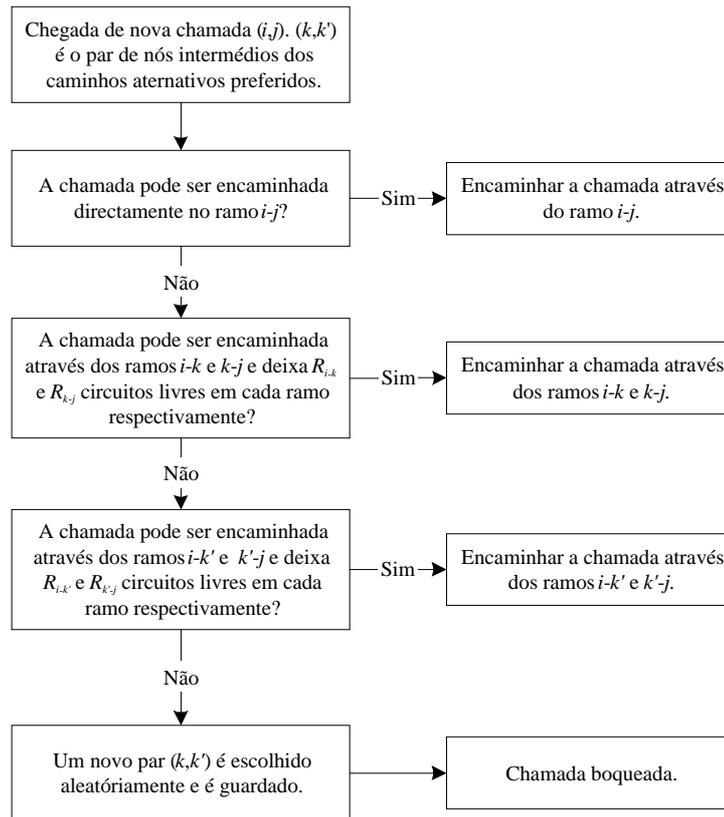


Fig. 3-13 - Diagrama de fluxo de SRS. Adaptada de [Mitra91a].

Neste método o tráfego pode ser encaminhado alternativamente através da central  $k$  ou  $k'$  se possível. Se a chamada puder ser transportada num dos dois caminhos, o par ordenado  $(k, k')$  continua a definir o par de caminhos preferidos. A ordem é mantida,  $k$  é sempre tentado primeiro. Se a chamada é bloqueada, é escolhido aleatoriamente um novo par de entre todos os pares no sistema, excepto  $(k, k')$ .

## 3.7 Métodos de encaminhamento híbridos/misto

### 3.7.1 Encaminhamento baseado no mapa de estados dos feixes (“Trunk Status Map Routing” - TSMR)

O TSMR (descrito em [Girard90] e [Ash98]) é um método de encaminhamento dinâmico pré planeado que incorpora o encaminhamento variável no tempo com os procedimentos associados ao projecto da rede tal como o DNHR. Os padrões de encaminhamento que dão um projecto de rede quase óptimo para as previsões de carga da rede são determinados nos procedimentos incorporados no projecto da rede. A rede é projectada para acomodar todos os padrões de carga esperados, portanto alguma capacidade permanecerá livre pelo menos algum tempo. Se os padrões de encaminhamento planeados

forem fixos, não se poderá tirar vantagem da capacidade temporariamente livre na rede, para completar chamadas que de outra forma serão bloqueadas. O encaminhamento dinâmico em tempo real procura utilizar esta capacidade livre da rede fazendo actualizações nas tabelas de encaminhamento.

Como discutido anteriormente, o encaminhamento em tempo real implementado no DNHR permite aumentar a sequência pré-planeada de caminhos (caminhos nominais) com uma sequência de caminhos adicionais chamados caminhos de utilização em tempo real. Os caminhos em tempo real são pesquisados sequencialmente para determinar se o número de circuitos livres em cada feixe excede o limiar de reserva. Se é encontrado um caminho em tempo real que tem capacidade livre excedendo o limiar de reserva em ambos os feixes, então a chamada é transportada nesse caminho.

No encaminhamento DNHR a ordem de escolha de um caminho e o conjunto de caminhos são fixos, apenas são modificados em períodos de tempo fixos predeterminados. A adaptação às condições da rede é pequena. A previsão da carga é usada para determinar a sequência de caminhos para cada período de tempo o que é sujeito a erros. Como as tabelas de encaminhamento são calculadas com pouca frequência, não podem tomar em conta períodos curtos de variação de carga.

Era desejável alguma forma de encaminhamento que se adaptasse a essas variações, pelo que surgiu assim uma nova versão do DNHR chamado *Trunk Status Map Routing* (TSMR). O TSMR trabalha com um conjunto de caminhos como o calculado para o DNHR original, mas a selecção efectiva de um caminho para uma chamada a ser encaminhada não é estritamente sequencial. Várias regras foram analisadas para determinar a ordem de selecção do caminho, através de estudos simulacionais [Girard90]. Na sequência destes estudos foi observado que a regra:

- Encaminhar a chamada no primeiro caminho calculado para o DNHR não adaptativo. Se este caminho não for possível, seleccionar o caminho de menor carga;

foi a que se comportou melhor dentro das admissíveis [Girard90], tendo um comportamento quase óptimo.

Na implementação do TSMR existe um mapa de estados dos circuitos (**T**runk **S**tatus **M**ap **T**SM) centralizado, que coordena as mudanças de encaminhamento, em tempo real, entre todas as centrais baseado no estado dos circuitos. Todos os **T** segundos cada central envia para uma base de dados da rede, TSM, uma actualização do número de circuitos livres em cada grupo de circuitos, desde que este número tenha mudado. O TSM recalcula uma nova sequência ordenada de caminhos para cada central, a qual é enviada às centrais para ser usada durante os próximos **T** segundos. A sequência de caminhos armazenados nas centrais consiste em duas partes:

1. O primeiro caminho calculado pela optimização do encaminhamento baseado nas previsões;
2. Os caminhos restantes, que são recomendações enviadas pelo TSM.

Vamos analisar agora mais detalhadamente todo este processo. Cada central na rede envia periodicamente (a cada  $T$  segundos) uma mensagem especial para ser encaminhada através da rede CCS, para o TSM. Estas mensagens indicam o número de circuitos livres em cada feixe ligado a cada central, e são enviadas apenas para os feixes que tenham mudado de estado. O TSM actualiza então os estados dos feixes identificados pela mensagem CCS.

Cada central mantém uma tabela de encaminhamento para cada destino a qual tem duas partes:

1. a primeira parte consiste num único caminho (variável no tempo), chamado o caminho de primeira escolha, o qual é actualizado pelo TSM uma vez em cada período de carga e consiste no primeiro caminho determinado pelo DNHR. Está associada uma *flag* com o caminho de primeira escolha, para permitir saltar um caminho de primeira escolha de 2 feixes, se este estiver ocupado.
2. a segunda parte da tabela de encaminhamento consiste nos caminhos restantes, chamados caminhos de encaminhamento de menor carga. Estes caminhos são actualizados todos os  $T$  segundos de acordo com o critério de menor carga, aplicado ao estado corrente da rede. Na prática, não é necessário recalcular todos os caminhos na segunda lista, pois mudar unicamente a segunda escolha de acordo com a regra de menor carga é suficiente para alcançar um desempenho adequado [Ash98].

Por cada sequência de caminhos que precise de ser actualizada, o TSM determina o caminho e a *flag* nessa sequência que precisam de ser alterados, e transmite estas mudanças à central.

Na figura 3-14 e tabela 3-11 temos um exemplo de actualização da tabela de encaminhamento para duas situações. Na figura 3-14 podemos ver todos os caminhos possíveis de A para B e também a capacidade corrente de todos os feixes desses caminhos. Na tabela 3-11 é mostrada a actualização da tabela de encaminhamento de acordo com a capacidade corrente dos feixes, para dois casos.

Se existir capacidade livre suficiente no caminho de primeira escolha (determinado para cada período de carga) esse caminho deve ser usado para completar a chamada. Notar que o caminho directo não é necessariamente o caminho de primeira escolha. Se não existirem circuitos disponíveis no caminho de primeira escolha, então um segundo caminho é seleccionado, de acordo com o critério de menor carga, como o caminho que tem o maior número de circuitos livres, o qual na figura seguinte corresponde ao caminho A-D-B.

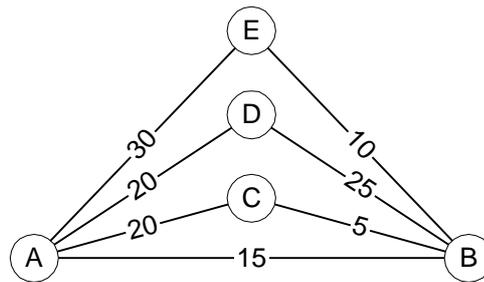


Fig. 3-14 - Caminhos possíveis para A-B. Adaptada de [Ash98].

	Caso 1	Caso 2	
DNHR – Tabela de encaminhamento para A-B no período de carga corrente	A-B A-C-B A-D-B A-E-B	A-E-B A-C-B A-B A-D-B	
Última tabela de encaminhamento enviada pelo TSM para a central A para o tráfego A-B	A-B A-E-B A-D-B A-C-B	A-E-B A-C-B A-D-B A-B	
Número corrente de circuitos livres	A-B=15 A-C-B=5 A-D-B=20 A-E-B=10	A-B=15 A-C-B=5 A-D-B=20 A-E-B=10	
Nova tabela de encaminhamento enviada pelo TSM para a central A para o tráfego A-B	A-B	A-E-B	} Caminho de 1ª escolha
	A-D-B	A-D-B	
	A-E-B	A-C-B	} Caminhos de menor carga
	A-C-B	A-B	

Tab. 3-11 –Tabelas de encaminhamento para A-B. Adaptada de [Ash98].

Uma componente das tabelas de encaminhamento, associada com o caminho de primeira escolha, é variável no tempo, e a outra componente das tabelas de encaminhamento, associada com os caminhos de menor carga, é dependente do estado em tempo real.

O procedimento necessário para implementar o TSMR é partilhado pelas centrais e pelo TSM. O TSM controla o caminho de primeira escolha nas tabelas de encaminhamento das centrais. Existem 2 casos: o caminho de primeira escolha ser caminho directo (caso 1 na tabela 3-11) ou ser um caminho de 2 feixes (caso 2 na tabela 3-11). Se o primeiro caminho determinado para o DNHR, para um período de carga particular, é o caminho directo, então o caminho de primeira escolha (directo) é transmitido para a central no início desse período de carga e não é marcado para “saltar” pelo TSM para todo esse período de carga. Por outro lado, se o primeiro caminho determinado para o DNHR para um período de carga particular for um caminho de 2 feixes, o TSM envia uma mensagem à central, no início desse período de carga, que estabelece o caminho de primeira escolha como sendo esse caminho, também para todo esse período de carga. Mas neste caso, o TSM determina periodicamente se este caminho (de 2 feixes) está ocupado, e se assim for, faz com que esse caminho seja marcado para “saltar” na central para um intervalo de **T** segundos, enviando para isso uma mensagem apropriada para a central. O uso

deste procedimento evita possíveis mensagens de *crankback* que ocorreriam se a central tentasse estabelecer chamadas no caminho ocupado.

A descrição anterior tem a ver apenas com a componente pré-planeada variável no tempo, associada com a manipulação do caminho de primeira escolha. A parte dinâmica do TSMR necessita que um caminho de segunda escolha (primeiro caminho de encaminhamento menos carregado) seja determinado como caminho de menor carga. O TSM determina este caminho menos carregado para cada par de centrais de uma lista de cerca de 20 caminhos candidatos (em média). Esta lista representa a união de todos os caminhos seleccionados pelo DNHR (os nominais e os de utilização em tempo real) para todos os períodos de carga. Em comparação com a manipulação de todos os caminhos do DNHR para cada período de carga, esta lista única de caminhos simplifica a administração do encaminhamento e reduz o espaço de armazenamento. A lista de caminhos candidatos é armazenada no TSM e é diferente da sequência de caminhos realmente armazenada nas centrais, as quais têm um número máximo de caminhos menor. Se o caminho de carga mínima difere do caminho de segunda escolha armazenado na central, o TSM envia uma mensagem apropriada para a central para alterar o conteúdo da sequência de caminhos de menor carga, para reflectir o novo caminho de menor carga, e para a central actualizar a ordem dos restantes caminhos na sequência.

Se o segundo feixe de um caminho de 2 feixes se torna ocupado no intervalo de actualização de  $T$  segundos, e se for tentado encaminhar uma chamada nesse caminho, nessa altura, é usada uma mensagem de *crankback* e a chamada tenta os caminhos subsequentes.

As centrais não necessitam de qualquer capacidade de encaminhamento variável no tempo por si próprias. De facto, todas as capacidades de encaminhamento dinâmicas são controladas pelo TSM, o qual actualiza o caminho de primeira escolha em todos os períodos de carga, e também controla o encaminhamento nos caminhos de menor carga de modo dinâmico. A central é, contudo, o único lugar onde os circuitos são seleccionados e atribuídos a uma chamada particular.

Todos os  $T$  segundos (intervalo de actualização do encaminhamento), o TSM recebe mensagens dos estados dos feixes. Para cada par de centrais, o TSM pode determinar se é necessário fazer a actualização do encaminhamento. A determinação da necessidade, ou não, de fazer a actualização no encaminhamento é apenas feita no caso típico do caminho de primeira escolha ser o caminho directo, e um caminho de 2 feixes ser o caminho de segunda escolha. Caso contrário (na inexistência de um caminho de primeira escolha directo) a actualização será sempre realizada, se houver alteração. Nos fluxos em que existe o ramo directo, se o número de circuitos livres nas duas primeiras escolhas for maior que um determinado limiar que é julgado suficiente para permitir o estabelecimento de todas as chamadas que provavelmente cheguem durante o próximo intervalo de actualização de  $T$  segundos, então não há necessidade de recalculer o caminho de menor carga, porque as chamadas podem ser

encaminhadas com baixa probabilidade de algum *crankback* ser gerado. Esta lógica reduz o número de pares de centrais que precisam de actualização dos caminhos de menor carga em cada intervalo de actualização, e ao mesmo tempo concentra-se na actualização daqueles pares de centrais para os quais essa actualização é benéfica. Se o primeiro caminho é um caminho de 2 feixes, e se o caminho não tem circuitos livres, então o TSM coloca a *flag* “saltar” no estado ligado. O efeito desta acção é evitar a geração de mensagens de *crankback* no caminho de primeira escolha no próximo intervalo de  $T$  segundos. Se o caminho de menor carga é o mesmo que o do período anterior, ou se não existem circuitos livres em nenhum caminho candidato, então a actualização do encaminhamento não é enviada para a central. Apenas quando o caminho de menor carga é alterado é que há lugar ao envio de uma mensagem para a central, alterando o conteúdo da sequência de encaminhamento.

Como exemplo, considere o caso 2 na tabela 3-11. A sequência de caminhos para B tem 4 caminhos de escolha, 3 dos quais são caminhos de escolha de menor carga. O TSM determina que o segundo caminho de escolha de menor carga (A-D-B) se tornou o caminho de menor carga e, como tal, envia esse caminho para a central. O novo caminho é colocado na posição de primeiro caminho na sequência dos caminhos de menor carga, e os restantes caminhos de menor carga são deslocados uma posição para baixo.

Circuitos são reservados automaticamente quando são identificados pares de centrais com bloqueio ponto-a-ponto elevado. São definidos três níveis de bloqueio (I, II ou III), os quais correspondem a limiares de bloqueio ponto a ponto de 1%, 5% e 30% respectivamente. É calculado também o bloqueio total na rede. Se o nível de bloqueio do par de centrais é II ou III são impostos circuitos reservados no ramo directo associado a esse par de centrais. O nível de reserva usado é aproximadamente 3% do número de circuitos no feixe. Se o bloqueio total na rede for superior a 5%, caminhos de 2 feixes sem circuitos livres são cancelados durante os próximos  $T$  segundos, excepto para pares de centrais que tenham níveis de bloqueio ponto a ponto alto (nível III).

Em suma, o TSMR é um método de encaminhamento centralizado, dependente do estado, em tempo real. É baseado no conceito de mapa centralizado de estados dos circuitos, no qual o estado de ocupado/livre de cada feixe na rede é conhecido em tempo real. O método encaminha o tráfego na parte da rede menos congestionada atribuindo chamadas ao caminho de menor carga no conjunto de caminhos de escolha. O primeiro caminho, que é determinado tal como para o DNHR não adaptativo, é usado se existirem circuitos livres. Se este primeiro caminho estiver ocupado, um segundo caminho será seleccionado (de uma lista dos outros caminhos, nominais e de tempo-real, determinados como para o DNHR), aquele que tiver maior número de circuitos livres. O TSMR é então uma extensão do método de encaminhamento DNHR que incorpora algumas características do método RCAR pois os valores dos caminhos são definidos pela sua capacidade residual. É por esta razão um método de

encaminhamento híbrido, pois tem características dos métodos de encaminhamento dependentes do tempo e também dos métodos de encaminhamento dependentes do estado.

Informação mais pormenorizada deste método de encaminhamento pode ser encontrada em [Ash98].

### **3.7.2 Encaminhamento Dependente do Tempo e do Estado (“State and Time-dependent Routing” - STR)**

O encaminhamento dependente do estado e do tempo (descrito em [Kawashima95], [Watanabe90] e [Mase90]) foi desenvolvido pela Nippon Telegraph and Telephone (NTT) e usa um mecanismo de selecção do caminho alternativo que combina o princípio de encaminhamento dependente do acontecimento do tipo DAR com um princípio de encaminhamento dependente do tempo do tipo DNHR. Logo utiliza uma combinação de duas espécies de controlo do encaminhamento, centralizado e isolado.

Neste método de encaminhamento começa-se por gerar os padrões de encaminhamento, sendo então determinado um conjunto ordenado de caminhos alternativos possíveis para cada par de centrais origem-destino. Os padrões de encaminhamento são determinados de acordo com a capacidade e com o tráfego da rede para períodos de tempo específicos, tais como o dia ou a noite (princípio de encaminhamento dependente do tempo).

Nas redes STR, cada chamada é oferecida primeiro ao caminho directo. Se bloqueada, transborda para um caminho alternativo de dois feixes. Os padrões de encaminhamento são alterados localmente em função da chamada ser bem sucedida ou falhar no caminho escolhido. Se a chamada que transbordou do caminho directo é encaminhada no caminho alternativo de dois feixes, esse caminho é mantido como o caminho alternativo atribuído nesse momento. Se o primeiro ou o segundo feixe do caminho alternativo atribuído nesse momento está ocupado, a chamada é bloqueada e o próximo caminho alternativo no conjunto de possíveis caminhos alternativos torna-se o caminho alternativo a usar.

Cada central tem uma lista de caminhos alternativos candidatos que são actualizados periodicamente baseado nas variações de tráfego previstas. Isto significa que o transbordo é restrito a caminhos que foram determinados na etapa de projecto de acordo com uma optimização similar à utilizada pelo DNHR.

Neste método de encaminhamento não é utilizado *crankback* de modo a minimizar o tempo de ocupação pelas chamadas que não são completadas. De modo a assegurar a estabilidade da rede é usada uma reserva de circuitos para dar prioridade às chamadas encaminhadas directamente.

Apresenta-se na figura seguinte um exemplo do modo de funcionamento deste método.

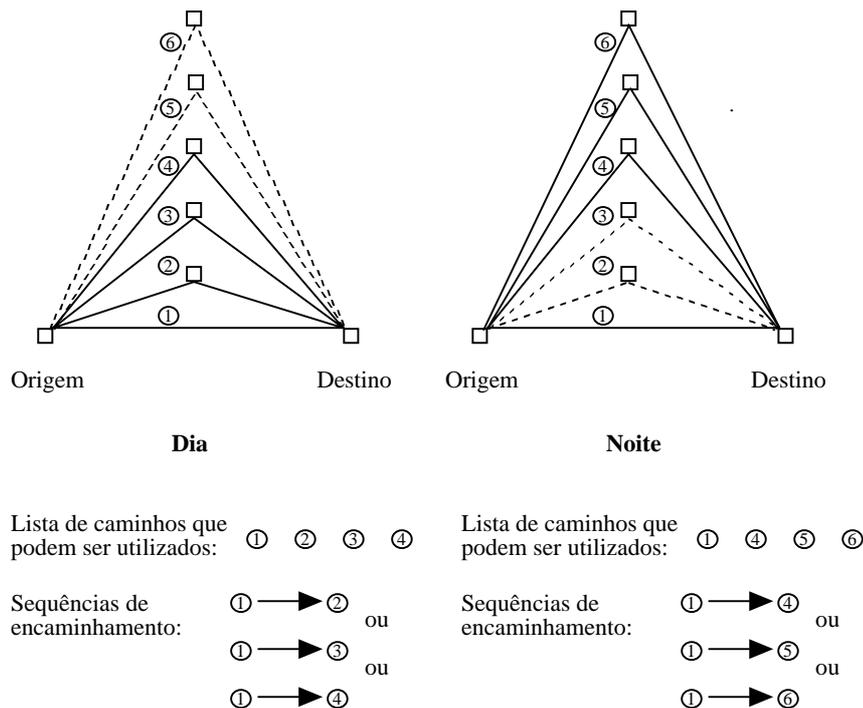


Fig. 3.15 - Método de encaminhamento STR. Adaptada de [Mase90].

### 3.7.2.1 Extensões do STR

Dois esquemas podem ser utilizados para diminuir o número de chamadas bloqueadas:

4 Seleccionar um novo caminho alternativo quando o número de circuitos livres no primeiro ou no segundo feixe do caminho alternativo corrente é menor do que um valor pré-determinado (correspondente a um limiar de congestão), prevenindo assim a perda da próxima chamada. As condições para mudar o caminho alternativo actual são então:

$$x_{i1} < K + T$$

ou

$$x_{i2} < K + T$$

onde  $x_{i1}$  e  $x_{i2}$  são o número de circuitos livres no primeiro e segundo feixe do caminho alternativo corrente  $i$ , respectivamente,  $K$  é o número de circuitos reservados para as chamadas encaminhadas directamente e  $T$  é o limiar de congestão para mudar o caminho alternativo corrente (o número de circuitos livres predeterminado).

5 Um segundo esquema é o mesmo que o anterior excepto no facto de permitir transbordos múltiplos sequenciais. Na figura seguinte, considere que a sequência de encaminhamento corrente é a sequência de encaminhamento (b). Quando surge uma nova chamada, a primeira escolha é sempre o caminho directo {1}. Se o caminho directo está ocupado, a segunda escolha é o caminho alternativo {5}. Se a

chamada é completada nesse caminho então a sequência de encaminhamento corrente é mantida e será usada para a próxima chamada que transborde do caminho directo. Se o primeiro feixe do caminho alternativo {5} está ocupado, a chamada transborda para o próximo caminho alternativo, {6}, de acordo com a sequência de encaminhamento (b). Se a chamada de transbordo é estabelecida no caminho alternativo {6}, a sequência de encaminhamento (c) na qual a segunda escolha é o caminho alternativo {6} torna-se a sequência de encaminhamento corrente.

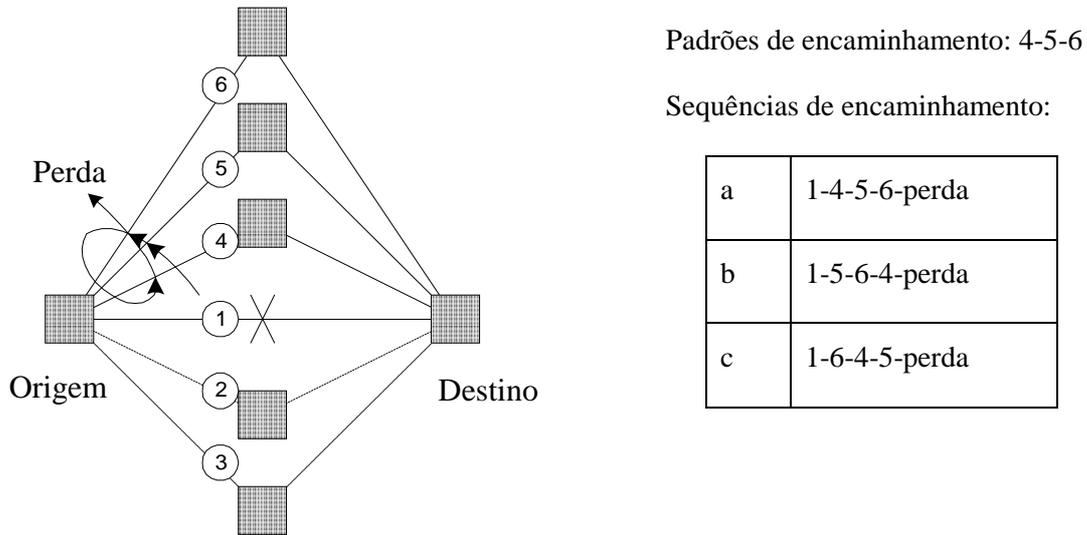


Fig. 3-16 – Encaminhamento no nível da chamada no STR. Adaptada de [Kawashima95].

Neste esquema é procurado, um caminho alternativo com o primeiro feixe livre, na sequência de encaminhamento corrente. Se for encontrado um caminho este é usado pela chamada e é atribuída uma nova sequência de encaminhamento. Isto evita a perda da chamada corrente.



# Capítulo 4

## Parametrização da Rede e Modelos de Tráfego Utilizados

Neste capítulo vai ser apresentada uma representação do sistema (rede de telecomunicações). O objectivo desta representação é obter as definições necessárias à análise em termos de engenharia de teletráfego da rede inter-centrais, em ambiente RDIS. Assume-se que todas as chamadas exigem serviço em tempo real (sistemas com perda), o modo de transferência é a comutação por circuitos, o estabelecimento da ligação é feito a pedido e a comunicação é ponto a ponto. Vão ser considerados diferentes ritmos de transferência simultaneamente.

### 4.1 Representação do sistema

Considere-se um sistema de tráfego (rede inter-centrais) constituído por vários centros de comutação localizados (centrais ou centrais hipotéticas representando uma rede) e grupos de canais de transmissão interligando-as (feixes), transportando tráfego nos dois sentidos.

A formulação matemática utilizada para representar a rede segue muito de perto a notação utilizada em [Esteves91] e [Gomes97]. A rede inter-centrais com comutação por circuitos vai ser representada matematicamente pelo 6-tuplo ordenado,

$$R = (V, L, F, P, C, K, R_a) \quad (4.1)$$

onde:

$V$  é um conjunto finito de elementos, que se designam por centrais (centros de comutação):

$$V = \{v_1, v_2, \dots, v_{|V|}\} \quad (4.2)$$

$L$  é um conjunto de elementos, que se designam por arcos, ramos ou feixes da rede:

$$L = \{l_1, l_2, \dots, l_{|L|}\}, \quad l_x = \{v_i, v_j\}, \quad v_i, v_j \in V, \quad x = 1, 2, \dots, |L| \quad (4.3)$$

$LV(v)$  é um subconjunto de  $L$ , que contém todos os feixes acessíveis a partir de  $v$ :

$$LV(v) = \{l_{v_1}, l_{v_2}, \dots, l_{v_{|LV(v)|}}\} \quad l_{v_j} = \{v, v_j\}, \quad v, v_j \in V, \quad j = 1, 2, \dots, |LV(v)| \quad (4.4)$$

$VV(v)$  é um subconjunto de  $V$ , que contém todas as centrais adjacentes à central  $v$ :

$$VV(v) = \{v'_1, v'_2, \dots, v'_{|VV(v)|}\} \quad l = \{v, v'_j\} \in LV(v), \quad j = 1, 2, \dots, |VV(v)| \quad (4.5)$$

$F$  é o conjunto de todos os fluxos de tráfego:

$$F = \{f \equiv (v_i, v_j) \in V \times V : v_i \neq v_j \wedge A(f) \neq 0\} = \{f_1, f_2, \dots, f_{|F|}\} \quad (4.6)$$

Em que  $A(f)$  é a média do tráfego (em erlangs), na hora mais carregada, oferecido à rede com origem na central  $v_i$  e com destino à central  $v_j$ .

Cada fluxo  $f \in F$  é uma entidade que descreve, em termos de engenharia de teletráfego, o fluxo das chamadas geradas numa central (central origem) e que pretendam ligação pelo sistema a outra central (central destino).

$F^{(k)}$  é o conjunto de fluxos de tráfego da classe  $k \in K$  ( $K$  é definido no final desta secção):

$$F^{(k)} = \{f^{(k)} = (f, k) \in F \times K : A^{(k)}(f^{(k)}) \neq 0\} = \{f_1^{(k)}, f_2^{(k)}, \dots, f_{|F^{(k)|}}^{(k)}\} \quad (4.7)$$

Em que  $A^{(k)}(f^{(k)})$  é a média do tráfego da classe  $k$  (em erlangs), na hora mais carregada, oferecido à rede com origem na central  $v_i$  e com destino à central  $v_j$ .

Cada fluxo  $f^{(k)} \in F^{(k)}$  é uma entidade que descreve o fluxo das chamadas, da classe  $k$ , geradas numa central e que pretendam ligação pelo sistema a outra central.

$P(f)$  é uma sequência de caminhos para o fluxo  $f = (v_s, v_t) \in F$ , constituída pelo conjunto de caminhos que podem ser utilizados pelas chamadas desse fluxo. Essa sequência será definida como o plano de encaminhamento desse fluxo.

$P^{(k)}(f^{(k)})$  é uma sequência de caminhos para o fluxo  $f^{(k)} \in F^{(k)}$ , constituído pelo conjunto de caminhos que podem ser utilizados pelas chamadas desse fluxo.

$$P^{(k)}(f^{(k)}) = \{p^1(f), p^2(f), \dots, p^s(f)\} \quad (4.8)$$

onde  $s = |\mathcal{P}^{(k)}(f^{(k)})|$ .

Em geral cada caminho é definido por uma sequência de feixes contíguos da rede:

$$p(f) = \{l'_1, l'_2, \dots, l'_r\} \quad (4.9)$$

tal que:

$$l'_j = (v'_j, v'_{j+1}) \in L, \quad j = 1, 2, \dots, r \quad (4.10)$$

onde  $v'_1 = v_s$  e  $v'_r = v_t$ .

$\mathcal{P}$  é o conjunto dos planos de encaminhamento:

$$\mathcal{P} = \{\mathcal{P}(f) : f \in F\} \quad (4.11)$$

$C$  uma função que define de forma genérica as capacidades dos feixes:

$$\begin{aligned} C: L &\rightarrow Z_0^+ \\ l &\mapsto C(l) \end{aligned} \quad (4.12)$$

O número de circuitos de transmissão instalados em cada feixe é sempre limitado. O facto de a capacidade de um feixe poder tomar o valor zero representa uma situação de avaria no feixe da rede ou uma situação em que não existe feixe entre os pares de centrais correspondentes.

$\mathcal{K}$  é um conjunto finito das designações de classes de chamadas:

$$\mathcal{K} = \{1, 2, \dots, |\mathcal{K}|\} \quad (4.13)$$

$\mathcal{R}_a$  é o conjunto de regras que juntamente com  $\mathcal{P}$  definem totalmente o algoritmo de encaminhamento para todos os fluxos.

Admite-se que:

- os bloqueios internos nos centros de comutação são nulos ou desprezáveis;
- o tempo de operação nos centros de comutação é desprezável.

## 4.1.1 Tipos de tráfego exógeno

Neste ponto vão ser apresentados os diferentes tipos de tráfego exógeno ao sistema que iremos considerar nos casos em estudo.

Assume-se que o tempo de duração de uma chamada em progresso no sistema é considerado com distribuição exponencial negativa de média  $\mu^{-1}$ , isto qualquer que seja o tipo de tráfego exógeno.

#### 4.1.1.1 Fluxos de Tráfego Gerados por Processos de Poisson

Quando :

- o número de fontes de tráfego é de elevada ordem de grandeza (relativamente ao número de chamadas que podem ser servidas pelo sistema);
- as chegadas ao sistema são estatisticamente independentes;
- o tempo entre duas chamadas consecutivas é considerado com distribuição exponencial negativa de média  $\lambda^{-1}$ .

Admite-se que os fluxos de tráfego oferecidos à rede, são originados como processos de Poisson, independentes entre si.

No caso do fluxo de chegada ser um processo de Poisson a intensidade de chegadas de chamadas ( $\lambda$ ) é constante e independente do número de chamadas já em curso no sistema. Então a probabilidade de ocorrer um pedido de serviço em qualquer intervalo  $(t, t + h)$  é independente do estado do sistema no instante  $t$ . Consequentemente, as distribuições probabilísticas do ponto de vista de um observador exterior ao sistema e de uma chamada que chega ao sistema são iguais.

O tráfego de chamadas oferecido ( $a$ ) é definido como o produto da intensidade de chegada ( $\lambda$ ) pelo tempo de serviço médio ( $\mu^{-1}$ ), e é medido em erlangs.

$$a = \lambda / \mu \quad (4.14)$$

O tráfego de circuitos oferecido,  $a'$ , é o número médio de circuitos simultaneamente ocupados pelas chamadas em serviço num sistema com um número infinito de circuitos, isto é, é o número médio de ocupações num grupo infinito de circuitos.

$$a' = c \times a \quad (4.15)$$

em que  $c$  é o número de circuitos usados para servir cada chamada desse fluxo de tráfego.

O tráfego de chamadas transportado,  $a_c$ , número médio de chamadas simultaneamente em serviço no sistema, é dado pela fracção do tráfego de chamadas oferecido que é aceite pelo sistema.

$$a_c = a(1 - B) \quad (4.16)$$

onde  $B$  é a congestão de chamadas.

O tráfego de circuitos transportado,  $a_c'$ , é o número médio de circuitos simultaneamente ocupados pelas chamadas em serviço, sendo dado pelo produto do tráfego de chamadas transportado pelo número de circuitos necessários para servir cada chamada.

$$a_c' = c \times a_c \quad (4.17)$$

#### 4.1.1.2 Fluxos de Tráfego Gerados por Populações Finitas

Seguem-se algumas considerações em relação a um sistema no qual os pedidos de serviço são gerados por um número finito de fontes (idênticas). Nestes sistemas a probabilidade que um pedido de serviço ocorra num intervalo  $(t, t+h)$  não é independente do estado do sistema no instante  $t$ . Essa probabilidade dependerá em geral do número de fontes livres (e portanto disponíveis para gerar novos pedidos) no instante  $t$ . Conseqüentemente, as distribuições probabilísticas do ponto de vista de um observador exterior ao sistema e de uma chamada que chega ao mesmo não são iguais, ao contrário do que se passa nos fluxos de Poisson.

Vamos assumir que se uma fonte particular está livre no instante  $t$ , a distribuição de tempo de  $t$  até à próxima geração de pedido de serviço é exponencial com média  $\gamma^{-1}$ ; isto é, a fonte não originará um pedido de serviço em  $(t, t+x)$  com probabilidade  $e^{-\gamma x}$ .

Cada fonte gera pedidos independentes dos estados das outras fontes à intensidade  $\gamma$  quando livre (quando não está a ser servida) e zero de outro modo. Este processo de chegada é denominado Chegadas QuaseAleatórias ou de Engset [Cooper81].

No caso do fluxo de chegada ser gerado por um número finito de fontes de tráfego, a intensidade de chegada de chamadas do fluxo ( $\lambda_j$ ) dependente do número de chamadas já em curso no sistema, isto é

$$\lambda_j = (n-j)\gamma, \quad 0 \leq j \leq n \quad (4.18)$$

sendo  $n$  o número de fontes que geram chamadas,  $j$  o número de fontes a ser servidas e  $\gamma$  a intensidade de chegadas por fonte livre. Então quando temos chegadas QuaseAleatórias, o tempo até à próxima chegada de uma chamada tem uma distribuição exponencial com média  $1/\lambda_j$ .

Como no caso de um número infinito de fontes também aqui é importante definir carga oferecida e carga transportada. Podemos manter as definições utilizadas no caso de entradas de Poisson, no entanto aqui a intensidade de chegadas  $\lambda_j$  depende do estado do sistema.

Denota-se por  $\beta$  o tráfego médio de chamadas por fonte livre (diferente do tráfego médio por fonte), sendo  $\beta = \gamma/\mu$ .

A relação entre  $\beta$  e a carga total oferecida  $a$  não é uma relação simples, no nosso estudo simulacional vamos considerar a aproximação:

$$a \approx n \times \beta \quad (4.19)$$

que é válida para um bloqueio de chamadas suficientemente baixo.

## 4.1.2 Tráfego Multiclasse com Perda

Em alguns dos sistemas em estudo vamos considerar tráfego multiclasse com perda. O tráfego é multiclasse quando há chamadas provenientes de fluxos de tráfego com diferentes características que partilham os recursos comuns da rede. Cada tipo de fluxo de tráfego constitui uma classe com características próprias.

As características que diferenciam os vários fluxos de tráfego podem ser os parâmetros dos processos estocásticos de chegada e/ou terminação, as próprias distribuições estatísticas subjacentes aos intervalos de tempo entre chegadas e terminações da mesma, o tamanho da população de utilizadores que geram os diferentes fluxos de tráfego, e parâmetros associados às próprias chamadas.

Vamos restringir o nosso estudo a fluxos de tráfego com perda o que significa que as chamadas que não podem ser servidas imediatamente são recusadas, isto é, não esperam por serviço, em oposição ao que se passa nos fluxos de tráfego com espera.

## 4.1.3 Parâmetros do tráfego – notação

Sendo  $R = (\mathcal{V}, \mathcal{L}, \mathcal{F}, \mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{R}_a)$  a representação da rede associada ao sistema, como já definido, vamos apresentar de seguida a notação para os parâmetros utilizados na análise de tráfego.

- $A^{(k)} : \mathcal{F}^{(k)} \rightarrow \mathbb{R}_0^+$ ,  $f^{(k)} \mapsto A^{(k)}(f^{(k)})$ ;  $\forall k \in \mathcal{K}$   
 $A^{(k)}(f^{(k)}) \equiv$  Média do tráfego exógeno do fluxo  $f^{(k)}$  da classe  $k$  (em erlangs).
- $A_L^{(k)} : \mathcal{L} \rightarrow \mathbb{R}_0^+$ ,  $l \mapsto A_L^{(k)}(l)$ ;  $\forall k \in \mathcal{K}$   
 $A_L^{(k)}(l) \equiv$  Média do tráfego (global) de circuitos oferecido ao feixe  $l$  pelas chamadas da classe  $k$ .
- $A_L : \mathcal{L} \rightarrow \mathbb{R}_0^+$ ,  $l \mapsto A_L(l)$   
 $A_L(l) \equiv$  Média do tráfego (global) de circuitos oferecido ao feixe  $l$ .
- $A : \mathcal{F} \rightarrow \mathbb{R}_0^+$ ,  $f \mapsto A(f)$   
 $A(f) \equiv$  Média do tráfego conjunto de chamadas oferecido à rede pelo fluxo  $f$  (em erlangs).

- $B^{(k)} : F^{(k)} \rightarrow [0,1] \subset \mathbb{R}$ ,  $f^{(k)} \mapsto B^{(k)}(f^{(k)})$ ;  $\forall k \in \mathbb{K}$   
 $B^{(k)}(f^{(k)}) \equiv$  Bloqueio médio ponto a ponto (ou seja central a central) do fluxo  $f^{(k)}$  da classe  $k$  (define a probabilidade de uma chamada do fluxo  $f^{(k)}$  da classe  $k$  ser perdida).
- $B : F \rightarrow [0,1] \subset \mathbb{R}$ ,  $f \mapsto B(f)$   
 $B(f) \equiv$  Bloqueio médio conjunto ponto a ponto (ou seja central a central) para cada um dos fluxos (define a probabilidade de uma chamada do fluxo  $f$  ser perdida).
- $B_L : L \rightarrow [0,1] \subset \mathbb{R}$ ,  $l \mapsto B_L(l)$   
 $B_L(l) \equiv$  Bloqueio médio (global) de chamadas no feixe  $l$  (define a probabilidade de uma chamada arbitrária oferecida ao feixe  $l$  ser perdida).
- $B_{1L} : L \rightarrow [0,1] \subset \mathbb{R}$ ,  $l \mapsto B_{1L}(l)$   
 $B_{1L}(l) \equiv$  Bloqueio médio (global) de circuitos no feixe  $l$  - equivalente de chamadas monocanal (define a probabilidade de uma “chamada” monocanal oferecida ao feixe  $l$  ser perdida).
- $F^{(k)} : \mathcal{V} \rightarrow \mathbb{Z}_0^+$ ,  $v \mapsto F^{(k)}(v)$ ;  $\forall k \in \mathbb{K}$   
 $F^{(k)}(v) \equiv$  Número de fontes que geram chamadas da classe  $k$  na central  $v$  (no caso do fluxo de tráfego  $k$  ser gerado por uma população finita).
- $C^{(k)} \equiv$  Número de circuitos necessários a uma chamada da classe  $k$ .
- $x^{(k)} : \mathcal{V} \rightarrow \mathbb{Z}_0^+$ ,  $v \mapsto x^{(k)}(v)$ ;  $\forall k \in \mathbb{K}$   
 $x^{(k)}(v) \equiv$  Número de chamadas da classe  $k$  em curso na central  $v$ .
- $\lambda^{(k)}(x^{(k)}(v)) \equiv$  Intensidade de chegada de chamadas da classe  $k$  na central  $v$ , quando há  $x^{(k)}(v)$  chamadas dessa classe em curso.
- $\frac{1}{\mu^{(k)}} \equiv$  Tempo médio de serviço das chamadas da classe  $k$ .
- $\lambda^{(k)} \equiv$  Intensidade de chegada de chamadas da classe  $k$  (no caso do fluxo de tráfego  $k$  de Poisson).
- $\gamma^{(k)}(v) \equiv$  Intensidade de chegadas de chamadas da classe  $k$  por fonte livre da classe  $k$  na central  $v$  (no caso do fluxo de tráfego  $k$  ser gerado por uma população finita).
- $\beta^{(k)}(v) = \gamma^{(k)}(v) / \mu^{(k)}$  - Tráfego médio de chamadas por fonte livre da classe  $k$  na central  $v$  (no caso do fluxo de tráfego  $k$  ser gerado por uma população finita).
- $AV^{(k)} : \mathcal{V} \rightarrow \mathbb{R}_0^+$ ,  $v \mapsto AV^{(k)}(v)$ ;  $\forall k \in \mathbb{K}$   
 $AV^{(k)}(v) \equiv$  Tráfego médio de chamadas gerado pelas fontes livres da classe  $k$  na central  $v$  (no caso do fluxo de tráfego  $k$  ser gerado por uma população finita).
- $B_G^{(k)} \equiv$  Bloqueio médio global na rede inter-centrais para chamadas da classe  $k$  (define a probabilidade de uma chamada de qualquer fluxo da classe  $k$  ser rejeitada pela rede):

$$B_G^{(k)} = \frac{\sum_{f^{(k)}} A^{(k)}(f^{(k)}) \times B^{(k)}(f^{(k)})}{A_G^{(k)}} \quad (4.20)$$

onde  $A_G^{(k)}$  é a média do tráfego global oferecido à rede inter-centrais pelas chamadas da classe  $k$  :

$$A_G^{(k)} = \sum_{f^{(k)}} A^{(k)}(f^{(k)}) \quad (4.21)$$

- $B_G$  - Congestão média de chamadas na rede inter-centrais (define a probabilidade média de uma chamada de qualquer fluxo ser rejeitada pela rede):

$$B_G = \frac{\sum_f A(f)B(f)}{\sum_f A(f)} = \frac{\sum_k \sum_{f^{(k)}} A^{(k)}(f^{(k)})B^{(k)}(f^{(k)})}{\sum_k \sum_{f^{(k)}} A^{(k)}(f^{(k)})} \quad (4.22)$$

- $B_{1G}$  - Bloqueio global médio de circuitos (equivalente de chamadas monocanal) na rede inter-centrais (define a probabilidade média de uma “chamada” monocanal de qualquer fluxo ser rejeitada pela rede):

$$B_{1G} = \frac{\text{tráfego total de circuitos perdido na rede}}{\text{tráfego total de circuitos oferecido à rede}} \quad (4.23)$$

então

$$B_{1G} = \frac{\sum_k \sum_{f^{(k)}} C^{(k)} A^{(k)}(f^{(k)}) B^{(k)}(f^{(k)})}{\sum_k \sum_{f^{(k)}} C^{(k)} A^{(k)}(f^{(k)})} \quad (4.24)$$

onde  $C^{(k)} A^{(k)}(f^{(k)})$  é o tráfego de circuito oferecido pelas chamadas do fluxo  $f^{(k)}$ , ou seja é o tráfego de chamadas monocanal equivalente ao tráfego  $A^{(k)}(f^{(k)})$ .

Definem-se agora algumas matrizes e vectores auxiliares, a partir dos parâmetros de tráfego apresentados anteriormente, para as quais se supõe fixada uma ordenação para os elementos dos conjuntos  $\mathcal{V}$ ,  $\mathcal{L}$ ,  $\mathcal{F}$  e  $\mathcal{K}$ .

- Matriz do tráfego exógeno da classe  $k$ :  $\bar{A}^{(k)} = [A_{ij}^{(k)}]$ ,  $i, j = 1, 2, \dots, |\mathcal{V}|$

$$A_{ij}^{(k)} = \begin{cases} A^{(k)}(v_i, v_j, k) & \text{se } f^{(k)} = (v_i, v_j, k) \in \mathcal{F}^{(k)} \\ 0 & \text{caso contrário} \end{cases} \quad (4.25)$$

- Vector do tráfego (global) oferecido aos feixes:  $\bar{AL} = [AL_i]$ ,  $i = 1, 2, \dots, |\mathcal{L}|$

$$AL_i = A_{\mathcal{L}}(l_i); \quad i = 1, 2, \dots, |\mathcal{L}|, l_i \in \mathcal{L} \quad (4.26)$$

- Vector de bloqueios médios (globais) nos feixes:  $\bar{BL} = [BL_i]$ ,  $i = 1, 2, \dots, |\mathcal{L}|$

$$BL_i = B_L(l_i); \quad i = 1, 2, \dots, |L|, l_i \in L \quad (4.27)$$

- Vector de capacidade dos feixes:  $\bar{C} = [c_i]$ ,  $i = 1, 2, \dots, |L|$

$$c_i = C(l_i); \quad i = 1, 2, \dots, |L|, l_i \in L \quad (4.28)$$

## 4.2 Especificação dos três casos particulares do sistema em estudo

### 4.2.1 Definição da Rede original

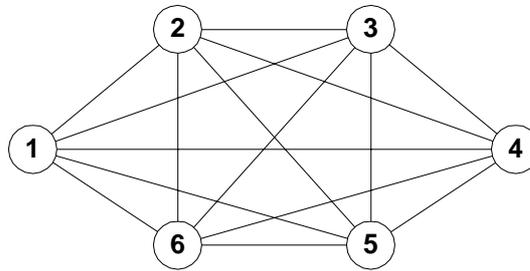


Fig. 4-1 - Topologia da rede. Adaptada de [Mitra91a].

Apresentam-se de seguida os parâmetros que definem a rede que usamos como caso de estudo neste trabalho.

Feixe	Circuitos								
1-2	36	2-3	96	3-4	12	4-5	192	5-6	336
1-3	24	2-4	96	3-5	48	4-6	84	-	-
1-4	324	2-5	108	3-6	24	-	-	-	-
1-5	48	2-6	96	-	-	-	-	-	-
1-6	48	-	-	-	-	-	-	-	-

Tab. 4-1 – Número de circuitos em cada feixe da rede.

Feixe	Tráfego								
1-2	27,47	2-3	25,11	3-4	11,92	4-5	79,42	5-6	127,11
1-3	6,97	2-4	101,61	3-5	6,86	4-6	83,00	-	-
1-4	257,81	2-5	76,78	3-6	13,25	-	-	-	-
1-5	20,47	2-6	82,56	-	-	-	-	-	-
1-6	29,11	-	-	-	-	-	-	-	-

Tab. 4-2 – Carga de tráfego exógeno (em Erlang) oferecido à rede.

Esta rede foi definida por Mitra e Seery em [Mitra91a] e foi escolhida por ter sido utilizada em diversos artigos entre os quais se encontram [Chung93], [Berezner97] e [Ash93].

A rede descrita é completamente ligada. Para cada par de centrais foi especificado o número de circuitos, tabela 4-1, e o tráfego oferecido, tabela 4-2. Note-se que a rede é altamente assimétrica e que a carga de tráfego exógeno oferecido ao par de centrais 2-4 excede o número de circuitos do seu ramo directo. Considera-se o tráfego inter-nodal simétrico e os feixes bidireccionais.

Nos sistemas que vamos considerar as chamadas das diferentes classes podem utilizar diferentes larguras de banda e diferentes intensidades de chegada e terminação. No sistema 3 existe ainda uma outra diferença pois existem dois fluxos de tráfego em que o tamanho da população que os gera é limitado e não infinito como se passa nos restantes casos (onde o tráfego gerado é de Poisson).

As chamadas que não são aceites pelo sistema são perdidas e não o influenciam, como já referido o estudo é restringido a fluxos de tráfego com perda.

O método de encaminhamento utilizado é comum às diferentes classes de chamadas (se bem que os parâmetros a usar nesses métodos e resultados obtidos possam variar por classe de chamada). Em certos casos, uma chamada recém chegada de qualquer classe pode ser rejeitada mesmo havendo recursos livres suficientes para a servir, de forma a garantir uma mais justa alocação de recursos pelas diferentes classes. Para o efeito, vários tipos possíveis de estratégias de controlo de acesso podem ser encontradas em [Gomes97] ou [Simões99]. É também usada reserva de circuitos nos feixes, comum a todas as classes, nos caminhos de transbordo.

Cada sistema vai ser analisado considerando várias situações de tráfego incorporando tráfego de pico, tráfego moderado e tráfego suave.

## 4.2.2 Sistema 1

Considere-se uma rede de comutação por circuitos, como definido anteriormente, em que todo o tráfego gerado é de Poisson monocanal (tráfego telefónico). Como temos neste sistema apenas uma classe de chamadas, não se utiliza aqui o índice  $k$ .

A matriz de tráfego  $[A_{ij}]$  é triangular pois considera-se que o tráfego inter-nodal é simétrico.

O fluxo de tráfego tem uma intensidade de chegadas  $\lambda$  chamadas/segundo e cada chamada utiliza um circuito (tráfego monocanal como já referido), em todos os feixes ao longo do seu caminho, para ser servida. E cada chamada é servida durante um intervalo de tempo que é uma variável aleatória com distribuição exponencial negativa (DEN) de média  $\mu^{-1}$  segundos.

## 4.2.3 Sistema 2

Consideramos agora a rede de comutação por circuitos com a topologia anterior em que o tráfego oferecido resulta de uma mistura de dois tipos de fluxos de Poisson, sendo a largura de banda associada às chamadas de cada classe diferente para cada um dos tipos de fluxos de tráfego. Como temos 2 tipos de classes de chamadas então  $|K| = 2$ .

### *Fluxo tipo 1 (64Kbits/s):*

O número de circuitos necessários às chamadas desta classe é  $C^{(1)} = 1$ .

Tipo de fluxo de tráfego como o definido no Sistema 1, com intensidade de chegada  $\lambda^{(1)}$  e tempo de serviço médio  $1/\mu^{(1)}$ . Com matriz de tráfego  $[A_{ij}^{(1)}] = [A_{ij}]$ .

### *Fluxo tipo 2 (128Kbits/s):*

O número de circuitos necessários às chamadas desta classe é  $C^{(2)} = 2$ .

Este tipo de fluxo de tráfego tem uma intensidade de chegadas  $\lambda^{(2)}$  chamadas/segundo. Cada uma das chamadas utiliza dois circuitos (tráfego multicanal), para ser servida durante um intervalo de tempo que é uma variável aleatória com DEN de média  $1/\mu^{(2)}$  segundos. Os circuitos são necessários durante todo o tempo de serviço da chamada, em todos os feixes ao longo do seu caminho.

Este tipo de fluxo é gerado de acordo com a matriz de tráfego  $[A_{ij}^{(2)}]$ .

Será necessário fazer um redimensionamento da rede pois além do tráfego para o qual foi planeada está-se agora a oferecer outro tipo de tráfego.

### 4.2.3.1 Redimensionamento grosseiro da rede original

Neste ponto vamos determinar o novo número de circuitos em cada feixe da rede, pelo motivo de pretendermos oferecer à rede além do tipo de fluxo de tráfego original um segundo tipo de fluxo de Poisson.

Na determinação do novo número de circuitos de cada feixe da rede vamos tentar obter uma rede com desempenho semelhante ao obtido na rede original. Para o conseguir vamos tentar aproximar os bloqueios médios de circuitos nos feixes para a rede com os dois tipos de fluxos (que designaremos por rede actual) aos bloqueios médios de circuitos nos feixes para a rede com apenas um tipo de fluxo (rede original), quando é usado um método de encaminhamento que permite às chamadas tentarem caminhos alternativos.

Sendo conhecida a matriz  $[A_{ij}^{(1)}]$ , e sabendo que a repartição do tráfego total de chamadas por cada classe é:

Classe de chamadas	Nº de circuitos	% do tráfego de chamadas
1	1	83,33%
2	2	16,67%

Tab. 4-3 - % do tráfego (de chamadas) oferecido por cada uma das classes de chamadas.

Vai supor-se que quase todo o tráfego, dos dois tipos de fluxos, é encaminhado directamente. Com esta consideração e com a informação anterior é possível calcular a média do tráfego (global) de circuitos oferecido ao feixe  $l$ ,  $A_L(l)$ .

Da suposição de que todo o tráfego é encaminhado directamente temos que:

$$A_L(l) \cong \sum_{k=1}^2 A^{(k)}(f^{(k)}) \times C^{(k)} : \tag{4-29}$$

$$f^{(k)} = (v_i, v_j, k) \in F^{(k)}, l = \{v_i, v_j\}, \forall l \in L$$

Como o necessário para os cálculos é o tráfego de circuitos e não de chamadas, é necessário converter os valores de percentagem de tráfego de chamadas da classe  $k$  ( $p^k$ ) apresentados na tabela 4-3, na percentagem correspondente de circuitos ocupados pelas chamadas desta classe ( $p_L^k$ ).

Sabendo que as percentagens de tráfego de chamadas relacionam os tráfegos de chamadas das diversas classes (equação 4-30):

$$\frac{A^{(k)}(f^{(k)})}{A^{(k')}(f^{(k')})} = \frac{p^k}{p^{k'}} \tag{4-30}$$

Através das equações 4-29 e 4-30 podemos calcular as percentagens de circuitos ocupados pelas chamadas da classe  $k$  ( $p_L^k$ ):

$$p_L^k = \frac{A_L^{(k)}(l)}{A_L(l)} = \frac{p^k \times C^{(k)}}{\sum_{k=1}^2 p^k \times C^{(k)}} \tag{4-31}$$

Classe de chamadas	Nº de circuitos	% do tráfego de circuitos
1	1	71,43%
2	2	28,57%

Tab. 4-4 - % do tráfego (de circuitos) oferecido por cada uma das classes de chamadas.

Considerando a expressão 4.29 juntamente com a informação obtida da tabela anterior, podemos calcular  $A_L(l)$  a partir de:

$$A_L(l) = \frac{A^{(1)}(f^{(1)})}{0.7143}; \quad f^{(1)} = (v_i, v_j, 1) \in F^{(1)}, l = \{v_i, v_j\}, \quad \forall l \in L \quad (4.32)$$

onde  $A^{(1)}(f^{(1)})$  é a média do tráfego exógeno do fluxo  $f^{(1)}$  da classe 1, que é conhecida, pois é conhecida a matriz  $[A_{ij}^{(1)}]$ .

Conhecendo  $A_L(l)$  podemos calcular  $A^{(2)}(f^{(2)})$  a partir de:

$$A^{(2)}(f^{(2)}) = \frac{0.2857}{2} A_L(l); \quad \forall f^{(2)} = (v_i, v_j, 2) \in F^{(2)} : l = \{v_i, v_j\} \in L \quad (4.33)$$

Pois  $C^{(1)} = 1$ ,  $C^{(2)} = 2$ .

Assumindo que são conhecidos, para a rede original (apenas com tráfego  $A^{(1)}(f^{(1)})$ ), os bloqueios médios de chamadas nos feixes  $B_L^{orig}(l)$ , vamos calcular os bloqueios médios de circuitos nos feixes (definidos como bloqueios fictícios, equivalentes aos bloqueios para chamadas monocanal) para a rede actual,  $B'_{lL}(l)$ , através de:

$$B'_{lL}(l) \cong 0.7 \times B_L^{orig}(l), \quad \forall l \in L \quad (4.34)$$

Ao considerar a aproximação de todo o tráfego ser encaminhado directamente e ao utilizar o modelo de Erlang-B estamos a subestimar o bloqueio nos feixes. Daí vem a justificação para a expressão anterior, a utilizar no redimensionamento da rede.

E vamos ajustar os tráfegos através de:

$$A'_L(l) \cong 1.2 \times A_L(l) \quad (4.35)$$

sendo esta ampliação justificada pela contribuição do tráfego de trânsito.

Como:

$$B'_{lL}(l) = E_B(C'(l), A'_L(l)), \quad \forall l \in L \quad (4.36)$$

em que  $E_B(C'(l), A'_L(l))$  é a fórmula de Erlang-B, então como é conhecida a média do tráfego oferecido aos feixes e o bloqueio médio de circuitos (equivalente para chamadas monocanal) nos feixes (obtido para a rede original) podemos, utilizando a inversão da fórmula de Erlang-B, determinar o número de circuitos para cada feixe.

### 4.2.3.1.1 Inversão da fórmula de Erlang-B – Determinação do número de circuitos

Considere-se um grupo de  $n \in \mathbb{N}$  circuitos com acesso completo, ao qual é oferecido um tráfego de Poisson com média  $a \in \mathbb{R}^+$ . A probabilidade de bloqueio num tal sistema, denotada por  $B(n, a)$  ou por  $E_B(n, a)$  pode ser determinada pela expressão:

$$B(n, a) = \frac{a^n/n!}{\sum_{j=0}^n a^j/j!} \quad (4.37)$$

que é sobejamente conhecida como fórmula de perda de Erlang ou fórmula de Erlang-B. Esta pode ser estendida, para o caso de um número não inteiro de circuitos  $x \in \mathbb{R}^+$  (equação de Fortet, ver [Syski59]):

$$B(x, a)^{-1} = a \int_0^{+\infty} e^{-ay} (1+y)^x dy, \quad \forall a, x \in \mathbb{R}^+ \quad (4.38)$$

Considerando o valor de  $a$  fixo, para obter o número real de circuitos  $x \in \mathbb{R}^+$  que satisfaz a relação anterior para uma dada probabilidade de bloqueio  $P \in ]0,1[$ , aplicaremos uma iteradora baseada no método de Newton-Raphson [Esteves91]:

$$x_n = x_{n-1} - \frac{B(x_{n-1}, a) - P}{B'_x(x_{n-1}, a)}, \quad n = 1, 2, 3, \dots \quad (4.39)$$

tomando

$$x_0 = (1 - P)(1 + a) \quad (4.40)$$

e

$$B'_x = \frac{\partial B(x, a)}{\partial x} \approx -B(x, a) \frac{\ln \alpha - 1/(2a\alpha)}{1 - B(x, a)/(2\alpha)} \text{ sendo } \alpha = \frac{x+1}{a} + B(x, a) \quad (4.41)$$

### 4.2.3.2 Resultados obtidos para o redimensionamento

Os bloqueios médios de chamadas nos feixes para a rede original,  $B_L^{\text{orig}}(l)$ , usados no redimensionamento, foram obtidos por simulação.

Quando é oferecido à rede original, com um número de circuitos dados na tabela 4.1, o tráfego exógeno dado na tabela 4.2, os bloqueios médios de chamadas nos feixes,  $B_L^{\text{orig}}(l)$ , são nalguns casos muito pequenos, muitas vezes quase nulos. Por esta razão, para fazer o redimensionamento, os valores dos bloqueios médios de chamadas nos feixes para a rede original, foram obtidos considerando que o tráfego oferecido à rede original foi ampliado. Este tráfego foi obtido multiplicando o tráfego original por 1,5 (o

factor de carga considerado foi então de 1,5). Os seus valores, para todos os fluxos, encontram-se na tabela 4-5. Esses tráfegos vão corresponder às médias de tráfego exógeno moderado para cada um dos fluxos da classe 1,  $A^{(1)}(f^{(1)})$ .

		Destinos				
		2	3	4	5	6
Origens	1	41,205	10,455	386,715	30,705	43,665
	2	0	37,665	152,415	115,17	123,84
	3	0	0	17,88	10,29	19,875
	4	0	0	0	119,13	124,5
	5	0	0	0	0	190,665

Tab. 4-5 – Cargas de tráfego exógeno ampliado (factor de carga = 1,5).

Na tabela seguinte encontram-se os valores dos bloqueios médios de chamadas nos feixes,  $B_L^{orig}(l)$ , obtidos através de simulação com a rede original, definida na tabela 4-1, quando a esta é oferecido o tráfego exógeno da tabela 4-5. Na simulação realizada foi escolhido o método de encaminhamento DAR.

		Destinos				
		2	3	4	5	6
Origens	1	49,88%	17,38%	19,01%	25,49%	20,71%
	2	0%	0,03%	43,06%	26,87%	30,74%
	3	0%	0%	75,75%	2,92%	31,59%
	4	0%	0%	0%	3,08%	41,80%
	5	0%	0%	0%	0%	0%

Tab. 4-6 – Bloqueios médios de chamadas nos feixes, para a rede original (com apenas um tipo de fluxo).

Os valores pretendidos para os bloqueios médios de circuitos nos feixes (equivalente de chamadas monocal),  $B_{1L}'(l)$ , para a rede com os dois tipos de fluxos, são os apresentados na tabela 4-7 e foram obtidos pela aproximação indicada na expressão 4-34. Os bloqueios médios de chamadas nos feixes para a rede original,  $B_L^{orig}(l)$ , usados nesses cálculos, foram os apresentados na tabela 4-6.

		Destinos				
		2	3	4	5	6
Origens	1	34,92%	12,17%	13,31%	17,84%	14,50%
	2	0%	0,02%	30,14%	18,81%	21,52%
	3	0%	0%	53,02%	2,05%	22,11%
	4	0%	0%	0%	2,16%	29,26%
	5	0%	0%	0%	0%	0%

Tab. 4-7 – Bloqueios médios de circuitos nos feixes pretendidos (equivalente de chamadas monocal).

Vamos agora determinar os valores para as cargas de tráfego moderado (global) de circuitos oferecidas aos feixes,  $A_L(l)$ , quando são oferecidos à rede os dois tipos de fluxos. Esses valores são apresentados na tabela seguinte e foram obtidos pela expressão 4-32, usando os valores da tabela 4-5.

		Destinos				
		2	3	4	5	6
Origens	1	57,687	14,637	541,401	42,987	61,131
	2	0	52,731	213,381	161,238	173,376
	3	0	0	25,032	14,406	27,825
	4	0	0	0	166,782	174,3
	5	0	0	0	0	266,931

Tab. 4-8 – Cargas de tráfego (global) de circuitos oferecidas aos feixes com os dois tipos de fluxos.

Usando a aproximação indicada na expressão 4-35, na qual se considerou um factor de ampliação de tráfego de trânsito de 1.2, devido ao transbordo, os tráfegos (globais) de circuitos oferecidos aos feixes,  $A_L(l)$ , passam a ser  $A'_L(l)$  cujos valores são apresentados na tabela seguinte.

		Destinos				
		2	3	4	5	6
Origens	1	69,224	17,564	649,681	51,584	73,357
	2	0	63,277	256,057	193,486	208,051
	3	0	0	30,038	17,287	33,390
	4	0	0	0	200,138	209,160
	5	0	0	0	0	320,317

Tab. 4-9 – Cargas de tráfego moderado (global) de circuitos oferecidas aos feixes, ajustadas.

Usando a iteradora baseada no método de Newton Raphson, expressão 4-39, com os valores dos bloqueios da tabela 4-7 e dos tráfegos da tabela 4-9, foram obtidas as capacidades apresentadas na tabela seguinte.

		Destinos				
		2	3	4	5	6
Origens	1	47	20	570	46	68
	2	0	91	182	161	167
	3	0	0	15	25	29
	4	0	0	0	214	151
	5	0	0	0	0	464

Tab. 4-10 – Capacidades, para cada feixe da rede, calculadas pela inversão da fórmula de Erlang-B.

Uma justificação possível para o facto de em alguns dos feixes, termos obtido valores de capacidades inferiores aos da rede original resulta de a rede original estar sobredimensionada em alguns feixes para possivelmente suportarem encaminhamento alternativo. O que se fez então foi comparar os valores das capacidades da tabela anterior com os valores das capacidades originais e escolher o maior dos dois valores. Daqui resultaram as capacidades apresentadas na tabela seguinte.

		Destinos				
		2	3	4	5	6
Origens	1	47	24	570	48	68
	2	0	96	182	161	167
	3	0	0	15	48	29
	4	0	0	0	214	151
	5	0	0	0	0	464

Tab. 4-11 – Capacidades, para cada feixe da rede, efectivamente a utilizar na simulação.

Neste sistema são então oferecidos à rede, com as capacidades apresentadas na tabela 4-11, dois tipos de fluxos, sendo tráfego de Poisson em ambos os tipos.

Para determinar as médias de tráfego exógeno para os fluxos da classe 2,  $A^{(2)}(f^{(2)})$ , são necessários os valores para as cargas de tráfego (global) de circuitos oferecidas aos feixes,  $A_L(l)$ , quando são oferecido à rede os dois tipos de fluxos. Esses valores são apresentados na tabela seguinte e foram obtidos pela expressão 4-32, usando os valores da tabela 4-2.

		Destinos				
		2	3	4	5	6
Origens	1	38,458	9,758	360,934	28,658	40,754
	2	0	35,154	142,254	107,492	115,584
	3	0	0	16,688	9,604	18,55
	4	0	0	0	111,188	116,2
	5	0	0	0	0	177,954

Tab. 4-12 – Cargas de tráfego (global) de circuitos oferecidas aos feixes com os dois tipos de fluxos.

As médias de tráfego exógeno para cada um dos fluxos da classe 1,  $A^{(1)}(f^{(1)})$ , e da classe 2,  $A^{(2)}(f^{(2)})$ , encontram-se respectivamente na tabela 4-2 e 4-13. Os valores da tabela 4-13 foram obtidos pela expressão 4-33.

		Destinos				
		2	3	4	5	6
Origens	1	5,494	1,394	51,562	4,094	5,822
	2	0	5,022	20,322	15,356	16,512
	3	0	0	2,384	1,372	2,650
	4	0	0	0	15,884	16,600
	5	0	0	0	0	25,422

Tab. 4-13 – Cargas de tráfego exógeno para as chamadas do tipo de fluxo 2.

### 4.2.4 Sistema 3

Consideremos ainda a rede de comutação por circuitos com a topologia anterior em que o tráfego oferecido resulta de uma mistura de dois tipos de fluxos de Poisson (como no Sistema 2) e também, em

algumas centrais, de dois tipos de fluxos de tráfego de população finita. A largura de banda associada às chamadas de cada classe é diferente para cada um dos tipos de fluxos de tráfego. Temos neste sistema classes de chamadas de banda larga e classes de chamadas de banda estreita. Como temos 4 tipos de classes de chamadas então  $|K| = 4$ .

**Fluxo tipo 1 (64Kbits/s), tipo Poisson:**

O número de circuitos necessários às chamadas desta classe é  $C^{(1)} = 1$ .

Como o Fluxo 1 definido no Sistema 2, com intensidade de chegadas  $\lambda^{(1)}$  e tempo de serviço médio  $1/\mu^{(1)}$ . Com matriz de tráfego  $[A_{ij}^{(1)}] = [A_{ij}]$ .

**Fluxo tipo 2 (128Kbits/s), tipo Poisson:**

O número de circuitos necessários às chamadas desta classe é  $C^{(2)} = 2$ .

Como o Fluxo 2 definido no Sistema 2, com intensidade de chegadas  $\lambda^{(2)}$  e tempo de serviço médio  $1/\mu^{(2)}$ . Com matriz de tráfego  $[A_{ij}^{(2)}]$ .

**Fluxo tipo 3 (384Kbits/s), tipo Engset:**

O número de fontes é calculado pelo método da secção 4.2.4.3 e o número de circuitos necessários às chamadas desta classe é  $C^{(3)} = 6$ .

Este tipo de fluxo de tráfego tem uma intensidade de chegadas  $\gamma^{(3)}$  chamadas/segundo por fonte livre e cada chamada utiliza 6 circuitos (tráfego multicanal), para ser servida durante um intervalo de tempo que é uma variável aleatória com distribuição exponencial negativa (DEN) de média  $1/\mu^{(3)}$  segundos. Os circuitos são necessários durante todo o tempo de serviço da chamada, em todos os feixes ao longo do seu caminho.

Este fluxo é gerado de acordo com a matriz de tráfego  $[A_{ij}^{(3)}]$ .

**Fluxo tipo 4 (1536Kbits/s), tipo Engset:**

O número de fontes é calculado pelo método da secção 4.2.4.3 e o número de circuitos necessários às chamadas desta classe é  $C^{(4)} = 24$ .

Este tipo de fluxo de tráfego tem uma intensidade de chegadas  $\gamma^{(4)}$  chamadas/segundo por fonte livre e cada chamada utiliza 24 circuitos, para ser servida durante um intervalo de tempo que é uma variável aleatória com DEN de média  $1/\mu^{(4)}$  segundos. Os circuitos são necessários durante todo o tempo de serviço da chamada, em todos os feixes ao longo do seu caminho.

Este fluxo é gerado de acordo com a matriz de tráfego  $[A_{ij}^{(4)}]$ .

Será necessário, também neste caso, fazer um redimensionamento da rede pois além do tráfego para o qual foi planeada estão-se agora a oferecer outros tráfegos.

#### 4.2.4.1 Novo redimensionamento grosseiro da rede original

Neste ponto torna-se necessário redimensionar a rede original dado pretendermos oferecer à rede além do tipo de fluxo de tráfego original outros tipos de fluxos.

Também aqui, como no redimensionamento anterior, vamos determinar o novo número de circuitos de cada feixe da rede tentando obter uma rede com desempenho semelhante ao obtido na rede original. Para o conseguir vamos também tentar aproximar os bloqueios médios de circuitos nos feixes para a rede com os quatro tipos de fluxos (rede actual) aos bloqueios médios de circuitos nos feixes para a rede com apenas um tipo de fluxo (rede original), quando é usado um método de encaminhamento que permite às chamadas tentarem caminhos alternativos.

Temos neste sistema 4 tipos de fluxos logo 4 classes de chamadas. Considera-se que a classe de chamadas  $k$  corresponde ao fluxo de tipo  $k$ .

Sendo conhecida a matriz  $[A_{ij}^{(1)}]$ , e sabendo que a repartição do tráfego total de chamadas por cada classe é:

Classe de chamadas	Nº de circuitos	% do tráfego de chamadas
1	1	75%
2	2	15%
3	6	9%
4	24	1%

Tab. 4.14 - % do tráfego (de chamadas) oferecido por cada uma das classes de chamadas.

Vai supor-se que quase todo o tráfego, dos vários tipos de fluxos, é encaminhado directamente. Com esta consideração e com a informação anterior é possível calcular a média do tráfego (global) de circuitos oferecido ao feixe  $l$ ,  $A_L(l)$ .

Da suposição de que todo o tráfego é encaminhado directamente temos que:

$$A_L(l) \cong \sum_{k=1}^4 A^{(k)}(f^{(k)}) \times C^{(k)} : \tag{4.42}$$

$$f^{(k)} = (v_i, v_j, k) \in F^{(k)}, l = \{v_i, v_j\}, \quad \forall l \in L$$

Como anteriormente, é necessário para os cálculos o tráfego de circuitos e não o de chamadas, pelo que é necessário converter os valores de percentagem de tráfego de chamadas da classe  $k$  ( $p^k$ ), na percentagem correspondente de circuitos ocupados pelas chamadas desta classe ( $p_L^k$ ).

Através das equações 4-42 e 4-30 podemos calcular as percentagens de circuitos ocupados pelas chamadas da classe  $k$  ( $p_L^k$ ):

$$p_L^k = \frac{A_L^{(k)}(l)}{A_L(l)} = \frac{p^k \times C^{(k)}}{\sum_{k=1}^4 p^k \times C^{(k)}} \quad (4.43)$$

Classe de chamadas	Nº de circuitos	% do tráfego de circuitos
1	1	40,98%
2	2	16,39%
3	6	29,51%
4	24	13,12%

Tab. 4-15 - % do tráfego (de circuitos) oferecido por cada uma das classes de chamadas.

Considerando a expressão 4-42 juntamente com a informação obtida da tabela anterior, podemos calcular  $A_L(l)$  a partir de:

$$A_L(l) = \frac{A^{(1)}(f^{(1)})}{0.4098} : f^{(1)} = (v_i, v_j, 1) \in F^{(1)}, l = \{v_i, v_j\} \quad \forall l \in L \quad (4.44)$$

onde  $A^{(1)}(f^{(1)})$  é a média do tráfego exógeno do fluxo  $f^{(1)}$  da classe 1, que é conhecida, pois é conhecida a matriz  $[A_{ij}^{(1)}]$ .

Conhecendo  $A_L(l)$  podemos calcular  $A^{(2)}(f^{(2)})$  a partir de:

$$A^{(2)}(f^{(2)}) = \frac{0.1639}{2} A_L(l); \quad \forall f^{(2)} = (v_i, v_j, 2) \in F^{(2)} : l = \{v_i, v_j\} \in L \quad (4.45)$$

e  $A^{(3)}(f^{(3)})$  a partir de:

$$A^{(3)}(f^{(3)}) = \frac{0.2951}{6} A_L(l); \quad \forall f^{(3)} = (v_i, v_j, 3) \in F^{(3)} : l = \{v_i, v_j\} \in L \quad (4.46)$$

e ainda  $A^{(4)}(f^{(4)})$  a partir de:

$$A^{(4)}(f^{(4)}) = \frac{0.1312}{24} A_L(l); \quad \forall f^{(4)} = (v_i, v_j, 4) \in F^{(4)} : l = \{v_i, v_j\} \in L \quad (4.47)$$

Pois  $C^{(1)} = 1$ ,  $C^{(2)} = 2$ ,  $C^{(3)} = 6$  e  $C^{(4)} = 24$ .

Assumindo que são conhecidos, para a rede original (apenas com tráfego  $A^{(1)}(f^{(1)})$ ), os bloqueios médios de chamadas nos feixes  $B_L^{\text{orig}}(l)$ , que são iguais aos bloqueios de circuitos, vamos calcular os bloqueios médios de circuitos nos feixes para a rede actual  $B'_L(l)$ , como já se fez no Sistema 2, através da expressão 4-34 e ajustar o tráfego através da expressão 4-35.

Utilizando a inversão da fórmula de Erlang-B (apresentada anteriormente), determinamos o número de circuitos para cada feixe. Temos então a rede original redimensionada novamente.

#### 4.2.4.2 Resultados obtidos para o novo redimensionamento

Os bloqueios médios de chamadas nos feixes, para a rede original,  $B_L^{\text{orig}}(l)$ , usados, foram obtidos por simulação, e são os mesmos que foram utilizados no redimensionamento anterior, ver tabela 4-6. Lembremo-nos que esses bloqueios foram obtidos quando foi oferecido à rede original tráfego ampliado, com os valores apresentados na tabela 4-5, e que esses tráfegos vão corresponder às médias de tráfego exógeno ampliado para cada um dos fluxos da classe 1,  $A^{(1)}(f^{(1)})$ .

Os valores que pretendemos para os bloqueios médios de circuitos nos feixes (equivalente de chamadas monocanal),  $B'_{1L}(l)$ , para a rede com os quatro tipos de fluxos, são os mesmos que os utilizados para a rede com dois tipos de fluxos. Por isso a aproximação utilizada foi também a aproximação indicada na expressão 4-34, pela qual se obtiveram de novo os valores apresentados na tabela 4-7.

Para determinar os valores para as cargas de tráfego moderado (global) de circuitos oferecidas aos feixes,  $A_L(l)$ , quando são oferecidos à rede quatro tipos de fluxos, vamos ter que tomar em atenção, para cada feixe, quais as de classes de chamadas que ele serve. Vamos considerar que neste sistema apenas é gerado tráfego da classe 3 e da classe 4 na central 1, e que os tráfegos das restantes classes são gerados em todas as centrais da rede. Considerando só encaminhamento directo, os feixes entre a central 1 e qualquer outra central da rede são utilizados por chamadas de qualquer uma das quatro classes. Todos os feixes restantes são utilizados apenas por chamadas da classe 1 e da classe 2.

A repartição do tráfego total de chamadas por cada classe é:

Classe de chamadas	Nº de circuitos	% do tráfego com 4 classes	% do tráfego com 2 classes
1	1	75%	83,33%
2	2	15%	16,67%
3	6	9%	
4	24	1%	

Tab. 4-16 - % do tráfego (de chamadas) oferecido por cada uma das classes de chamadas quando são consideradas 4 classes ou apenas 2 classes.

De acordo com as fórmulas apresentadas anteriormente, podemos obter a tabela seguinte, correspondente à repartição do tráfego total de circuitos por cada classe.

Classe de chamadas	Nº de circuitos	% do tráfego com 4 classes	% do tráfego com 2 classes
1	1	40,98%	71,43%
2	2	16,39%	28,57%
3	6	29,51%	
4	24	13,12%	

Tab. 4-17 - % do tráfego (de circuitos) oferecido por cada uma das classes de chamadas quando são consideradas 4 classes ou apenas 2 classes.

De acordo com o referido anteriormente e considerando a repartição de tráfego apresentada na tabela anterior vão-se determinar os valores para as cargas de tráfego moderado (global) de circuitos oferecidas aos feixes. Esses valores são apresentados na tabela 4-18. Os valores da primeira linha foram obtidos pela expressão 4-44, enquanto que os restantes valores foram obtidos pela expressão 4-32, sendo usados no cálculo de ambas as expressões os valores da tabela 4-5.

		Destinos				
		2	3	4	5	6
Origens	1	100,540	25,510	943,585	74,920	106,543
	2	0	52,731	213,381	161,238	173,376
	3	0	0	25,032	14,406	27,825
	4	0	0	0	166,782	174,300
	5	0	0	0	0	266,931

Tab. 4-18 – Cargas de tráfego ampliado (global) de circuitos oferecidas aos feixes, com os quatro tipos de fluxos em alguns.

Vamos também aqui ajustar os tráfegos pela aproximação indicada na expressão 4-35, passando os tráfegos ampliados (globais) de circuitos oferecidos aos feixes,  $A_L(l)$ , a ser  $A'_L(l)$  cujos valores são apresentados na tabela seguinte.

		Destinos				
		2	3	4	5	6
Origens	1	120,648	30,612	1132,302	89,904	127,851
	2	0	63,277	256,057	193,486	208,051
	3	0	0	30,038	17,287	33,390
	4	0	0	0	200,138	209,160
	5	0	0	0	0	320,317

Tab. 4-19 – Cargas de tráfego ampliado (global) de circuitos oferecidas aos feixes, ajustadas, para a rede com 4 tipos de fluxos.

Usando, de novo, a iteradora baseada no método de Newton Raphson, expressão 4-39, com os valores dos bloqueios da tabela 4-7 e dos tráfegos da tabela 4-19, foram obtidas as capacidades apresentadas na tabela seguinte.

		Destinos				
		2	3	4	5	6
Origens	1	81	32	988	78	115
	2	0	91	182	161	167
	3	0	0	15	25	29
	4	0	0	0	214	151
	5	0	0	0	0	464

Tab. 4-20 – Capacidades, para cada feixe da rede, calculadas pela inversão da fórmula de Erlang-B, considerando 4 tipos de fluxos.

Pela mesma razão que a apresentada no caso da rede com apenas 2 tipos de fluxos, também aqui o valor da capacidade de cada feixe, a usar na simulação, vai ser o maior entre o valor respectivo da tabela anterior e o valor da capacidade original. Os valores que daí resultam encontram-se na tabela seguinte.

		Destinos				
		2	3	4	5	6
Origens	1	81	32	988	78	115
	2	0	96	182	161	167
	3	0	0	15	48	29
	4	0	0	0	214	151
	5	0	0	0	0	464

Tab. 4-21 – Capacidades, para cada feixe da rede, efectivamente a utilizar na simulação, quando são considerados 4 tipos de fluxos.

Neste sistema são, como já referido, oferecidos à rede, com as capacidades apresentadas na tabela anterior, dois tipos de fluxos, sendo tráfego exógeno de Poisson em ambos os tipos. E, além destes dois tipos de fluxos são ainda oferecidos mais dois tipos de fluxos, mas apenas à central 1. É preciso determinar as médias de tráfego exógeno para os fluxos de cada uma das classes.

Lembre que se considerou que as médias do tráfego exógeno para cada um dos fluxos da classe 1,  $A^{(1)}(f^{(1)})$ , tanto neste redimensionamento como no anterior, serem iguais aos valores das médias do tráfego exógeno para a rede original,  $A(f)$ .

Falta apenas determinar as médias de tráfego exógeno para os fluxos das classes 2, 3 e 4,  $A^{(2)}(f^{(2)})$ ,  $A^{(3)}(f^{(3)})$  e  $A^{(4)}(f^{(4)})$  respectivamente. Para tal são necessários os valores para as cargas de tráfego (global) de circuitos oferecidas aos feixes,  $A_L(l)$ , quando são oferecidos à rede os quatro tipos de fluxos. Esses valores são apresentados na tabela seguinte e foram obtidos pelas expressões 4-44 e 4-32, usando os valores da tabela 4-2 no cálculo de ambas as expressões. Os valores da primeira linha foram obtidos pela expressão 4-44, todos os outros foram obtidos pela expressão 4-32.

		Destinos				
		2	3	4	5	6
Origens	1	67,027	17,007	629,056	49,947	71,028
	2	0	35,154	142,254	107,492	115,584
	3	0	0	16,688	9,604	18,550
	4	0	0	0	111,188	116,200
	5	0	0	0	0	177,954

Tab. 4-22 – Cargas de tráfego (global) de circuitos oferecidas aos feixes, com os quatro tipos de fluxos em alguns feixes.

As médias de tráfego exógeno para cada um dos fluxos da classe 1 ( $A^{(1)}(f^{(1)})$ ), 2 ( $A^{(2)}(f^{(2)})$ ), 3 ( $A^{(3)}(f^{(3)})$ ) e 4 ( $A^{(4)}(f^{(4)})$ ) encontram-se respectivamente na tabela 4-2, 4-23, 4-24 e 4-25.

Os valores da tabela 4-23 foram obtidos pelas expressões 4-45 e 4-33, usando os valores da tabela 4-2 no cálculo de ambas. Os valores da primeira linha foram obtidos pela expressão 4-45, todos os outros foram obtidos pela expressão 4-33. Repare que os valores desta tabela são os mesmos que os da tabela 4-13 o que era de esperar pois  $A^{(2)}(f^{(2)})$  é uma proporção de  $A^{(1)}(f^{(1)})$  e o valor da proporção é igual quando são consideradas 2 ou 4 classes.

		Destinos				
		2	3	4	5	6
Origens	1	5,494	1,394	51,562	4,094	5,822
	2	0	5,022	20,322	15,356	16,512
	3	0	0	2,384	1,372	2,650
	4	0	0	0	15,884	16,600
	5	0	0	0	0	25,422

Tab. 4-23 – Cargas de tráfego exógeno para as chamadas do tipo de fluxo 2.

Os valores da primeira linha da tabela 4-24, foram obtidos pela expressão 4-46, tendo sido usados no cálculo os valores da tabela 4-2. Os restantes valores são nulos pelo facto de se considerar que o tráfego deste tipo é gerado apenas na central 1.

		Destinos				
		2	3	4	5	6
Origens	1	3,296	0,836	30,937	2,456	3,493
	2	0	0	0	0	0
	3	0	0	0	0	0
	4	0	0	0	0	0
	5	0	0	0	0	0

Tab. 4-24 – Cargas de tráfego exógeno para as chamadas do tipo de fluxo 3.

Os valores da primeira linha da tabela 4-25 foram obtidos pela expressão 4-47, tendo sido usados no cálculo os valores da tabela 4-2. Os restantes valores são nulos pela razão apresentada para a tabela anterior.

		Destinos				
		2	3	4	5	6
Origens	1	0,366	0,093	3,437	0,273	0,388
	2	0	0	0	0	0
	3	0	0	0	0	0
	4	0	0	0	0	0
	5	0	0	0	0	0

Tab. 4-25 – Cargas de tráfego exógeno para as chamadas do tipo de fluxo 4.

#### 4.2.4.3 Cálculo do número de fontes e das intensidades de chamadas por fonte livre para os tipos de fluxos de tráfego $k$ gerados por populações finitas

O número de fontes que geram chamadas da classe  $k$  na central  $v$ ,  $F^{(k)}(v)$ , vão ser determinadas a partir de:

$$F^{(k)}(v) \times C^{(k)} = 2 \max_{l \in LV(v)} \{C(l)\} \quad (4.48)$$

Para determinar a intensidade de chegada de chamadas por fonte livre da classe  $k$  na central  $v$ ,  $\gamma^{(k)}(v)$ , é necessário conhecer o tráfego de chamadas por fonte livre da classe  $k$  na central  $v$ ,  $\beta^{(k)}(v)$ , pois sabemos que:

$$\beta^{(k)}(v) = \frac{\gamma^{(k)}(v)}{\mu^{(k)}} \quad (4.49)$$

Os  $\beta^{(k)}(v)$  podem ser determinados a partir de:

$$AV^{(k)}(v) = \beta^{(k)}(v) \times F^{(k)}(v) \tag{4.50}$$

Tomamos como primeira aproximação:

$$AV^{(k)}(v) \approx \sum_{u \in \mathcal{V}\mathcal{V}(v)} A^{(k)}(f^{(k)}), \quad f^{(k)} = (v, u, k) \in F^{(k)} \tag{4.51}$$

notando que o tráfego gerado pelas fontes livres ( $AV^{(k)}(v)$ ) é em geral superior ao tráfego médio gerado pelas fontes, sendo os seus valores tanto mais próximos quanto menor for o bloqueio de chamadas experimentado pelos fluxos correspondentes.

Além dos valores dos  $\gamma^{(k)}(v)$  é necessário saber as percentagens do tráfego gerado na central  $v$  com destino a cada uma das suas centrais adjacentes. Estas percentagens são definidas do seguinte modo:

$$\alpha_i = \frac{A^{(k)}(f^{(k)})}{AV^{(k)}(v)}, \tag{4.52}$$

$$f^{(k)} = (v, u_i, k) \in F^{(k)}, \quad u_i \in \mathcal{V}\mathcal{V}(v), \quad \forall i = 1, 2, \dots, |\mathcal{V}\mathcal{V}(v)|$$

#### 4.2.4.4 Valores obtidos para os parâmetros $F^{(k)}(v)$ , $\gamma^{(k)}(v)$ e $\alpha_i$

A tabela 4-26 apresenta o número de fontes que geram chamadas das classes 3 e 4 em cada central. O número de fontes que geram chamadas das classes 3 e 4 na central 1 foram determinadas a partir da expressão 4-48.

		Centrais					
		1	2	3	4	5	6
Classes	3	329	0	0	0	0	0
	4	82	0	0	0	0	0

Tab. 4-26 – Número de fontes, por classe de chamada, em cada central em que é gerado tráfego de população finita.

O tráfego de chamadas gerado pelas fontes livres ( $AV^{(k)}(v)$ ), para cada uma das classes, é apresentado na tabela 4-27. O tráfego médio de chamadas gerado pelas fontes livres para a classe 3 e 4 na central 1 foi determinado a partir aproximação dada pela expressão 4-51.

		Centrais					
		1	2	3	4	5	6
Classes	3	41,020	0	0	0	0	0
	4	4,558	0	0	0	0	0

Tab. 4-27 – Tráfego de chamadas gerado pelas fontes livres, por cada uma das classe de chamada.

O tráfego médio de chamadas por fonte livre ( $\beta^{(k)}(v)$ ), para cada uma das classes, é apresentado na tabela 4-28. O tráfego médio de chamadas por fonte livre para a classe 3 e 4 na central 1 foi determinado a partir da expressão 4-50.

		Centrais					
		1	2	3	4	5	6
Classes	3	0,125	0	0	0	0	0
	4	0,056	0	0	0	0	0

Tab. 4-28 – Tráfego de chamadas por fonte livre, para cada uma das classes de chamadas.

A intensidade de chegada de chamadas para cada classe, por fonte livre de cada classe em cada central ( $\gamma^{(k)}(v)$ ), é apresentada na tabela 4-29. A intensidade de chegada de chamadas para a classe 3 e 4, por fonte livre de cada classe na central 1 foi determinada a partir da expressão 4-49. Os valores desta tabela são os mesmos da tabela anterior por ter sido usado, na expressão 4-49, a unidade para o valor do tempo médio de serviço das chamadas de ambas as classes.

		Centrais					
		1	2	3	4	5	6
Classes	3	0,1247	0	0	0	0	0
	4	0,0556	0	0	0	0	0

Tab. 4-29 – Intensidade de chegada de chamadas por fonte livre.

As percentagens do tráfego gerado na central 1 com destino a cada uma das sua centrais adjacentes ( $\alpha_i$ ), para a classe 3 e 4 são apresentadas na tabela 4-30. Os seus valores foram obtidos a partir da expressão 4-52.

		Centrais				
		2	3	4	5	6
Classes	3	8,04%	2,04%	75,42%	5,99%	8,52%
	4	8,04%	2,04%	75,42%	5,99%	8,52%

Tab. 4-30 – Percentagem do tráfego gerado na central 1 para cada uma das outras centrais, por classe de chamada.



# Capítulo 5

## Descrição e Implementação do Simulador

### 5.1 Introdução

Sendo as redes inter-centrais com encaminhamento dinâmico demasiado complexas para serem estudadas analiticamente através de um modelo suficientemente rigoroso, nomeadamente em termos de desempenho dinâmico (tendo em conta a natureza estocástica dos fluxos de tráfego) recorreremos a um modelo simulacional para fazer o seu estudo.

Neste capítulo far-se-á uma revisão dos princípios e conceitos de simulação que estão na base do desenvolvimento de uma plataforma computacional para a análise de desempenho de redes inter-centrais com comutação por circuitos, com métodos de encaminhamento dinâmico. Por outro lado, apresentam-se as características relevantes da linguagem de simulação OMNeT++, que foi utilizada, bem como uma descrição dos aspectos essenciais e módulos do pacote de software desenvolvido neste âmbito.

### 5.2 Conceitos introdutórios sobre simulação

Uma referência geral para os conceitos de simulação aqui utilizados é [Law91].

Com a simulação pretende-se obter uma representação, tão realista quanto possível, da operação ao longo do tempo de um sistema real. É necessário para o efeito que seja gerada uma realização (função amostra) do comportamento do sistema. A partir da simulação são recolhidos dados como se se estivesse a observar o sistema real. Os dados gerados pela simulação são usados para estimar as medidas de desempenho do sistema.

Segue-se a definição de alguns termos para ajudar na compreensão e análise do sistema (como apresentado em [Banks96]).

Uma *entidade* é um objecto de interesse no sistema.

Um *atributo* é uma propriedade de uma entidade.

Um *recurso* é um objecto que fornece um serviço a uma entidade.

Uma *actividade* representa um período de tempo de comprimento especificado.

O *estado do sistema* é armazenado num conjunto de variáveis de estado do sistema. As variáveis de estado são as variáveis necessárias para descrever o sistema relativamente aos objectivos do estudo.

Um *acontecimento* é definido como uma ocorrência instantânea que pode alterar o estado do sistema.

Os sistemas podem ser discretos ou contínuos. Um sistema discreto é um sistema no qual as variáveis de estado mudam apenas num conjunto discreto de pontos no tempo. Um sistema contínuo é um sistema no qual as variáveis de estado mudam continuamente ao longo do tempo. Na prática poucos sistemas são totalmente discretos ou contínuos, mas como existe sempre um tipo de mudanças que predomina para muitos sistemas, será possível normalmente classificar um sistema como contínuo ou discreto [Law91].

Estudos do sistema são normalmente baseados na modelação do sistema. O modelo é definido como uma representação do sistema com o objectivo de o estudar sob uma determinada perspectiva. Os modelos são representados da mesma forma que foram representados os sistemas, sendo os seus componentes também as entidades, os atributos, as actividades e os recursos. O modelo deve ser suficientemente pormenorizado para permitir tirar conclusões válidas acerca do sistema real. No entanto só devem fazer parte do modelo os aspectos relevantes ao estudo.

Os modelos de simulação podem ser estáticos ou dinâmicos. Um modelo de simulação é estático quando representa o sistema num determinado instante. Um modelo de simulação é dinâmico quando representa a evolução do sistema ao longo do tempo.

Os modelos de simulação podem ser determinísticos ou estocásticos. Se o modelo de simulação não contém nenhum componente probabilístico (aleatório) diz-se que é um modelo determinístico. Se o modelo tiver algum componente aleatório diz-se que é um modelo estocástico. As saídas dos modelos estocásticos são saídas aleatórias e devem por isso ser consideradas como estimativas das verdadeiras características do sistema.

Os modelos podem ser discretos ou contínuos. A definição de modelo discreto e de modelo contínuo é a mesma que foi utilizada na definição de sistema discreto ou contínuo.

Um acontecimento é uma ocorrência instantânea que geralmente muda as variáveis de estado do modelo. No entanto, há situações em que tal não ocorre, como por exemplo o acontecimento que provoca o fim de uma corrida de simulação.

Na modelação de fenómenos do mundo real existem poucas situações onde as acções das entidades dentro do sistema possam ser completamente previsíveis. Para modelar as actividades do sistema, que são geralmente imprevisíveis e incertas, recorre-se a distribuições estatísticas e às variáveis aleatórias associadas. É necessário então determinar as distribuições apropriadas dos dados de entradas para cada simulação. Depois de saber quais as distribuições estatísticas a usar num determinado modelo é necessário saber quais os modos a usar para gerar amostras destas distribuições para serem usadas como entradas nesse modelo de simulação. Entradas típicas numa simulação de sistemas estocásticos de serviço são a distribuição do tempo entre chegadas e a distribuição do tempo de serviço.

Os números aleatórios são o ingrediente básico na simulação de quase todos os sistemas discretos. Os números aleatórios numa linguagem de simulação são usados para gerar os instantes em que ocorrem os acontecimentos e para gerar outras variáveis aleatórias.

Uma sequência de números aleatórios deve ter as propriedades de independência e uniformidade, ou seja (muito genericamente) cada um dos números da série deve ser independente dos restantes e a probabilidade de obter qualquer dos valores no intervalo definido para esses números deve ser igual. Os números aleatórios que são usados numa linguagem de simulação são designados por números pseudo-aleatórios por serem gerados a partir de um método conhecido e reproduzível.

Na análise das saídas de uma simulação faz-se a análise dos dados gerados na simulação, com o objectivo de estimar o desempenho do sistema modelizado.

## 5.2.1 Simulação por acontecimentos discretos

A simulação por acontecimentos discretos está relacionada com a modelação da evolução do sistema ao longo do tempo por uma representação na qual as variáveis de estado mudam de valor, instantaneamente, em pontos separados no tempo. Os modelos de simulação por acontecimentos discretos têm uma forma característica de representação do tempo através de um relógio de simulação. O relógio de simulação é a designação para a variável que no modelo de simulação dá o valor corrente do tempo simulado (também designado por tempo de simulação). O avanço do relógio de simulação pode ser de um dos dois tipos seguintes:

- Para o próximo acontecimento;
- Incrementos fixos – avanço do tempo simulado por incrementos constantes.

O avanço utilizado mais frequentemente (quase exclusivamente) é o avanço do relógio de simulação para o próximo acontecimento. Neste caso o relógio de simulação avança de acontecimento em acontecimento sendo actualizado o estado do sistema em cada avanço. Este processo continua até que seja satisfeita uma condição de paragem previamente especificada.

Neste tipo de simulação existe uma lista de acontecimentos, também designada por lista de acontecimentos futuros, que é uma lista de notificações para acontecimentos futuros, ordenada pelo tempo de ocorrência desses acontecimentos.

## 5.2.2 Outros tipos de simulação

A simulação contínua relaciona-se com a modelação de um sistema, ao longo do tempo, por uma representação na qual as variáveis de estado mudam continuamente em função do tempo.

A simulação combinada contínua - discreta é usada em sistemas que não são completamente discretos nem completamente contínuos e nos quais haja necessidade de construir um modelo com aspectos de simulação contínua e também de simulação por acontecimentos discretos.

A simulação de Monte Carlo<sup>2</sup> segundo Law e Kelton [Law91] é um esquema que utiliza números aleatórios e que é usado para resolver certos problemas estocásticos ou determinísticos onde a representação explícita da passagem do tempo não tem grande importância. Daí que as simulações de Monte Carlo sejam geralmente estáticas e não dinâmicas.

É importante notar que um modelo de simulação discreto nem sempre é usado para modelar um sistema discreto, assim como um modelo de simulação contínuo nem sempre é usado para modelar um sistema contínuo. A escolha do tipo de simulação a usar é uma função das características do sistema e dos objectivos do estudo.

## 5.2.3 Tipos de software de simulação para simular redes de comunicação

Podem considerar-se três tipos principais de software de simulação para redes de comunicação [Law94]:

1. Linguagens de simulação de âmbito geral: pacote de simulação que pode ser utilizado em sistemas de diversas áreas e tipos, mas que tem características especiais viradas para redes de comunicação. A maior vantagem de muitas destas linguagens é a sua capacidade para modelar quase todos os tipos de redes de comunicação.
2. Linguagens de simulação orientada para comunicações: pacote de simulação orientada especificamente para o estudo de redes de comunicações.
3. Simulador orientado para comunicações: pacote de simulação que permite simular uma rede numa classe específica de redes de comunicação, sem necessidade de programação. É seleccionada uma

---

<sup>2</sup> É frequente encontrar autores que definem simulação de Monte Carlo qualquer simulação que envolva o uso de números aleatórios.

rede particular para simulação (dentro do domínio do pacote) escolhendo itens dos menus, preenchendo caixas de diálogo e pelo uso de gráficos. Um exemplo de um simulador orientado para comunicações é o COMNET III.

As linguagens de simulação usam uma das duas aproximações básicas seguintes:

- Aproximação orientada para o acontecimento;
- Aproximação orientada para o processo.

Para modelar o sistema, na aproximação orientada para o acontecimento, começa-se por identificar os acontecimentos característicos e depois escreve-se um conjunto de rotinas de acontecimentos que dão uma descrição detalhada das mudanças de estado que ocorrem na altura de cada acontecimento. A simulação evolui ao longo do tempo pela execução de acontecimentos (rotinas de acontecimento), ordenados por ordem crescente do tempo da sua ocorrência. Uma propriedade básica de uma rotina de acontecimentos é que não há passagem do tempo simulado durante a sua execução.

Na aproximação orientada para o processo, este é entendido como uma sequência ordenada pelo tempo de acontecimentos inter-relacionados, separados pela passagem do tempo, os quais descrevem a experiência completa de uma entidade, enquanto ela flui ao longo do sistema. Um sistema ou modelo de simulação pode ter vários tipos diferentes de processos. Existe uma rotina, correspondente a cada processo no modelo, que descreve a história completa da sua entidade, e a forma como ela se altera através do processo correspondente.

## **5.3 Breve descrição da linguagem de simulação utilizada - OMNeT++**

O OMNeT++ é uma linguagem de simulação por acontecimentos discreta com aproximação ao processo.

O OMNeT++ assim como o OPNET são linguagens de simulação de âmbito geral (de acordo com a classificação anterior do software de simulação para redes de comunicação). Possuem um editor gráfico que pode ser usado para especificar graficamente a topologia da rede.

### **5.3.1 Abordagem geral ao OMNeT++**

De acordo com a filosofia de modelação que o OMNeT++ implementa, um sistema consiste em várias entidades designadas por módulos. Um modelo em OMNeT++ consiste em vários módulos, módulos esses que comunicam entre si pela passagem de mensagens. Estas mensagens podem conter estruturas de

dados complexas. Os módulos activos são designados por módulos simples. O código dos módulos simples executa-se quase em paralelo (os módulos simples são implementados como co-rotinas – processos concorrentes). O código dos módulos simples é escrito em C++ usando a biblioteca de classes do OMNeT++. Os módulos simples podem ser agrupados em módulos designados por módulos compostos.

Existe uma linguagem separada para definir a estrutura do modelo (linguagem NED - “**NE**twork **D**escription”). Em vez de usar a linguagem directamente pode usar-se um editor gráfico, GNED - “**G**raphical **N**etwork **E**Ditor”, que produz ficheiros NED a partir da representação gráfica. A descrição do modelo em linguagem NED é traduzida em C++ (usando o compilador NEDC) sendo esta adicionada ao código da simulação.

Num ficheiro de configuração, designado por ficheiro *ini* (por normalmente ter extensão *ini*) são inseridas opções que controlam como a simulação é executada e valores para os parâmetros do modelo. Este ficheiro é um ficheiro de texto, consistindo em entradas agrupadas em diferentes secções. Neste ficheiro podem ser seleccionadas as sementes para os geradores de números aleatórios. O OMNeT++ tem instrumentos para ajudar na escolha de boas sementes.

Para tornar uma simulação executável é necessário ligar o código produzido (código C++ correspondente à implementação dos módulos simples e à descrição do modelo) ao núcleo (*kernel*) do OMNeT++ e a um dos interfaces com o utilizador oferecido pelo OMNeT++.

Todas as corridas incluídas no ficheiro de configuração são usualmente executadas automaticamente umas após as outras. As opções que controlam como a simulação é executada e os valores para os parâmetros do modelo podem ser especificados globalmente ou individualmente para cada corrida. Os resultados da simulação são escritos em ficheiros de saída.

### **5.3.2 Linguagem utilizada para definir o modelo - Linguagem NED**

A descrição que vamos passar a fazer apenas tem como objectivo servir de base para a compreensão do modelo criado. Para uma descrição completa da linguagem consultar o Manual do Utilizador do OMNeT++ [Varga00]. Um exemplo de utilização da linguagem NED pode ser encontrado no apêndice A.

A topologia do modelo pode ser representada usando o editor gráfico GNED, resultando uma descrição do modelo em linguagem NED.

Um modelo OMNeT++, normalmente designado por rede, consiste em módulos embutidos hierarquicamente. O módulo no nível topo é designado por módulo de sistema que pode conter

submódulos os quais por sua vez podem conter outros submódulos. Os módulos que possuem submódulos são designados por módulos compostos, e os módulos que estão no nível mais baixo da hierarquia são designados por módulos simples. Não existindo limite para a profundidade dos módulos, é permitido ao utilizador reflectir a estrutura lógica do sistema na estrutura do modelo. Os módulos comunicam com a passagem de mensagens, mensagens estas que podem conter estruturas de dados complexas. Os módulos podem enviar mensagens directamente para os destinos ou por um caminho predefinido, 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 são usados principalmente para três fins:

- Personalizar o comportamento dos módulos;
- Criar topologias flexíveis para o modelo;
- Para comunicação entre os módulos, como variáveis partilhadas.

Os parâmetros podem ser usados para construir topologias num modo flexível. Assim, dentro de um módulo composto, os parâmetros podem definir o número de submódulos, o número de portas e o modo como as ligações internas são feitas. Os módulos compostos podem passar parâmetros para os seus submódulos.

Os módulos simples são definidos declarando os seus parâmetros e as suas portas. Os parâmetros ficam com valores reais apenas quando os módulos simples forem utilizados na construção de um módulo composto ou de um módulo de sistema. É permitido utilizar vectores de portas, podendo cada um conter um determinado número de portas simples. O número de portas simples de um vector é dado quando este for utilizado na construção de um módulo composto.

Os módulos compostos são definidos de forma análoga aos módulos simples excepto que estes também têm secções para especificar os submódulos e as ligações dentro do módulo. Os parâmetros dos módulos compostos podem ser utilizados para definir a topologia interna da rede. Existem secções nos módulos compostos para definir para cada submódulo os valores reais dos seus parâmetros e o número de portas simples de cada vector de portas. Nos módulos compostos é também possível criar um vector de submódulos.

Na definição de um módulo composto são feitas as ligações entre as portas do módulo composto e as portas dos submódulos imediatos e também as ligações entre as portas dos seus submódulos. Uma porta não pode ser usada em mais do que uma ligação e uma ligação não pode ser feita mais do que uma vez.

Às ligações podem estar associados parâmetros sendo um deles o atraso na propagação (tempo que a mensagem gasta a percorrer a ligação).

A definição da rede (ou definição do módulo de sistema) especifica o módulo sistema. O modelo fica completo quando se define o módulo de sistema. Exemplos das definições de todos os tipos de módulos podem ser obtidos no capítulo 4 do Manual do Utilizador do OMNeT++ [Varga00].

### 5.3.3 Desenvolvimento dos módulos simples – Programação dos algoritmos.

Os módulos do nível hierárquico mais baixo são implementados pelo utilizador, e contêm os algoritmos do modelo. Durante a execução de uma simulação, os módulos simples parecem correr em paralelo, pois são implementados como co-rotinas (por vezes designadas como processos). Os módulos simples são implementados em C++.

Todos os elementos da simulação (mensagens, módulos, parâmetros, etc) são representados por objectos. Alguns exemplos de classes da biblioteca das classes do OMNeT++ são: *cModule*, *cGate*, *cPar*, *cMessage*.

Cada módulo simples é implementado com uma classe C++. As classes dos módulos simples são derivadas de uma classe base (*cSimpleModule*), redefinindo uma função virtual designada por *activity* com o algoritmo do processo. Uma outra função membro da classe *cModule* é a função *finish()* que é executada quando uma simulação termina. A função *finish()* também pode ser redefinida nas classes dos módulos simples. É possível acrescentar outras funções membro à classe ou mesmo membros de dados (variáveis internas).

Os parâmetros dos módulos simples podem ser acessíveis pelos seus algoritmos.

As funções e objectos mais importantes que podem ser usados na implementação dos módulos simples encontram-se referidos sinteticamente no apêndice B. Para compreensão do código implementado é necessário o conhecimento destes componentes. Para uma descrição mais pormenorizada deve ser consultado o apêndice D do Manual do Utilizador do OMNeT++ [Varga00].

### 5.3.4 Geração de números aleatórios

O gerador de números aleatórios utilizado no OMNeT++ é um gerador congruencial linear (“Linear Congruential Generator” - LCG) com um comprimento cíclico de  $2^{31} - 2$ .

Se um programa de simulação usar números aleatórios para mais do que um fim, então esses números deverão vir de geradores de números aleatórios diferentes. O OMNeT++ tem vários geradores de números aleatórios independentes.

Para evitar correlações indesejadas é importante que sejam usadas séries não sobrepostas de números aleatórios nas várias corridas de simulação e nas várias fontes de números aleatórios dentro de uma corrida de simulação. Por esta razão, os geradores devem começar com sementes bem separadas.

### 5.3.5 Compilar e executar uma simulação

Para executar uma simulação é necessário ligar o código produzido (código C++ correspondente à implementação dos módulos simples e à descrição do modelo) ao núcleo (*kernel*) do OMNeT++ e a um dos interfaces com o utilizador oferecido pelo OMNeT++. Os interfaces com o utilizador oferecidos pelo OMNeT++ são o Cmdenv, o Tvenv e o Tkenv os quais vão ser descritos na próxima secção.

Um modelo OMNeT++ consiste fisicamente nas seguintes partes:

- Na descrição da topologia na linguagem NED. Ficheiro com extensão *ned*;
- Nos módulos simples. Aos módulos simples correspondem ficheiros C++, onde se definem as operações dos componentes do modelo.

Os ficheiros NED são compilados em C++ usando o compilador NEDC que faz parte do OMNeT++.

O sistema de simulação fornece os seguintes componentes que fazem parte de qualquer simulação executável:

- O núcleo de simulação com a biblioteca de classes de simulação;
- Interface com o utilizador.

Os programas de simulação são construídos a partir dos componentes anteriores. Primeiro os ficheiros NED são compilados em código fonte C++ usando o compilador NEDC. Depois todos os ficheiros fonte C++ (implementação dos módulos simples e descrição da topologia) são compilados e ligados ao núcleo de simulação e ao interface do utilizador para construir uma simulação executável.

Como já foi referido, devem ser inseridas, num ficheiro de configuração designado por ficheiro *ini*, as opções que controlam como a simulação é executada e valores para os parâmetros do modelo. É sobre esse ficheiro que se vai falar mais em pormenor já a seguir.

#### 5.3.5.1 Ficheiro de configuração

Os ficheiros de configuração (também chamados ficheiros *ini*, por terem normalmente a extensão *ini*) contêm opções que controlam como a simulação é executada e podem também conter inicializações dos

parâmetros do modelo. Neste ficheiro podem ser seleccionadas as sementes para os geradores de números aleatórios (o OMNeT++ tem instrumentos para ajudar na escolha de boas sementes). O ficheiro *ini* é um ficheiro de texto que consiste em várias entradas agrupadas em várias secções. As secções que podem existir são as seguintes:

1. Se se pretender fazer inicializações comuns a todas as corridas<sup>3</sup>:
  - [*General*] – inicializações gerais;
  - [*Cmdenv*], [*Tvenv*], [*Tkenv*], . . . – inicializações específicas do interface com o utilizador;
  - [*Parameters*] – valores dos parâmetros dos módulos;
2. Se se pretender fazer inicializações para cada corrida individualmente:
  - [*Run 1*], [*Run 2*], [*Run 3*], . . . - inicializações específicas do interface com o utilizador, valores dos parâmetros dos módulos ou mapeamento das máquinas lógico-físico.

Nestas secções, os caracteres # e ; são usados para colocar comentários. Os valores dos parâmetros dos módulos, caso não tenham sido inicializados previamente no ficheiro NED, são pesquisados no ficheiro *ini*. Caso não sejam encontrados são pedidos valores ao utilizador.

O OMNeT++ pode executar várias corridas de simulação automaticamente, em sequência. Se foram seleccionadas várias corridas em sucessão, as opções de inicialização e o valor dos parâmetros podem ser dados individualmente para cada corrida ou conjuntamente para todas as corridas, através do ficheiro *ini*.

### 5.3.6 Interfaces com o utilizador

Os interfaces com o utilizador do OMNeT++ são usados na execução de uma simulação. O seu principal fim é tornar o interior do modelo visível ao utilizador, iniciar ou parar a execução da simulação e, eventualmente permitir a intervenção do utilizador pela alteração de objectos e variáveis dentro do modelo. Isto é muito importante na fase de desenvolvimento e correcção do projecto de simulação.

No OMNeT++ o interface com o utilizador está separado do núcleo de simulação. Isto faz com que seja possível usar vários tipos de interfaces com o utilizador sem alterar o núcleo de simulação.

É também possível executar o mesmo modelo de simulação com diferentes interfaces com o utilizador sem qualquer mudança nos ficheiros do modelo. Desta forma, na fase de teste e correcção, pode usar-se um poderoso interface gráfico, e no final executar a simulação com um interface simples e rápido, que suporte a execução em lote.

---

<sup>3</sup> Para além das apresentadas, existem duas secções comuns que só têm significado na execução distribuída do simulador:

- [*Slaves*] – opções referentes a cada unidade lógica de processamento distribuído;
- [*Machines*] – mapeamento lógico-físico das máquinas, na execução distribuída, indicando quais processos vão correr em quais máquinas;

Os interfaces com o utilizador disponíveis no OMNeT++ são:

- Cmdenv: interface com o utilizador em linha de comando, para execução em lote;
- Tvenv: interacção textual, interface com o utilizador em janelas (DOS, Turbo Vision);
- Tkenv: interacção gráfica baseada no Tk<sup>4</sup>, interface em janelas (X-Window, Win95, WinNT, etc.).

Segue-se uma descrição mais detalhada acerca de cada um dos interfaces com o utilizador.

### **Cmdenv**

O interface com o utilizador em linha de comando é um interface pequeno, portátil e rápido que compila e executa em todas as plataformas (UNIX; DOS; ou WinNT). O interface Cmdenv foi projectado principalmente para execuções em lotes.

O Cmdenv executa todas as corridas de simulação que forem descritas no ficheiro de configuração.

### **Tvenv<sup>5</sup>**

No interface com o utilizador Tvenv a interacção é textual, o interface com o utilizador em janelas suporta a execução interactiva da simulação. Tvenv é recomendado para ser usado na fase de desenvolvimento da simulação ou para apresentações, pois permite obter uma imagem detalhada do estado da simulação em qualquer ponto da execução e seguir o que acontece dentro da rede.

### **Tkenv**

O interface Tkenv é um interface gráfico, portátil que utiliza janelas. O Tkenv suporta a visualização da simulação como o Tvenv. O Tkenv também suporta a visualização dos resultados de simulação durante a execução. Os resultados podem ser mostrados como histogramas ou séries temporais. Isto pode acelerar o processo da operação de verificação e correcção do programa de simulação e fornece um bom ambiente para experiências com o modelo durante a execução.

## **5.4 Análise do sistema a implementar**

Dada uma estrutura específica da rede de telecomunicações pretende-se analisar o grau de serviço (“Grade Of Service” - GOS) da rede em função do método de encaminhamento de tráfego utilizado.

---

<sup>4</sup> Tk é um toolkit de interacção com o utilizador que fornece primitivas de interacção mais avançadas do que as normalmente presentes no sistema Unix.

<sup>5</sup> Este interface com o utilizador tem vindo a ser depreciado no OMNeT++, já não se encontrando presente na versão 2.0 do sistema, já disponível.

O nosso objectivo é comparar algumas medidas de desempenho da rede, tais como bloqueios ponto a ponto e tráfego transportado, conseguidos pelos vários métodos de encaminhamento, em situações de carga e sobrecarga.

### 5.4.1 Descrição do sistema

O sistema que se pretende estudar é uma rede de comutação por circuitos, à qual é oferecido tráfego que pode ser de vários tipos e na qual o tráfego é encaminhado de acordo com um determinado método de encaminhamento, de entre vários possíveis.

Da rede de comutação por circuitos fazem parte as centrais de comutação e os feixes que as interligam. Numa rede deste tipo quando uma chamada é estabelecida num dado caminho são ocupados canais em cada arco (feixe) desse caminho, que ficam indisponíveis para outras chamadas durante todo o tempo de serviço da chamada.

Os fluxos de tráfego oferecidos à rede são descritos pela população geradora das chamadas, pela natureza das chegadas e das terminações e pelos parâmetros associados às próprias chamadas.

A população geradora das chamadas pode ser: população de tamanho infinito e/ou população de tamanho finito, tal como foi referido no capítulo 4.

As chamadas são geradas aleatoriamente. Uma chamada recém chegada é servida imediatamente se houver canais (ou circuitos) disponíveis (necessários para o estabelecimento da chamada), caso contrário a chamada é perdida. O tempo de serviço de cada chamada tem uma distribuição exponencial negativa. No caso das chamadas serem geradas por populações de tamanho infinito o tempo entre chegadas tem uma distribuição exponencial negativa. No caso das chamadas serem geradas por populações de tamanho finito o tempo até à próxima chegada tem uma distribuição exponencial negativa que depende do número de fontes livres na população correspondente.

Para estabelecer uma chamada é necessário um determinado número de circuitos em cada feixe, do caminho que seja utilizado pela chamada. O número de circuitos necessários (em cada feixe) depende do tipo de fluxo a que a chamada pertence, pois admite-se tráfego multiclasse com chamadas multicanal.

Os métodos de encaminhamento que estão implementados na actual versão do simulador para o encaminhamento das chamadas são: o Directo, o FAR, o DAR, o DCR e o RTNR.

Podemos definir o estado da rede (conjunto de variáveis necessárias para descrever o sistema) como sendo o número de chamadas entre cada par origem/destino, o número de circuitos livres entre cada par de centrais, etc. Neste sistema existem apenas dois acontecimentos que podem alterar o estado do sistema, que são o estabelecimento de uma chamada e a terminação de uma chamada.

Um sistema de Teletráfego deste tipo é um sistema discreto, apenas do ponto de vista das variáveis de estado, uma vez elas são alteradas apenas quando há a geração ou terminação de uma chamada. Do ponto de vista da variável tempo (índice) é contínuo, pois as mudanças de estado podem ocorrer em qualquer instante. Os tempos de chegada e os tempos de ocupação são variáveis aleatórias, cujos valores são obtidos a partir das distribuições a elas associadas.

Vai ser desenvolvido um modelo de simulação para imitar o comportamento do sistema ao longo do tempo. O modelo de simulação vai ser usado como instrumento para prever o desempenho do sistema num conjunto variado de circunstâncias.

O comportamento ao longo do tempo deste tipo de sistemas pode ser descrito por uma simulação por acontecimentos discretos. O modelo envolverá também componentes aleatórios (associados à geração e terminação das chamadas). O modelo a implementar será então discreto, dinâmico e estocástico.

## 5.5 Descrição do modelo de simulação implementado

Foi desenvolvido um modelo que pretende representar o comportamento estocástico da rede, com o objectivo de permitir o estudo comparativo do desempenho de vários métodos de encaminhamento. O modelo construído foi um modelo de simulação por acontecimentos discretos com aproximação orientada para o processo.

Do modelo fazem parte dois tipos de processos que são o processo gerador e o processo central.

As centrais e os geradores são as entidades usadas no modelo. As chamadas vão ser modeladas por mensagens, e os feixes entre as centrais do sistema vão ser modelados por ligações entre as centrais que são associadas aos recursos do nosso sistema (por exemplo a duração das chamadas é um parâmetro das mensagens). Como exemplo de variáveis de estado do nosso sistema podemos considerar o número de chamadas concluídas entre qualquer par de centrais, o número de circuitos ocupados num feixe e o número de chamadas no sistema.

Os dois acontecimentos principais no sistema são o estabelecimento e a conclusão de uma chamada. No modelo temos vários acontecimentos. A cada acontecimento, no modelo, está sempre associada uma mensagem. Existem vários tipos de mensagens que podem ser divididos em dois grupos: mensagens que correspondem a chamadas e mensagens de controlo. As mensagens que correspondem a chamadas podem ser de quatro tipos: mensagem que corresponde a uma chamada para ser estabelecida (tipo “chamada a estabelecer”), mensagem que corresponde a uma chamada que foi concluída (tipo “chamada concluída”), mensagem que corresponde a uma chamada que não foi possível estabelecer (tipo “chamada impossível”) e mensagem que corresponde ao *crankback* de uma chamada (tipo “chamada crankback”). As mensagens de controlo podem ser de três tipos: mensagem de inicialização, mensagem

de terminação da simulação ou mensagem para actualização da informação usada no encaminhamento das chamadas (tipo “mensagem de encaminhamento”), por exemplo actualização das tabelas de encaminhamento.

A lista de acontecimentos usada para determinar o próximo acontecimento é processada internamente pela linguagem de simulação. Essa lista contém os instantes em que os acontecimentos, de cada tipo, ocorrem, ordenada pelos valores desses instantes.

### 5.5.1 Representação do modelo na linguagem NED

A rede inter-centrais pode ser modelada em OMNeT++ como uma rede com vários módulos `central` (centrais de comutação) e vários módulos `gerador` (geradores de chamadas), tantos geradores como centrais. Estando cada `central` ligada a todas as outras centrais e cada gerador ligado à sua `central`. Os módulos comunicam pela troca de mensagens. Os módulos geradores geram chamadas para as centrais correspondentes, as centrais recebem chamadas que pretendem ser estabelecidas.

Usando a linguagem NED (NEtwork Description) foi definida a estrutura do modelo da rede. A descrição da estrutura do modelo da rede que foi construído encontra-se no apêndice A. A partir do modelo em linguagem NED, resulta a topologia da rede de telecomunicações que vai ser estudada. Na figura 5-1 apresenta-se um exemplo de uma topologia resultante, tendo sido usadas 6 centrais.

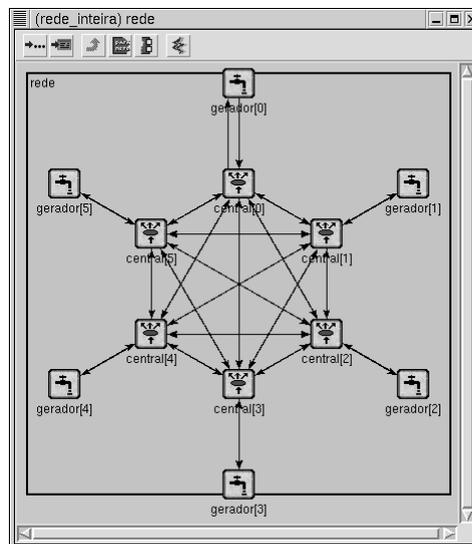


Fig. 5-1 – Topologia do modelo implementado (exemplo com 6 centrais).

### 5.5.2 Módulos simples implementados

O funcionamento dos módulos simples (`gerador` e `central`), foi implementado na linguagem C++ ([Stroustrup97] e [Jobart00]) utilizando a biblioteca de classes de simulação do OMNeT++.

diagramas que descrevem o funcionamento do sistema, apresentados neste capítulo e no apêndice E foram desenvolvidos usando a notação UML ([Larman98]) e foram na sua maior parte criados usando a ferramenta Rational Rose ([Rational00]).

Cada módulo simples é definido através de uma classe C++, derivada de uma classe chamada `cSimpleModule`, onde se redefine a função membro `activity()`. As funções `activity()` dos módulos simples na rede são executadas como co-rotinas, parecendo que são executadas em paralelo.

Vamos começar esta subsecção com uma descrição da interação entre os processos `gerador` e os processos `central`. Lembrar que o processo `gerador` é o responsável pela criação das chamadas, chamadas essas que são trocadas entre processos `central` desde a origem até esta atingir o destino. Estas chamadas quando são enviadas causam uma inativação da fonte que a gerou no caso de população finita, e esta fonte só volta a actividade quando recebe de volta a mensagem que enviou. No caso de populações infinitas, este mecanismo de inativação não é necessário uma vez que se assume a existência de um número infinito de fontes independentes.

Para simplificação da descrição que se segue, sempre que pretenda referir um processo `central` que recebeu uma chamada do processo `gerador` para ser encaminhada, vamos utilizar simplesmente a designação `central origem`. Pela mesma razão vamos também usar as designações `central intermédia` e `central destino` nas situações correspondentes a um processo `central` que acabou de receber uma chamada vinda de outro processo `central` e o qual ainda não é o processo `central` para onde a chamada se destina; e a um processo `central` que acabou de receber uma chamada que lhe era destinada vinda de outro processo `central`, respectivamente.

Os processos do tipo `gerador` e `central` (a executar durante toda a simulação) são os responsáveis pela criação e encaminhamento das chamadas na rede.

A função do `gerador` é gerar mensagens, mensagens essas que correspondem a chamadas a ser estabelecidas. Quando o `gerador` gera uma mensagem, correspondente a uma chamada do tipo “chamada a estabelecer” ela é enviada para a `central` à qual está ligado. A `central` (onde a chamada surgiu - `central origem`) ao receber uma chamada determina se a capacidade livre do feixe que a liga à `central` para onde a chamada pretende ser encaminhada (`central destino`) permite o estabelecimento dessa chamada. Se a capacidade disponível o permitir a chamada a estabelecer é enviada, por esse feixe, para a `central destino`. Quando a `central destino` recebe uma chamada do tipo “chamada a estabelecer” cria uma mensagem do tipo “chamada concluída” (para enviar a si própria) que receberá quando o tempo de ocupação da chamada terminar. Quando isso ocorrer a `central destino` envia essa mensagem para a `central origem`. A `central origem` ao receber esta mensagem envia-a para o `gerador` (apenas no caso de ela corresponder a um fluxo de tráfego gerado por uma população finita,

de forma a indicar o fim da inactivação de uma dada fonte). Tudo isto caso o mecanismo de encaminhamento tenha indicado, à central origem, o caminho directo (caminho formado apenas pelo feixe que liga a central origem à central destino) como caminho a usar no encaminhamento da chamada. Este processo pode ser observado nos dois diagramas seguintes.

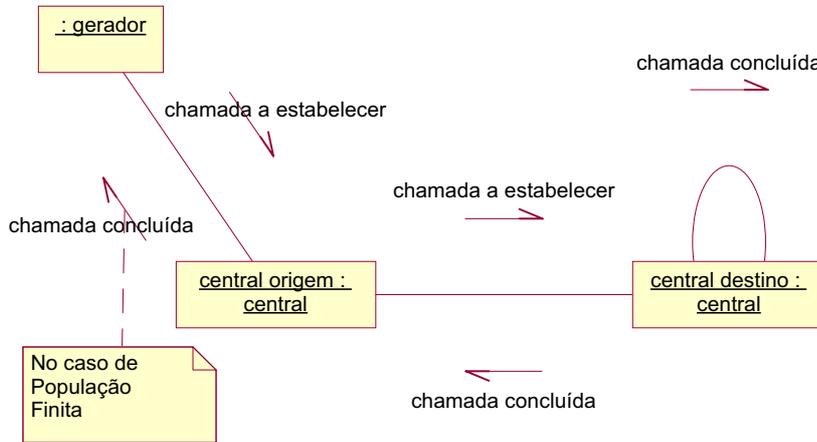


Fig. 5-2 - Diagrama de colaboração (encaminhamento Directo).

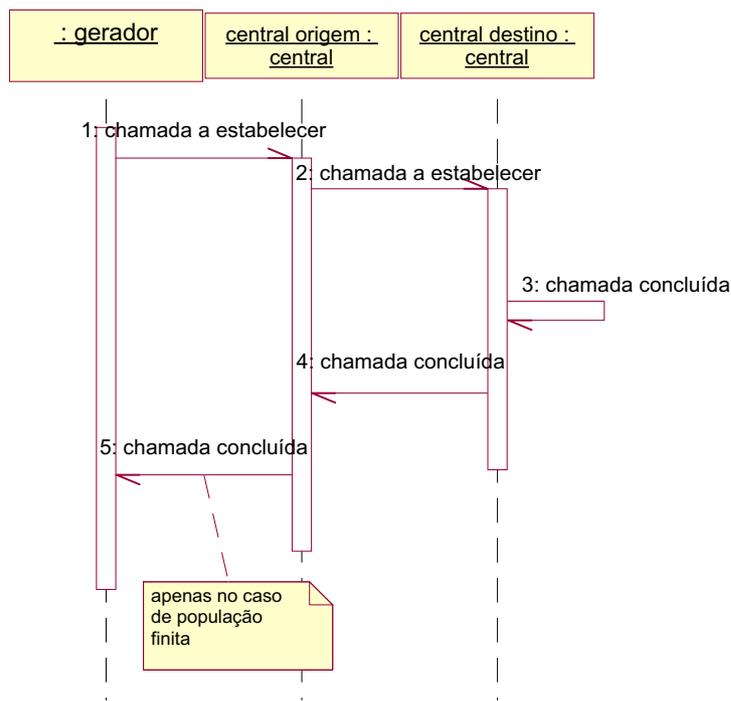


Fig. 5-3 – Diagrama de sequência (encaminhamento Directo).

No segundo diagrama pode ver-se também a sequência temporal da passagem das mensagens.

Vamos agora analisar o caso em que o mecanismo de encaminhamento fornece à central origem, depois dela ter recebido do gerador uma chamada do tipo “chamada a estabelecer”, um caminho com uma central de trânsito. Neste caso a central origem determina se existe capacidade disponível no feixe que a liga à central intermédia e se assim for envia para a central intermédia a chamada do

tipo “chamada a estabelecer”. Por sua vez a *central* intermédia, após ter recebido uma chamada do tipo “chamada a estabelecer”, determina se existe capacidade disponível no feixe que a liga à *central* destino e se assim for envia para a *central* destino a chamada do tipo “chamada a estabelecer”. Quando a *central* destino recebe uma chamada do tipo “chamada a estabelecer” cria uma mensagem do tipo “chamada concluída” que vai receber quando o tempo de ocupação da chamada terminar. Quando isso ocorrer a *central* destino envia essa mensagem para a *central* intermédia. A *central* intermédia ao receber esta mensagem envia-a para a *central* origem. A *central* origem ao receber esta mensagem envia-a para o gerador, mas, apenas no caso de ela corresponder a um fluxo de tráfego gerado por uma população finita. Os diagramas seguintes ilustram este processo.

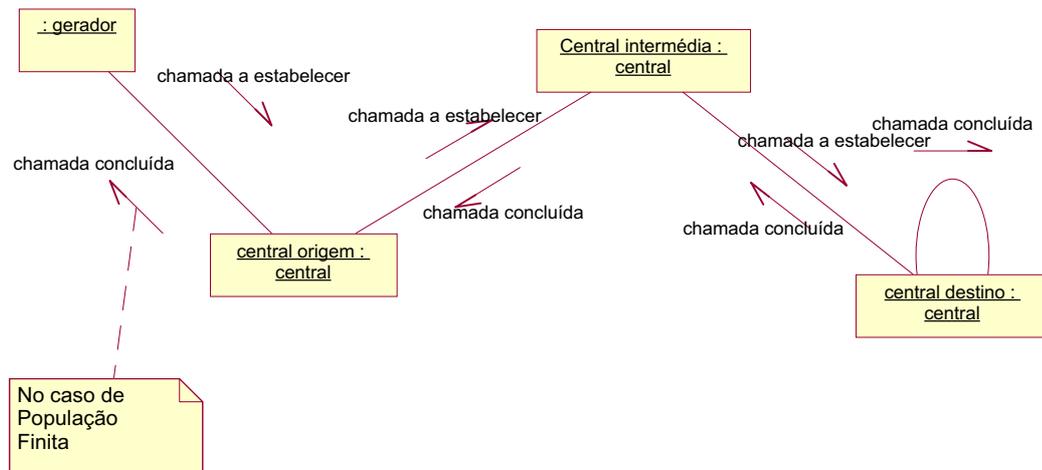


Fig. 5-4 - Diagrama de colaboração (encaminhamento alternativo).

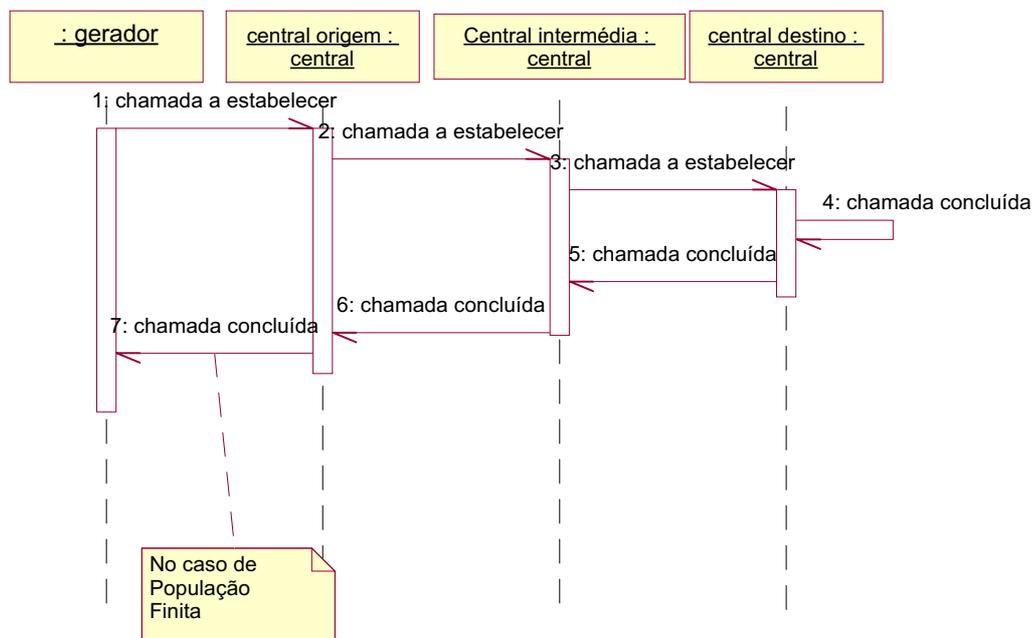


Fig. 5-5 – Diagrama de sequência (encaminhamento alternativo).

No segundo diagrama pode ver-se também a sequência temporal da passagem das mensagens.

Uma outra situação que pode ocorrer no caso anterior é, quando a chamada do tipo “chamada a estabelecer” chegar à central intermédia, esta determinar que não existe capacidade disponível no feixe que a liga à central destino. Neste caso duas coisas podem acontecer dependendo da central intermédia permitir ou não *crankback*. No caso da central intermédia não permitir *crankback*, ela envia uma mensagem do tipo “chamada impossível” para a central origem. A central origem ao receber esta mensagem envia-a para o gerador, mas, apenas no caso de ela corresponder a um fluxo de tráfego gerado por uma população finita.

No caso da central intermédia permitir *crankback*, ela envia uma mensagem do tipo “chamada crankback” para a central origem. A central origem ao receber uma mensagem desse tipo solicita ao mecanismo de encaminhamento que lhe indique um caminho alternativo para estabelecer a chamada. O caminho indicado, à central origem, pelo mecanismo de encaminhamento é normalmente um caminho que tem uma central de trânsito (situação mais frequente). Neste caso a central origem determina se existe capacidade disponível no feixe que a liga à nova central intermédia e se assim for envia para a nova central intermédia a chamada do tipo “chamada a estabelecer”. Por sua vez esta central, após ter recebido uma chamada do tipo “chamada a estabelecer”, determina se existe capacidade disponível no feixe que a liga à central destino e se assim for envia para a central destino a chamada do tipo “chamada a estabelecer”. Quando a central destino recebe uma chamada do tipo “chamada a estabelecer” cria uma mensagem do tipo “chamada concluída” que vai receber quando o tempo de ocupação da chamada terminar. Quando isso ocorrer a central destino envia essa mensagem para a central intermédia (central que foi utilizada no encaminhamento da chamada). Esta central intermédia ao receber essa mensagem envia-a para a central origem. A central origem ao receber a mensagem envia-a para o gerador, mas, apenas no caso de ela corresponder a um fluxo de tráfego gerado por uma população finita. Este caso pode ser observado nos dois diagramas seguintes.

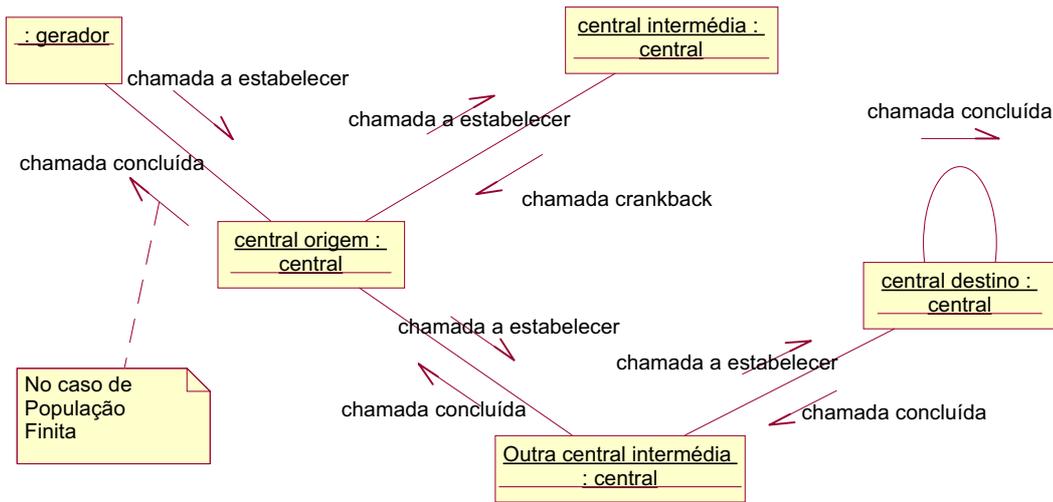


Fig. 5-6 - Diagrama de colaboração (encaminhamento alternativo após *crankback*).

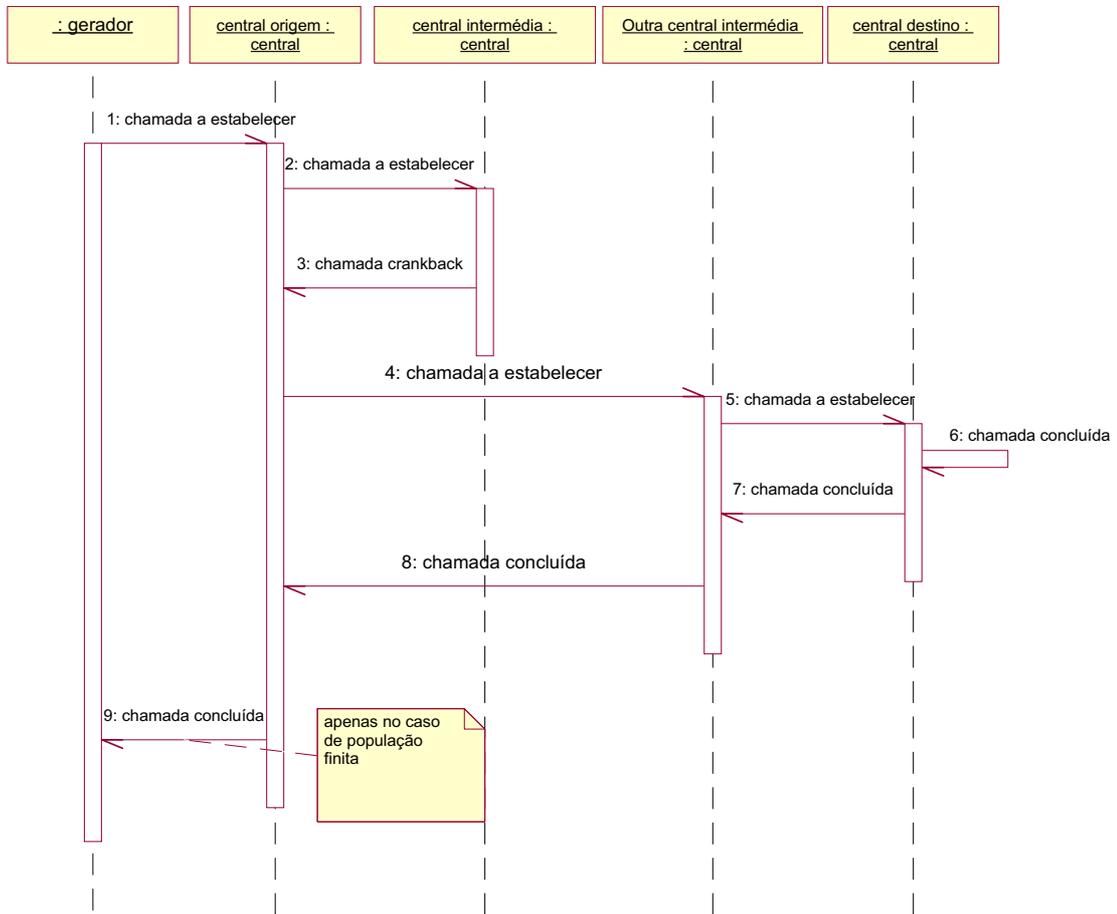


Fig. 5-7 – Diagrama de sequência (encaminhamento alternativo após *crankback*).

Tal como nos casos anteriores, apresenta-se no segundo diagrama a sequência temporal da passagem das mensagens, neste caso bastante útil dado o número de mensagens utilizadas.

Note-se que as mensagens que são passadas entre os vários processos, em todos os diagramas, são assíncronas o que significa que o processo que envia uma mensagem não fica inactivo, continua o processamento independentemente do tratamento que seja dado à mensagem pelo processo destino.

Os casos aqui apresentados embora englobem as situações principais do comportamento do processo `central` não são exaustivos.

Antes de entrar nos pormenores referentes ao funcionamento do processo `central` e `gerador` vão ser apresentadas sucintamente as características de cada um deles. Para isso lembra-se que existem tantos processos centrais no modelo como centrais na rede e que a cada processo `central` está associado um processo `gerador`.

As características principais de cada `gerador` são:

- Modelar a geração de vários tipos de fluxos de tráfego, podendo coexistir vários no mesmo modelo; os fluxos de tráfego gerados, de diferentes tipos, podem usar parâmetros diferentes para os processos estocásticos de chegada e/ou terminação;
- Obter no início da simulação, a partir de um ficheiro, valores para vários parâmetros; no caso de tráfego exógeno de Poisson esses valores são as intensidades de tráfego entre quaisquer dois pares de centrais; no caso de tráfego exógeno de população finita (tipo Engset) esses valores são o número de fontes de tráfego em cada origem, a intensidade de chegadas por fonte livre e a proporção de tráfego gerado nessa `central` para cada destino;
- Para ambos os tipos de tráfego exógeno haverá ainda que ler (a partir do ficheiro referido anteriormente) o número de canais (de 64 kbps) necessários pelas chamadas de cada tipo de fluxo;
- Criar, para a `central` à qual está ligado, ao longo de toda a simulação, tráfego que respeite os parâmetros referidos anteriormente.

As características principais de cada `central` são:

- Chamar as funções necessárias para determinar o caminho a seguir por cada uma das mensagens (chamadas) recebidas do `gerador` (ou de outras centrais) durante toda a simulação;
- Chamar as funções necessárias para armazenar as estatísticas respeitantes a esse tráfego;
- Enviar as chamadas da origem até ao destino passando por todas as centrais intermédias (mesmo que o caminho completo - caminho que a chamada deve seguir até ao destino - tenha sido conhecido logo na `central` origem). Decrementar a capacidade disponível dos feixes sempre que uma chamada é enviada por eles em direcção ao destino;

- Aquando do término da chamada, reenviar uma mensagem do tipo “chamada concluída” do destino em direcção à origem, pelo percurso inverso ao utilizado pela chamada a estabelecer. Incrementar as capacidades disponíveis à medida que essa mensagem passa nos várias feixes em direcção à origem;
- Chamar as funções necessárias para actualizar a informação a ser usada na decisão de encaminhamento das chamadas.

### 5.5.2.1 Processo Central

Segue-se a descrição em pormenor do funcionamento do processo `central`, indo para tal usar-se o diagrama seguinte.

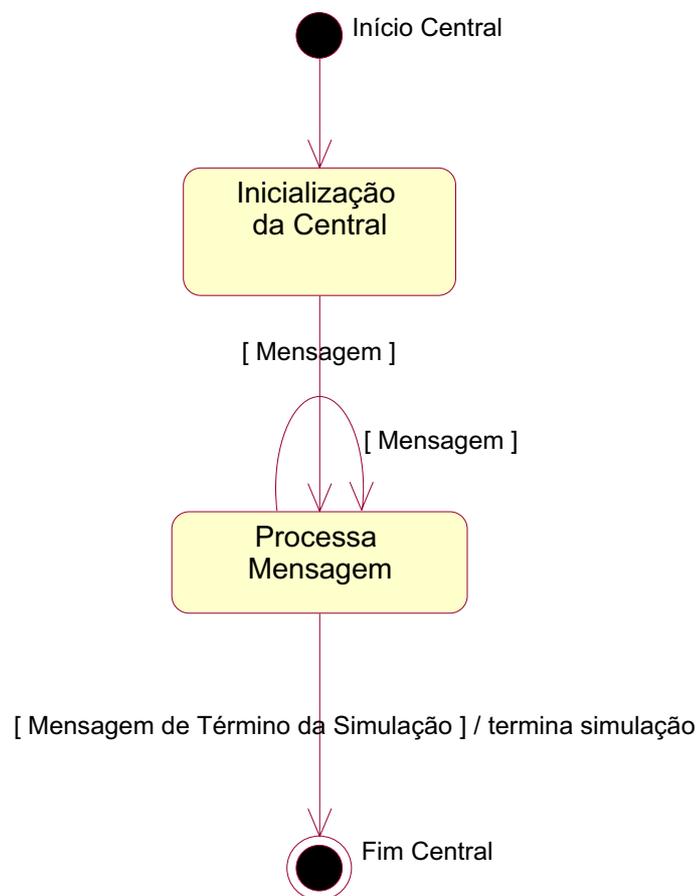


Fig. 5-8 - Diagrama de estados do processo `central`.

O funcionamento de cada `central` pode ser descrito por dois estados principais. Quando se inicia a simulação cada processo `central` entra no primeiro estado, designado no diagrama anterior por `Inicialização da Central`. No diagrama de actividade seguinte, correspondente a este estado, podemos observar as várias actividades desencadeadas neste estado bem como os objectos intervenientes.

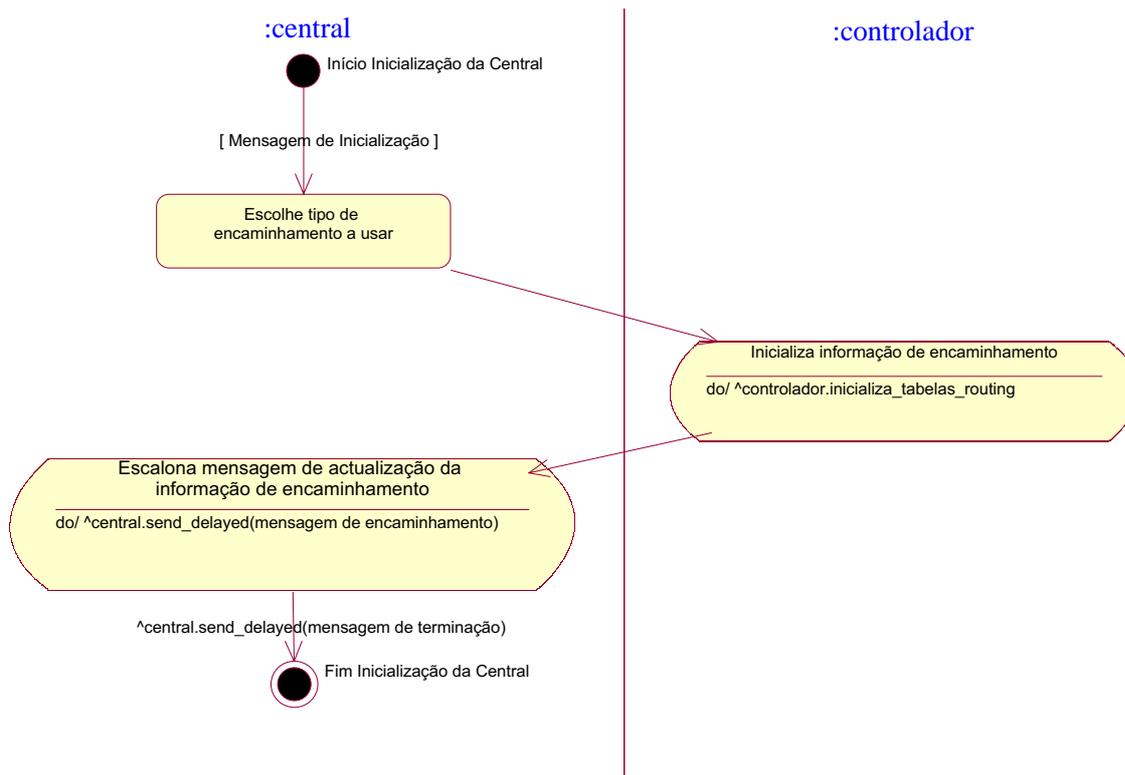


Fig. 5-9 – Diagrama de actividade do estado Inicialização da Central.

Cada *central* antes de iniciar o processamento necessita de receber do processo gerador uma mensagem de inicialização para obter dela o seu número na rede. Depois desta recepção, passa ao estado em que determina qual o encaminhamento a usar, bem como o tipo de controlo (modo de actualização da informação a usar pelo método de encaminhamento). Existe uma função que permite escolher explicitamente quais dos componentes de encaminhamento construídos serão usados numa simulação particular. A função chama-se *escolhe\_encaminhamento* e é invocada por cada processo *central* como indicado no diagrama da figura 5-9. Nessa função são criadas instâncias das classes derivadas da classe *stats*, da classe *controlador* e da classe *router*, correspondentes ao componente de encaminhamento que se pretende simular. As classes *stats*, *controlador* e *router* e as classes derivadas delas são referidas adiante.

Depois disso chama um método do objecto *controlador*, ou de uma sua especialização, que faz a inicialização da informação que é usada no encaminhamento das chamadas (por exemplo a inicialização das tabelas de encaminhamento). A próxima actividade executada por cada *central* é Escalona Mensagem de Actualização da Informação (o que corresponde a enviar uma mensagem para si própria), para ser feita a actualização da informação usada no encaminhamento das chamadas. Por último, cada *central* escalona uma mensagem de terminação para a altura em que a simulação deva terminar, o que corresponde a enviar uma mensagem de término da simulação para si própria e que vai ser recebida por si quando for o instante em que a simulação deva terminar.

Terminado o estado de Inicialização da Central cada uma das centrais inicia o estado Processa Mensagem. As centrais só deixam de estar neste estado quando ocorrer a mensagem de término da simulação. Quando esta mensagem ocorrer a simulação termina.

No diagrama de actividade seguinte, correspondente à decomposição do estado Processa Mensagem, podemos observar as várias actividades desencadeadas neste estado bem como os objectos intervenientes.

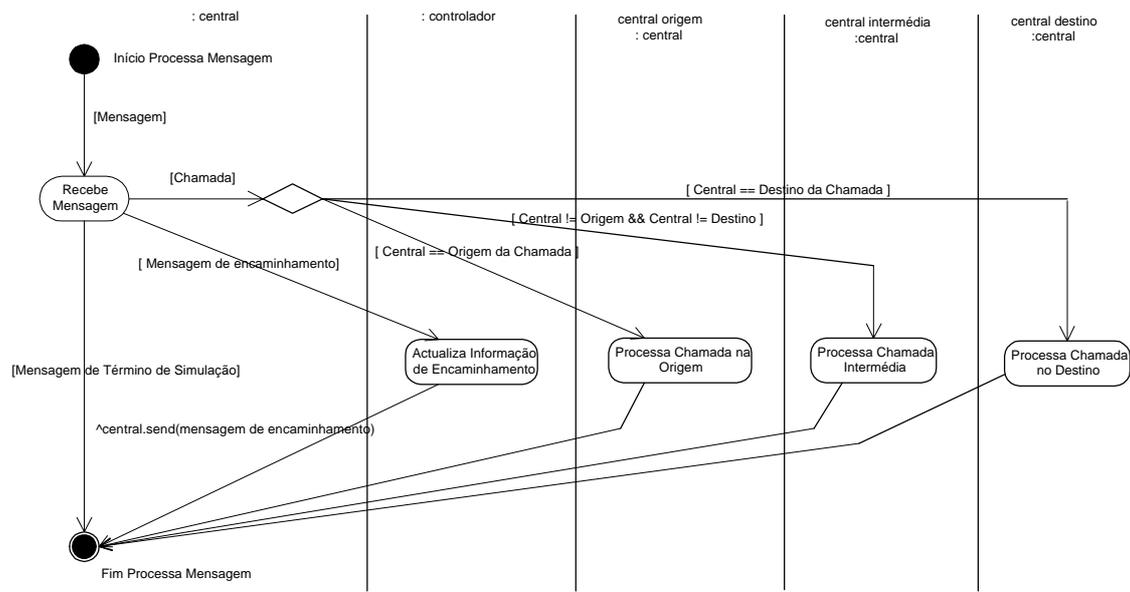


Fig. 5-10 – Diagrama de actividade do estado Processa Mensagem.

No estado Processa Mensagem cada central espera por uma mensagem, podendo a mensagem ter sido enviada pela própria central, por qualquer outra central ou pelo gerador. A mensagem foi gerada pela própria central no caso da mensagem ser do tipo “mensagem de encaminhamento”, ou ser do tipo “chamada concluída”, ou ainda de outros. A mensagem foi gerada pelo gerador no caso da mensagem ser do tipo “chamada a estabelecer”. A mensagem foi gerada por outra central no caso da mensagem ser do tipo “chamada impossível”, ou do tipo “chamada concluída”, ou ainda de outros. Quando chegar uma mensagem à central ela vai processá-la. O tratamento a dar a cada mensagem depende do tipo da mensagem. Caso a mensagem recebida corresponda a uma chamada a central vai determinar se se trata de uma chamada para a qual ela é a central origem (central onde a chamada surgiu do gerador), ou se se trata de uma chamada para a qual ela é a central destino (central para onde a chamada pretende ser estabelecida) ou se se trata de uma chamada para a qual ela é uma central intermédia (central de trânsito usado no estabelecimento da chamada). Depois disso chama o método do objecto central correspondente à condição que for verificada. A central fica assim no estado Processa Chamada na Origem, ou no estado Processa Chamada Intermédia ou no estado Processa Chamada no Destino.

Caso a mensagem corresponda a uma mensagem de actualização da informação de encaminhamento, a `central` passa neste caso ao estado `Actualiza Informação de Encaminhamento`. Neste caso a `central` chama um método do objecto `controlador`, ou de uma sua especialização, que faz a actualização da informação a usar no encaminhamento das chamadas. Depois do estado `Actualiza Informação de Encaminhamento` a `central` vai escalonar (o que corresponde a enviar a mensagem para si própria) uma mensagem do tipo “mensagem de encaminhamento”. Esta mensagem de encaminhamento será a próxima mensagem, deste tipo, que a `central` irá receber.

Caso a mensagem seja uma mensagem de término da simulação a `central` desencadeia as operações que devem ser realizadas imediatamente antes da simulação terminar.

No diagrama de actividade da figura 5-11, correspondente ao estado `Processa Chamada na Origem`, podemos observar as várias actividades desencadeadas neste estado.

A primeira coisa que a `central` faz quando entra no estado `Processa Chamada na Origem` é verificar qual o tipo da mensagem que lá chegou. Caso a mensagem seja do tipo “chamada a estabelecer” a `central` vai tratar uma chamada que acabou de chegar do `gerador`. Para isso começa por chamar um método do objecto `stats`, ou de uma sua especialização, que faz a contabilização do número de chamadas novas. De seguida chamará um método do objecto `router`, ou de uma sua especialização, que determina a próxima `central` e/ou o caminho previsto para atingir a `central` destino. Depois disto a `central` verifica a capacidade disponível do feixe que liga esta `central` à próxima `central`. Se a capacidade disponível permitir estabelecer a chamada nesse feixe a actividade seguinte será `Envia Chamada para Próxima Central`.

Se a actividade a realizar for `Envia Chamada para Próxima Central`, então a chamada irá ocupar os circuitos necessários nesse feixe. É assim necessário efectuar as acções seguintes: decrementar a capacidade disponível desse feixe, chamar um método do objecto `router`, ou de uma sua especialização, para o informar de que a capacidade disponível do feixe foi alterada, chamar um método do objecto `stats`, ou de uma sua especialização, para ser contabilizada a ocupação de circuitos no feixe e enviar a mensagem para a próxima `central`. Após a realização desta actividade o processamento da mensagem termina.

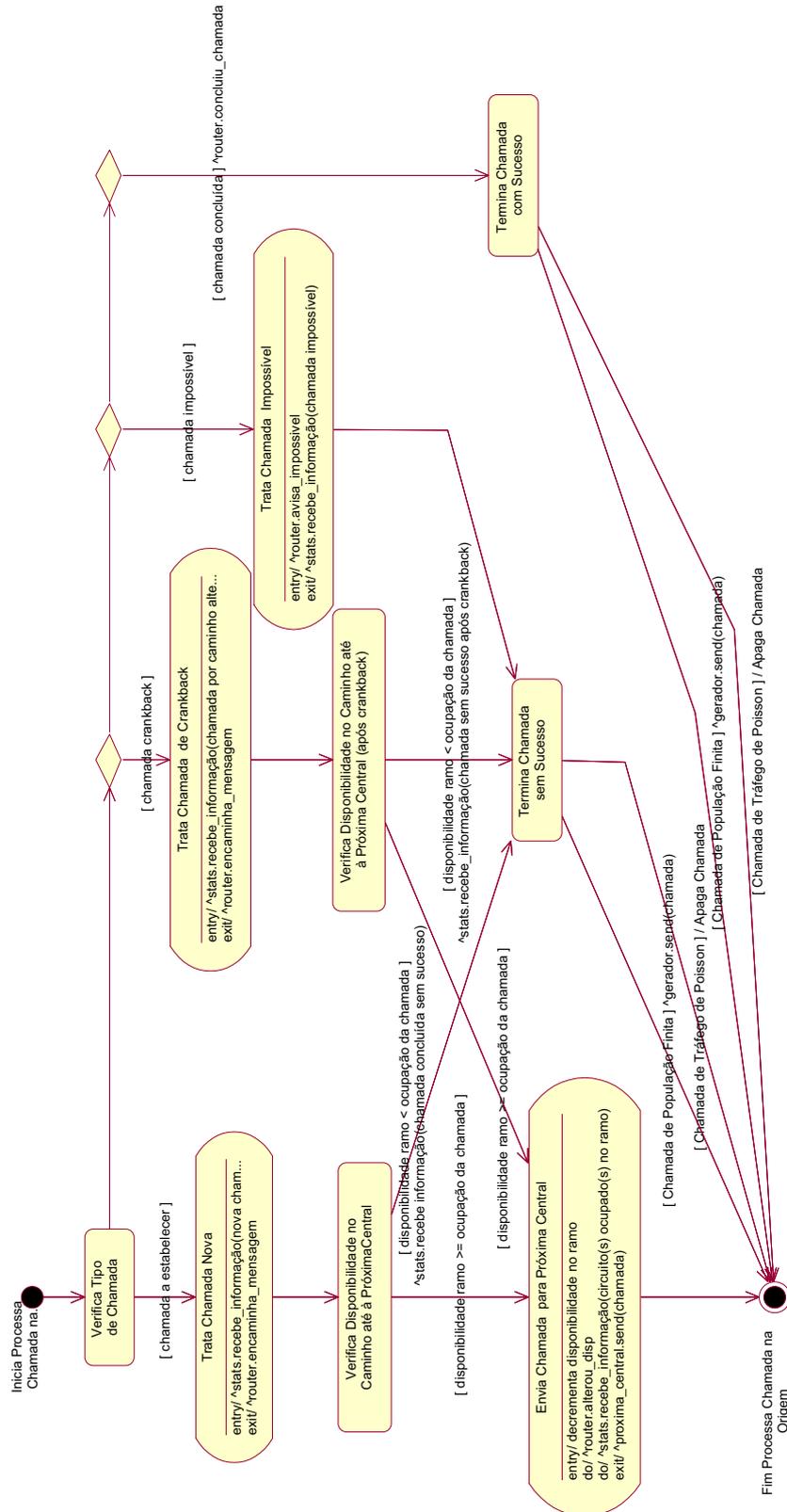


Fig. 5-11 – Diagrama de actividade do estado Processa Chamada na Origem.

Se, ainda no caso da mensagem ser do tipo “chamada a estabelecer”, a capacidade disponível não permitir estabelecer a chamada, no feixe que liga à próxima *central*, é necessário chamar um método do objecto *stats*, ou de uma sua especialização, para ser contabilizada uma chamada terminada sem sucesso. Neste caso a *central* passa ao estado de escolha Termina Chamada sem Sucesso. Se a chamada corresponder a um fluxo de tráfego de população finita a *central* envia a chamada para o gerador. Caso contrário, isto é, no caso da chamada corresponder a um fluxo de tráfego de Poisson a *central* apaga simplesmente a mensagem. Em ambas as situações, termina o processamento da mensagem.

Caso a mensagem seja do tipo “chamada crankback” a *central* realiza a actividade Trata Chamada de Crankback. Para isso é necessário executar as duas acções seguintes: chamar um método do objecto *stats*, ou de uma sua especialização, para ser contabilizada uma chamada por caminho alternativo e chamar um método do objecto *router*, ou de uma sua especialização, para determinar a próxima *central* e/ou o caminho previsto para atingir a *central* destino. Depois disto a *central* verifica a capacidade disponível do feixe que liga esta *central* à próxima *central*. Se a capacidade disponível permitir estabelecer a chamada nesse feixe a actividade seguinte será Envia Chamada para Próxima Central. Esta actividade é a mesma que a realizada no caso da mensagem ser do tipo “chamada a estabelecer”, que já foi descrita. Depois desta actividade ser realizada, termina o processamento da mensagem.

Se, ainda no caso da mensagem ser do tipo “chamada crankback”, a capacidade disponível não permitir estabelecer a chamada, no feixe que liga à próxima *central*, é necessário chamar um método do objecto *stats*, ou de uma sua especialização, para ser contabilizada uma chamada terminada sem sucesso depois de ter ocorrido *crankback*. Neste caso a *central* passa ao estado de escolha Termina Chamada sem Sucesso, já anteriormente descrito.

Caso a mensagem seja do tipo “chamada impossível” a *central* chama um método do objecto *router*, ou de uma sua especialização. Depois chama um método do objecto *stats*, ou de uma sua especialização, para contabilizar uma chamada concluída sem sucesso. E de seguida passa ao estado de escolha Termina Chamada sem Sucesso. A partir daqui tudo se passa como nos casos anteriores.

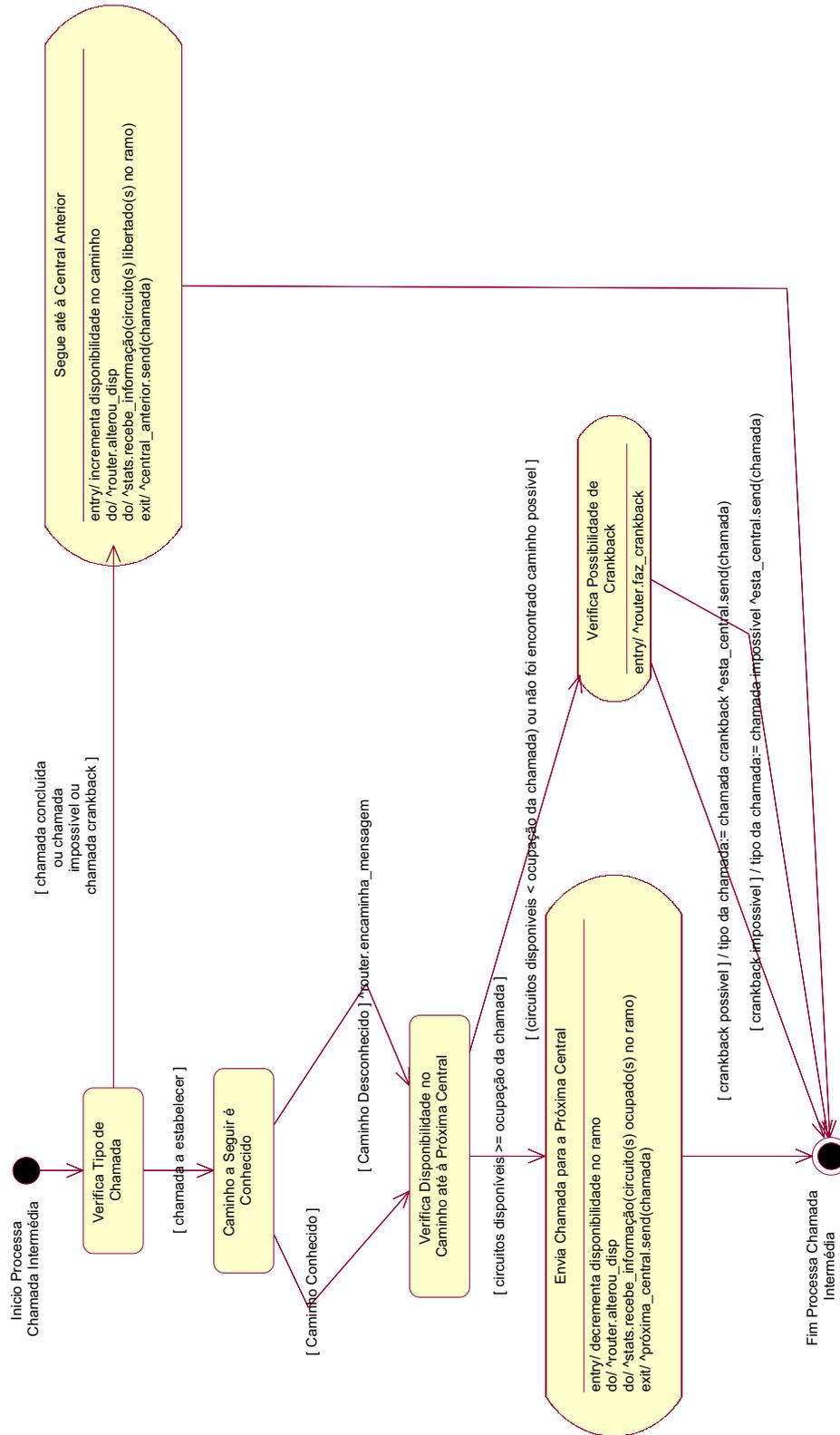


Fig. 5-12 – Diagrama de actividade do estado Processa Chamada Intermédia.

Caso a mensagem seja do tipo “chamada concluída” a *central* chama um método do objecto *router*, ou de uma sua especialização, para o informar que foi concluída uma chamada. Neste caso a *central* passa ao estado de escolha Termina Chamada com Sucesso. Se a chamada corresponder a um fluxo de tráfego de população finita a *central* envia a chamada para o gerador. Caso contrário, isto é, no caso da chamada corresponder a um fluxo de tráfego de Poisson a *central* apaga simplesmente a mensagem. Em ambas as situações, termina o processamento da mensagem. Acaba assim o estado Processa Chamada na Origem.

No diagrama de actividade da figura 5-12, correspondente ao estado Processa Chamada Intermédia, podemos observar as várias actividades desencadeadas neste estado.

A primeira coisa que a *central* faz quando entra no estado Processa Chamada Intermédia é verificar qual o tipo da mensagem que lá chegou. Caso a mensagem seja do tipo “chamada a estabelecer” a *central* vai tratar uma chamada que acabou de chegar a uma *central* de trânsito. Quando uma chamada chega a uma *central* de trânsito uma das duas situações pode acontecer: a chamada já conhece o caminho que deve seguir ou a chamada ainda não conhece o caminho a seguir. Caso o caminho a seguir já seja conhecido, a *central* vai verificar a capacidade disponível do feixe que liga esta *central* (*central* onde a chamada se encontra) à próxima *central* no caminho. Se a capacidade disponível do feixe permitir estabelecer a chamada a *central* passa a realizar a actividade Envia Chamada para a Próxima Central. As acções a executar para realizar esta actividade são as seguintes: decrementar a capacidade disponível desse feixe, chamar um método do objecto *router*, ou de uma sua especialização, para o informar de que a capacidade disponível do feixe foi alterada, chamar um método do objecto *stats*, ou de uma sua especialização, para ser contabilizada a ocupação de circuitos no feixe e enviar a mensagem para a próxima *central*. Após a realização desta actividade o processamento da mensagem termina.

Considere-se ainda o caso da mensagem ser do tipo “chamada a estabelecer”, mas em que se verificou uma das duas situações seguintes: não havia capacidade disponível no feixe que liga à próxima *central* ou não foi encontrado caminho possível. Nestes casos a *central* determina se é possível fazer *crankback*, para isso recorrendo ao objecto *router*, ou uma sua especialização. Se for possível fazer *crankback* envia para si própria uma mensagem, do tipo “chamada crankback”. Se não for possível fazer *crankback* envia para si própria uma mensagem, do tipo “chamada impossível”.

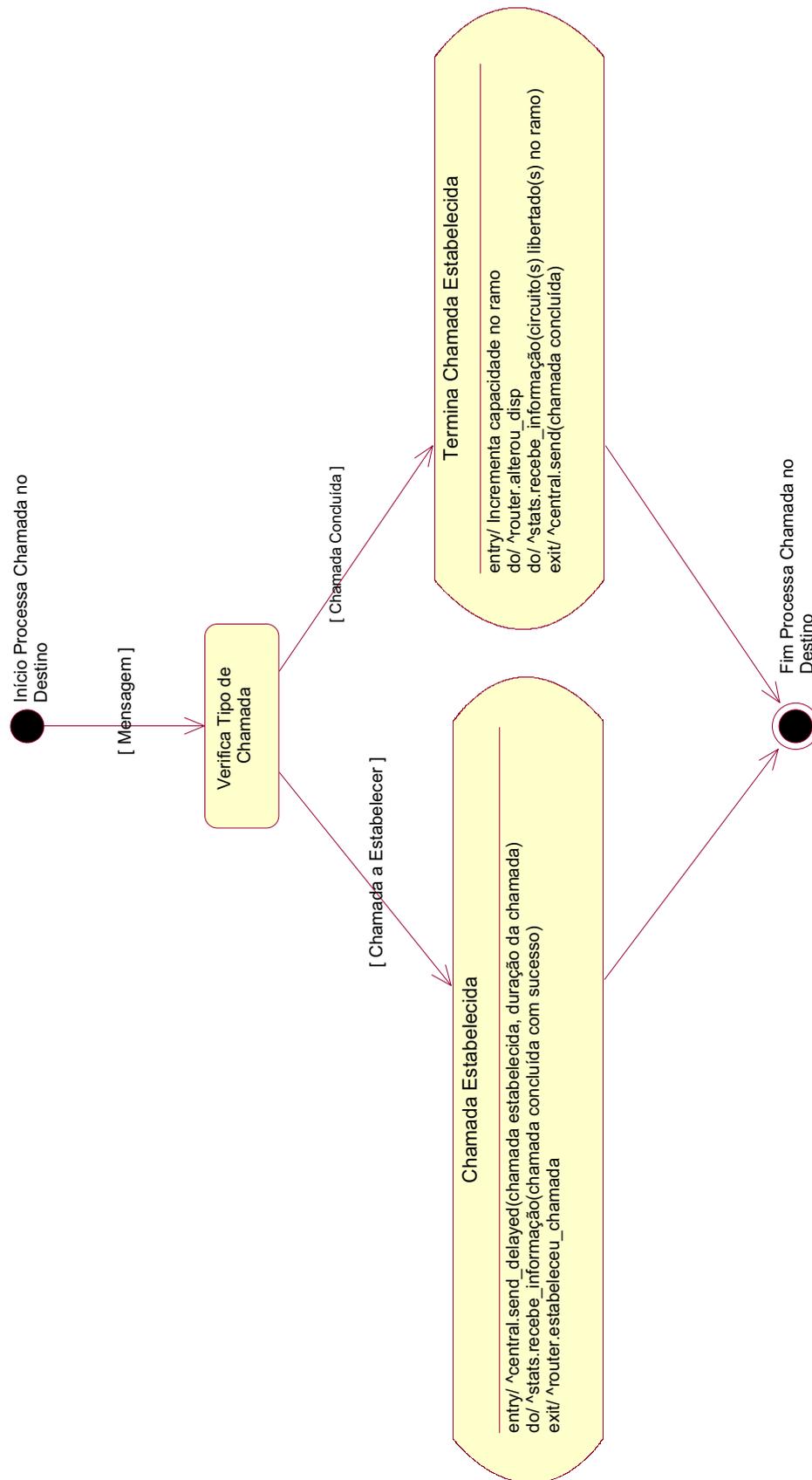


Fig. 5-13 – Diagrama de actividade do estado Processa Chamada no Destino.

No caso ainda da mensagem ser do tipo “chamada a estabelecer”, mas na situação do caminho a seguir não ser conhecido, a `central` chama um método do objecto `router`, ou de uma sua especialização, para determinar a próxima `central` e/ou caminho previsto para atingir a `central` destino. Tudo o que se passa a seguir é o mesmo que no caso anterior a partir do estado de escolha `Verifica Disponibilidade no Caminho até à Próxima Central`.

No caso da mensagem ser do tipo “chamada concluída”, “chamada impossível” ou “chamada crankback” a única actividade a realizar é `Determina Caminho a Seguir até à Origem`. Para realizar esta actividade é necessário executar as seguintes acções: incrementar a capacidade disponível do feixe que liga esta `central` à `central` anterior, chamar um método do objecto `router`, ou de uma sua especialização, para o informar de que a capacidade disponível do feixe foi alterada, chamar um método do objecto `stats`, ou de uma sua especialização, para ser contabilizada a libertação de circuitos no feixe e por último enviar a chamada para a `central` anterior. Após a realização destas acções o processamento da mensagem termina. Note que, neste parágrafo utilizou-se a designação `central` anterior quando se queria referir a `central`, que foi utilizada no caminho usado no estabelecimento da chamada, por onde a chamada tinha passado, no seu estabelecimento, imediatamente antes de chegar à `central` onde a chamada se encontra.

No diagrama de actividade da figura 5-13, correspondente à decomposição do estado `Processa Chamada no Destino`, podemos observar as várias acções desencadeadas neste estado.

A primeira coisa que a `central` faz quando entra no estado `Processa Chamada no Destino` é verificar qual o tipo da mensagem que lá chegou. Caso a mensagem seja do tipo “chamada a estabelecer” a `central` vai tratar uma chamada que acabou de chegar ao destino. Para isso começa por escalonar uma mensagem do tipo “chamada concluída” (o que corresponde a enviar uma mensagem para si própria). Esta mensagem vai ser recebida pela própria `central` com um atraso igual ao tempo de duração da chamada. Depois chama um método do objecto `stats`, ou de uma sua especialização, que faz a contabilização do número de chamadas concluídas com sucesso. E por fim chama um método do objecto `router`, ou de uma sua especialização, para o informar que foi estabelecida uma chamada.

Caso a mensagem seja do tipo “chamada concluída” a `central` vai tratar uma chamada que ela própria escalonou. Esta chamada indica que o tempo de duração de uma chamada que tinha sido estabelecida chegou ao fim. Por isso a `central` vai realizar a actividade `Termina Chamada Estabelecida`. Nesta actividade são executadas as seguintes acções: incrementar a capacidade disponível do feixe que liga esta `central` à `central` anterior, chamar um método do objecto `router`, ou de uma sua especialização, para o informar de que a capacidade disponível do feixe foi alterada, chamar um método do objecto `stats`, ou de uma sua especialização, para ser contabilizada a libertação de circuitos no feixe e por último enviar a chamada para a `central` anterior. Após a realização destas acções o

processamento da mensagem termina. Neste parágrafo utilizou-se a designação `central` anterior com o mesmo sentido utilizado na descrição do estado `Processa Chamada Intermédia`.

### 5.5.2.2 Processo Gerador

Esta subsecção vai ser dedicada ao modo de funcionamento do processo `gerador`. Para ilustrar o seu funcionamento vai usar-se o diagrama seguinte:

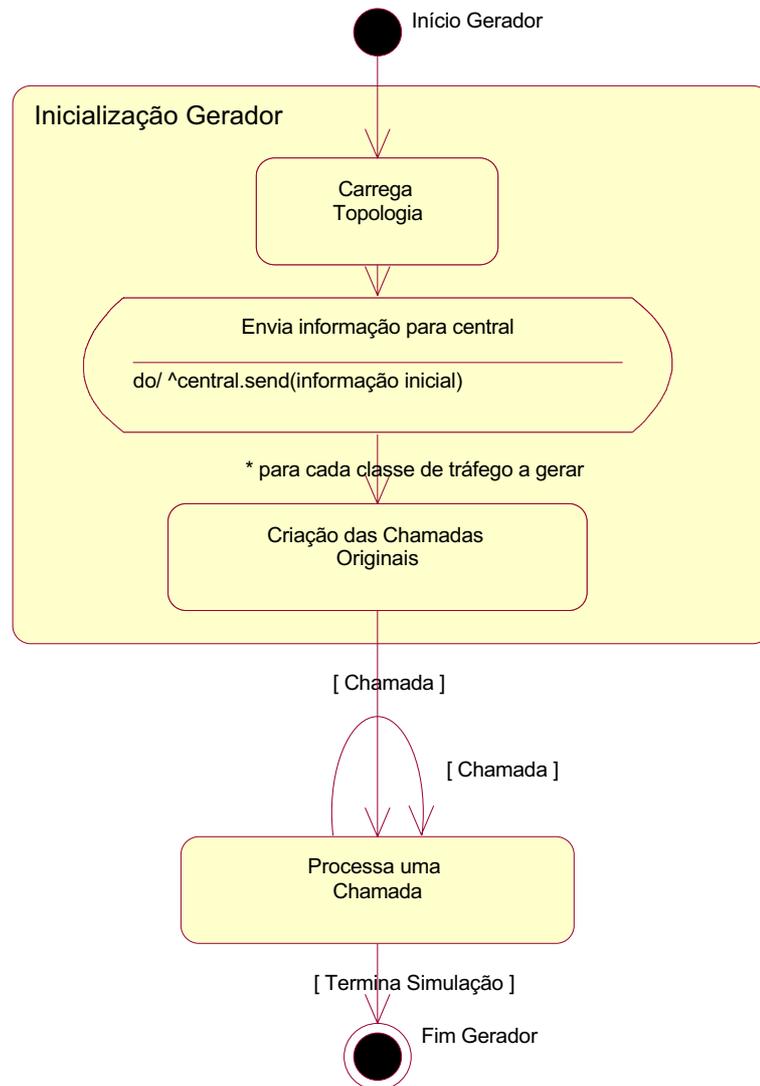


Fig. 5-14 – Diagrama do processo gerador.

O funcionamento de cada `gerador` pode ser também descrito por dois estados principais, como aconteceu para o processo `central`. Quando se inicia a simulação cada processo `gerador` começa no estado designado por `Inicialização Gerador`. Este estado pode ser decomposto em vários subestados. No primeiro subestado designado por `Carrega Topologia`, o `gerador` vai ler a configuração da rede de um ficheiro, que contém a informação sobre a topologia da rede, e ler também desse ficheiro os parâmetros do modelo, por exemplo os parâmetros dos tráfegos. O conteúdo deste ficheiro será explicado em

pormenor no apêndice C, secção C.1. Depois do subestado Carrega Topologia estar completo, a próxima acção efectuada pelo *gerador* é enviar para a *central*, à qual está ligado, uma mensagem com informação necessária à inicialização da *central*. Após isto, e para completar o processo de inicialização, o *gerador* tem ainda que passar no subestado designado por Criação das Chamadas Originais, tantas vezes quanto o número de classes de tráfego a gerar.

No diagrama de actividades seguinte, correspondente ao subestado Criação das Chamadas Originais, podemos observar as várias acções desencadeadas neste subestado.

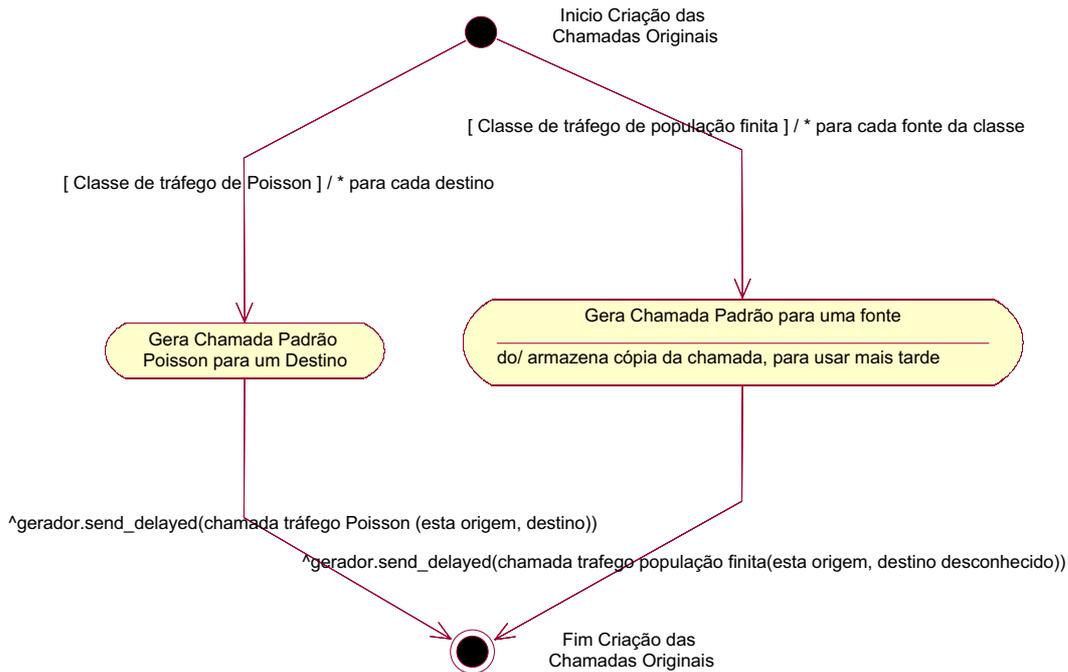


Fig. 5-15 – Diagrama de actividade do subestado Criação das Chamadas Originais.

As acções a serem realizadas dependem da classe de tráfego a gerar (tipo de fluxo de tráfego a gerar). Se o tráfego a gerar corresponde a uma classe de tráfego de Poisson (o que corresponde ao caso de, no sistema, existir um tipo de fluxo de tráfego no qual o tráfego exógeno é de Poisson) é criada, para cada uma das centrais destino (centrais destino são todas as centrais para as quais exista tráfego com origem na *central* que se encontra ligada ao *gerador*), uma mensagem do tipo “chamada a estabelecer” e é escalonada essa mensagem (o que corresponde a enviá-la para si próprio). Cada uma dessas mensagens vai ser recebida pelo próprio *gerador* com um atraso igual ao tempo aleatório que demora a ocorrer uma nova chamada para o destino correspondente.

Se o tráfego a gerar corresponde a uma classe de tráfego de população finita (o que corresponde ao caso de, no sistema, existir um tipo de fluxo de tráfego no qual o tráfego exógeno é de população finita) é criada, para cada indivíduo dessa população, uma mensagem do tipo “chamada a estabelecer” e é escalonada essa mensagem. Esta mensagem vai ser recebida pelo próprio *gerador* com um atraso igual

ao tempo, aleatório, que demora a ocorrer uma nova chamada desse indivíduo. Além de criar, para cada fonte, uma mensagem do tipo “chamada a estabelecer” é também guardada uma cópia de cada uma dessas mensagens, antes de serem escalonadas. Cada uma dessas cópias vai ser usada quando for necessário escalonar uma nova chamada com as mesmas características (ver o estado Coloca uma Chamada em Espera no diagrama de estados Processa uma Chamada, apresentado na figura 5-16).

A cada mensagem, que corresponda a uma chamada, vão ser associados, quando a mensagem é criada, os seguintes parâmetros:

- Origem – número da central origem da chamada;
- Destino – número da central destino da chamada;
- Duração – duração da chamada;
- Ocupação - número de circuitos (canais a 64 kbps) necessários às chamadas dessa classe em cada feixe, ou seja é permitido a uma chamada ocupar mais do que um canal simultaneamente (tráfego multicanal);
- Caminho desde a origem – caminho seguido pela mensagem da origem até à central corrente (inclusive);
- Caminho previsto – caminho previsto para a mensagem seguir, desde a próxima central até ao destino (inclusive);
- Caminho percorrido – caminho colocado nas mensagens que voltam para trás quer por *crankback*, impossibilidade ou conclusão de ligação. Este caminho indica as centrais por onde a mensagem passou até à última central a que a mensagem chegou;
- Classe – tipo de fluxo de tráfego;
- População Finita – indica se a mensagem é ou não de tráfego de população finita;
- Número da fonte – número da fonte que gerou tráfego, apenas com significado no caso de mensagens de tráfego de população finita.

A cada chamada, que corresponda a uma classe de tráfego de população finita vai ser associado também o seguinte parâmetro:

- Espera – que indica se uma fonte está inactiva (em espera) ou não.

Quando as mensagens são criadas, são atribuídos valores iniciais a todos os parâmetros. Alguns dos parâmetros são inicializados com valores que se mantêm durante toda a duração da simulação, entre os quais a classe, a informação sobre se é ou não tráfego de população finita e a ocupação (número de circuitos a ocupar pela chamada, o que é necessário para suportar tráfego multicanal); outros são

inicializados a nulo ou vazio. Alguns destes valores serão depois substituídos antes da mensagem ser enviada para a central correspondente.

É criado na fase de inicialização tráfego da central à qual cada gerador está ligado para todas as outras. Todo o tráfego restante é derivado deste, como veremos em seguida.

Terminado o estado Inicialização Gerador cada um dos geradores inicia o estado Processa uma Chamada. Os geradores permanecem neste estado até ao instante em que ocorra a terminação da simulação. Para auxiliar a descrição deste estado vamos utilizar o diagrama seguinte:

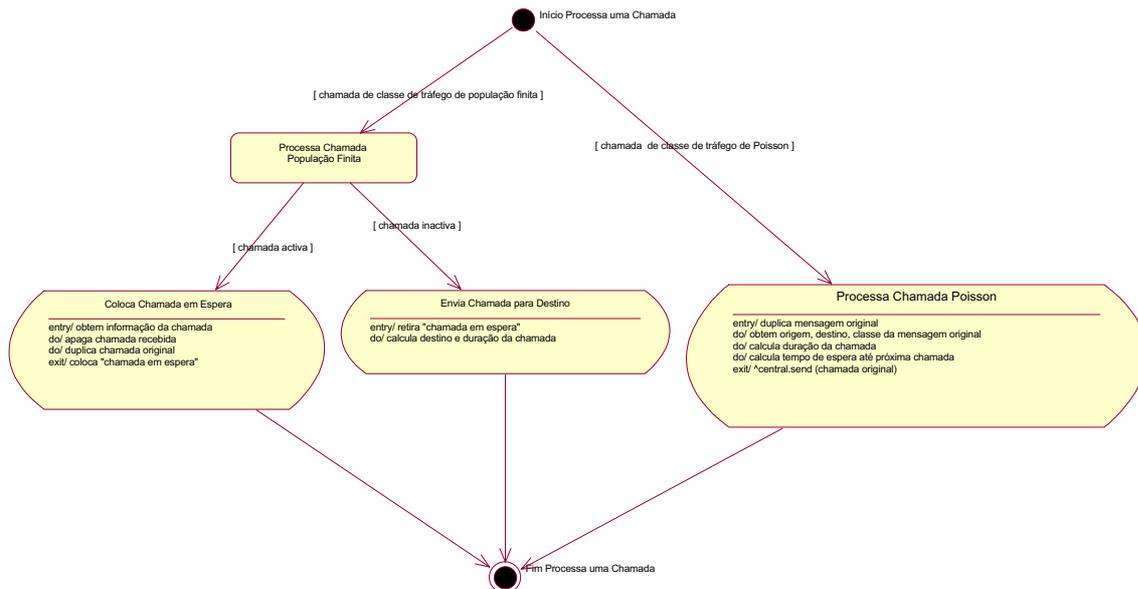


Fig. 5.16 – Diagrama de actividade do estado Processa uma Chamada.

No estado Processa uma Chamada cada gerador espera por uma mensagem (mensagem que corresponde sempre a uma chamada). Quando chegar uma chamada ao gerador este testa a mensagem para determinar se se trata de uma chamada de uma classe de tráfego de população finita ou se se trata de uma chamada de uma classe de tráfego de Poisson. Caso a chamada seja de uma classe de tráfego de população finita ela pode ter sido enviada pelo próprio ou pela central que está ligada ao gerador. A mensagem ter sido enviada pelo próprio gerador ocorre quando uma das fontes que estava inactiva esgotou o seu tempo de inactividade. A mensagem ter sido enviada pela central que está ligada ao gerador ocorre quando uma das fontes que estava activa acabou o seu tempo de serviço. Se a chamada que o gerador recebeu tinha sido enviada pela central (chamada activa) a próxima actividade, do gerador, é a actividade Coloca Chamada em Espera. Nesta situação as acções que é necessário realizar são as seguintes: determinar qual a chamada (fonte) que acabou o tempo de serviço; apagar a mensagem que foi recebida (porque a mensagem pode conter informação que não seja válida para a mensagem que vai ser escalonada); obter uma cópia da mensagem, correspondente à chamada que se recebeu, a partir das mensagens originais que temos guardadas; colocar o parâmetro espera com o valor verdadeiro (isto

indica que a chamada passa a estar inactiva). De seguida, a mensagem que se criou, é escalonada, isto é, o `gerador` envia a mensagem para si próprio. Esta mensagem vai ser recebida pelo próprio `gerador` com um atraso igual ao tempo que foi determinado, aleatoriamente, como sendo o tempo de inactividade da fonte. Se a chamada que o `gerador` recebeu tinha sido enviada por si (chamada inactiva) a próxima actividade, do `gerador`, é a actividade Envia Chamada para Destino. Neste caso as acções que é necessário realizar são as seguintes: colocar o parâmetro `espera` com o valor falso (isto indica que a chamada passa a estar activa); determinar o destino, (aleatoriamente, de acordo com as probabilidades cumulativas, que representam as proporções de tráfego da central origem para cada uma das centrais na rede); colocar o valor do destino determinado no parâmetro correspondente da mensagem; determinar a duração da chamada (utilizando um gerador de números aleatórios) e colocar esse valor no parâmetro correspondente da mensagem. De seguida o `gerador` vai enviar a mensagem para a `central` que está ligada a si e que corresponde à `central` origem para a chamada.

Caso a chamada seja de uma classe de tráfego de Poisson o `gerador` passa à realização da actividade Processa Chamada Poisson. Nesta situação o `gerador` tem que realizar as acções seguintes: criar uma mensagem igual à que recebeu; determinar quais os geradores de números aleatórios a usar, para isto é necessário saber qual é a origem, o destino e a classe da chamada; determinar o valor do tempo de duração para a chamada (utilizando um gerador de números aleatórios); actualizar o parâmetro `duração` da mensagem; determinar o tempo até ocorrer uma nova chamada (utilizando um gerador de números aleatórios diferente e independente do anterior); e enviar para a `central` que está ligada a si a chamada original. Para terminar o estado Processa uma Chamada, para o caso apresentado, falta apenas escalonar a mensagem que se criou no estado Processa Chamada Poisson, isto é, o `gerador` envia a mensagem para si próprio. Esta mensagem vai ser recebida pelo próprio `gerador` com um atraso igual ao tempo que foi determinado, aleatoriamente, como sendo o tempo até ocorrer uma nova chamada.

Pela análise da descrição anterior constata-se que para modelizar as várias classes de tráfego foi necessário recorrer a geradores de números aleatórios.

Para simular as chamadas de populações de tamanho infinito foi utilizada uma distribuição exponencial negativa tanto para gerar o instante da próxima chamada como para determinar o tempo de serviço de cada chamada.

Para simular as chamadas de populações de tamanho finito foi utilizada uma distribuição exponencial negativa para determinar o tempo de serviço de cada chamada. Para gerar o instante da próxima chamada podia ser seguida uma de duas abordagens seguintes:

1. Considerar uma “fonte” que gerasse tráfego equivalente ao gerado por todas as diferentes fontes. As chamadas geradas por essa “fonte” equivalente não seguem nenhuma distribuição fixa, pois a intensidade de chamadas depende do número de fontes livres;

2. Considerar as diversas fontes separadamente. Neste caso para determinar o instante da próxima chamada (após a anterior ter concluído) pode ser usada a distribuição exponencial negativa. É necessário utilizar um número de variáveis aleatórias, associadas a essa distribuição, igual ao número de fontes (indivíduos da população).

Optou-se pela segunda por ser mais simples de implementar. Como pode ver-se no apêndice D os resultados obtidos por simulação, considerando este princípio de geração de chamadas, são semelhantes aos obtidos pelo método analítico, como era esperado.

O processo `gerador` recorre a números aleatórios obtidos directamente de geradores de números aleatórios e também a números aleatórios obtidos de distribuições. O OMNeT++ possuía, já implementados, modos de gerar amostras para as distribuições que precisamos de usar.

Devido ao elevado número de geradores necessários na geração de chamadas de população finita o número de geradores de números aleatórios independentes disponíveis na linguagem de simulação (32 por omissão) foi aumentado para 4096, alterando o código fonte do OMNeT++ (nas definições usadas pelo núcleo do simulador). O código do OMNeT++ possui uma ferramenta geradora de sementes que permite escolher sementes iniciais para os geradores, separadas entre si de um número fixo (escolhido pelo utilizador) de forma a que os ciclos dos geradores não coincidam. Como o ciclo do gerador é de  $2^{31}-2$ , mesmo com mais geradores podemos separar as sementes entre si de cerca de 500000 valores sem problemas. Embora tal não fosse estritamente necessário, procedeu-se também a uma alteração no código da ferramenta geradora de sementes, de forma a tornar mais rápida a geração de sementes em certos casos.

### 5.5.3 Componentes gerais para todos os encaminhamentos

O simulador foi construído de modo a permitir com facilidade simular sistemas com uma grande diversidade de características, para isto contribuiu a forma modular estruturada que foi usada na implementação, tirando partido da orientação a objectos permitida pela linguagem C++.

Na figura seguinte apresenta-se um diagrama com as principais classes do modelo de simulação que foi construído, designadamente classes `controlador`, `router`, `stats`, `gerador` e `central`. As classes que correspondem ao processo `gerador` e ao processo `central` são respectivamente as classes `gerador` e `central`. As restantes classes, que correspondem aos componentes `controlador`, `encaminhador` e `estatísticas` são respectivamente as classes `controlador`, `router` e `stats`. Estas últimas são substituídas por suas especializações para implementar cada método de encaminhamento em particular, enquanto as primeiras fornecem suporte a todos os tipos de métodos de encaminhamento, como referido anteriormente.

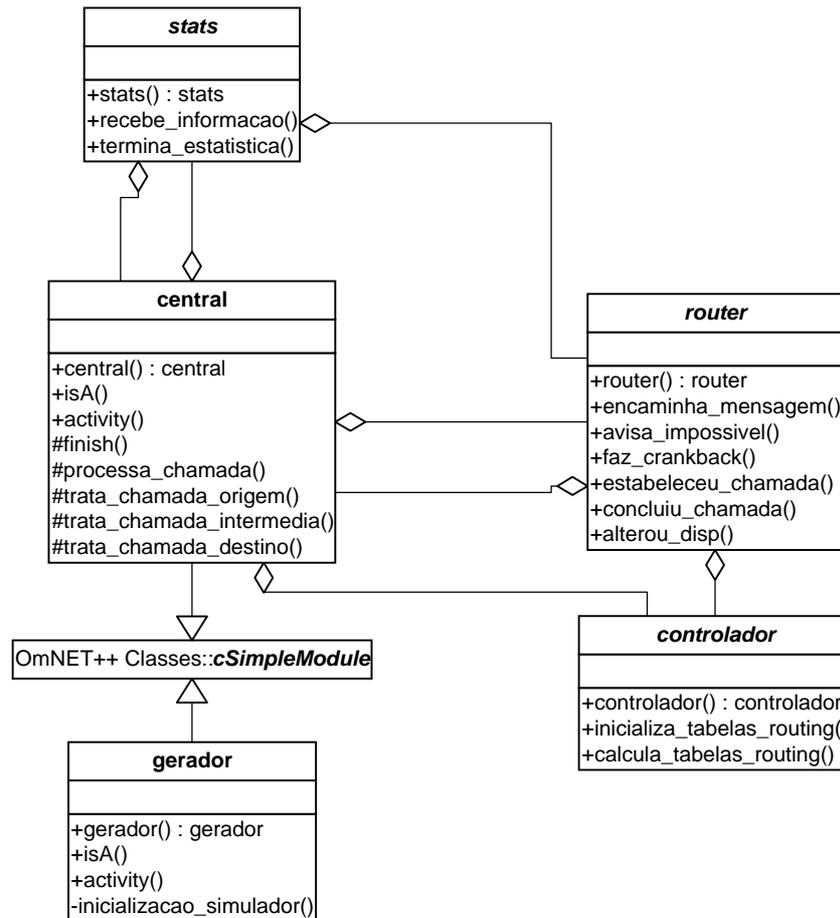


Fig. 5-17 – Diagrama das classes principais do modelo.

Tanto a classe gerador como a classe central são classes que correspondem a módulos simples definidos no modelo, derivadas por isso de uma classe (cSimpleModule) da biblioteca de classes de simulação do OMNeT++. A classe gerador e a classe central herdam desta algumas propriedades, de entre as quais as funções activity e finish, que redefinem.

O código das funções activity dos vários processos (central e gerador) do modelo são executados em paralelo, desde o início da simulação até ao fim da simulação. É portanto nestas funções, de cada processo, que está definido o código que corresponde a todas as operações que cada processo deve realizar durante a simulação.

O processo central possui uma função (finish) que é executada automaticamente quando a simulação termina. Foi por esta razão o lugar escolhido para pedir o registo dos resultados da simulação em ficheiros.

Como os funcionamentos dos processos central e gerador foram descritos na secção anterior, não se vai aqui descrever a actividade que os processos realizam durante toda a simulação.

O programa de simulação (incluindo o que diz respeito aos métodos de encaminhamento) foi construído a partir de vários tipos de componentes. São eles os componentes comuns a todos os tipos de encaminhamento e os componentes específicos de cada método de encaminhamento. Os componentes comuns a todos os tipos de encaminhamento são, para além do `gerador` e da `central`, os designados de componentes base e ainda outros componentes (que são usados para funções específicas de suporte aos componentes base). Os componentes específicos de cada método de encaminhamento são os componentes derivados dos componentes base e os componentes auxiliares.

A sua descrição detalhada é apresentada na secção E.1 do apêndice E.

### 5.5.3.1 Componentes base

Estes componentes fornecem as funções necessárias para o encaminhamento das chamadas e para o estudo do seu comportamento, sendo estas funções chamadas pelo processo `central`. Cada um dos componentes (controlador, encaminhador e estatísticas) deve ser redefinido para cada método de encaminhamento, ou mais precisamente, “sobreposto” por outro código que redefine algumas funções destes, através do polimorfismo permitido pela orientação a objectos. Esta sobreposição é conseguida na linguagem utilizada, C++, através do processo de herança. Assim são definidos novos componentes (por exemplo `controlador_dar`) que sendo derivados da classe base (neste caso `controlador`) herdam destas um conjunto de propriedades e redefinem um conjunto de métodos, de forma a implementar o comportamento desejado.

#### 5.5.3.1.1 Controlador

O controlador vai modelar o comportamento do processador (centralizado ou distribuído) nos métodos de encaminhamento que o usam. O controlador vai obter informação de variáveis de estado e de acordo com o resultado do processamento dessa informação resultará a actualização da forma de encaminhamento das chamadas. O controlador tem essas funções apenas para os métodos de encaminhamento onde haja necessidade de actualizar as tabelas de encaminhamento. Nestes casos a actualização ocorre durante toda a simulação, com intervalos mais ou menos longos, dependendo da frequência com que é necessário actualizar a informação a utilizar no encaminhamento das chamadas.

O controlador fornece também as funções necessárias para inicializar as tabelas de encaminhamento, o que ocorre apenas uma vez, no início da simulação.

Ver os detalhes da implementação na secção E.1.1.1 do apêndice E.

#### 5.5.3.1.2 Encaminhador

A função principal do encaminhador é indicar à `central` o modo como ela deve tentar encaminhar uma chamada que tenha lá chegado. Isto é, cada `central` (origem ou intermédia) espera receber a

identificação da `central` (intermédia ou destino) para onde deve enviar a chamada e possivelmente também o caminho completo que a chamada deve seguir até à `central` destino; se a identificação que recebeu não corresponder a uma `central` válida então é porque o encaminhador não conseguiu determinar uma `central` para onde seja possível enviar a chamada de modo a que ela chegue ao destino.

Além desta o encaminhador deve desempenhar outras funções: quando a chamada se encontra numa `central` intermédia e se verifica que não é possível encaminhar a chamada pelo caminho que estava a ser tentado, a `central` solicita o encaminhador para que lhe seja indicado se a chamada pode tentar outro caminho alternativo; se o encaminhador indicar que ainda pode ser tentado outro caminho alternativo a chamada deve retornar por *crankback* para a `central` origem para de seguida tentar outro caminho alternativo; caso contrário, se o encaminhador indicar que não pode ser tentado outro caminho alternativo a chamada deve retornar para a `central` origem como chamada que não foi possível estabelecer.

O encaminhador pode ser solicitado pela `central` onde a chamada teve origem, como já referido, mas pode também ser chamada por uma `central` intermédia que seja usada pela chamada no seu encaminhamento. Uma das situações em que ocorre esta segunda possibilidade é quando a chamada chega a uma `central` intermédia sem conhecer o caminho que deve seguir, isto é, não sabe para que `central` deve seguir para chegar ao destino.

Ver os detalhes da implementação igualmente no apêndice E, na secção E.1.1.2.

### **5.5.3.1.3 Estatísticas**

O objectivo principal do modelo de simulação implementado é a obtenção de parâmetros estatísticos que exprimam os resultados do seu comportamento ao longo de cada corrida do simulador.

O componente estatísticas fornece as funções de recolha de informação (chamadas a cada momento pelos processos `central`), e algumas funções de consolidação de informação (chamadas no fim da simulação para apresentar a informação consolidada). Outras análises aos dados da simulação podem também ser realizadas no exterior do simulador.

O componente estatísticas, solicitado pelo processo `central`, permite armazenar vários valores e com esses valores produzir e registar os resultados da simulação em ficheiros. Os valores armazenados são actualizados pelo componente estatísticas sempre que ocorre uma das acções seguintes:

- Início de uma chamada (sendo a função chamada pela `central` origem);
- Uma chamada foi concluída com sucesso (sendo a função chamada pela `central` destino);
- Uma chamada foi concluída sem sucesso (sendo a função chamada pela `central` origem);

- Uma chamada vai tentar um caminho alternativo após ter ocorrido *crankback* (sendo a função chamada pela `central` origem);
- Uma chamada foi concluída sem sucesso depois de ter ocorrido *crankback* (sendo a função chamada pela `central` origem);
- Ocupação de circuitos de um feixe (sendo a função chamada pela `central` origem e pela `central` intermédia);
- Libertação de circuitos de um feixe (sendo a função chamada pela `central` intermédia e pela `central` destino);
- Um feixe foi tentado sem sucesso (sendo a função chamada pelo `router` associado à `central`).

Os pormenores da implementação são apresentados na secção E.1.1.3 do apêndice E.

### 5.5.3.2 Outros componentes

Componentes de apoio chamados pelos processos `central` e/ou pelo processo `gerador`, armazenam informação, fornecem informação e geram os dados manipulados pelo sistema.

Nos dois diagramas das figuras seguintes, figura 5-18 e figura 5-19, estão representadas as classes auxiliares que foram utilizadas no modelo. As classes que correspondem aos componentes caminhos e armazém de dados são respectivamente as classes com os nomes `caminhos` e `arm_dados` e estão representadas na figura 5-18. As classes que correspondem aos componentes reserva e tráfego são respectivamente as classes com os nomes `rr` e `classe_trafego` e estão representadas na figura 5-19. Repare que nessa figura aparece também um tipo de dados designado por `MPTR` (usado pelo processo `gerador` na declaração da variável que armazena cópias das mensagens, do tipo “chamada a estabelecer”, para os tráfegos de população finita) que permite aceder uma `cMessage`.

Ver os detalhes da implementação destes componentes na secção E.1.2 do apêndice E.

#### 5.5.3.2.1 Caminhos

Quando se pretende armazenar um caminho este é armazenado de forma codificada. O componente `caminhos` é quem trata de fazer a codificação e descodificação dos caminhos que estão armazenados numa chamada e na tabela de encaminhamento.

Podem-se ver os detalhes da implementação na secção E.1.2.1 do apêndice E.

### 5.5.3.2 Armazém de dados

O componente armazém de dados guarda a informação comum a todos os processos sendo esta informação partilhada por todos os processos (utilização de memória partilhada).

A informação armazenada neste componente é: o número de circuitos disponíveis e o número de circuitos existentes para cada feixe da rede, o número de circuitos reservados e o número de circuitos planeados em cada feixe da rede para cada uma das classes. O número de circuitos planeados para uma classe em cada feixe ou por outras palavras a largura de banda mínima garantida para uma classe quando existe bloqueio é uma fracção dos circuitos existentes no feixe que é atribuída a essa classe quando as condições de bloqueio, das chamadas dessa classe, o justificam.

Podem-se ver outros pormenores da implementação na secção E.1.2.2 do apêndice E.

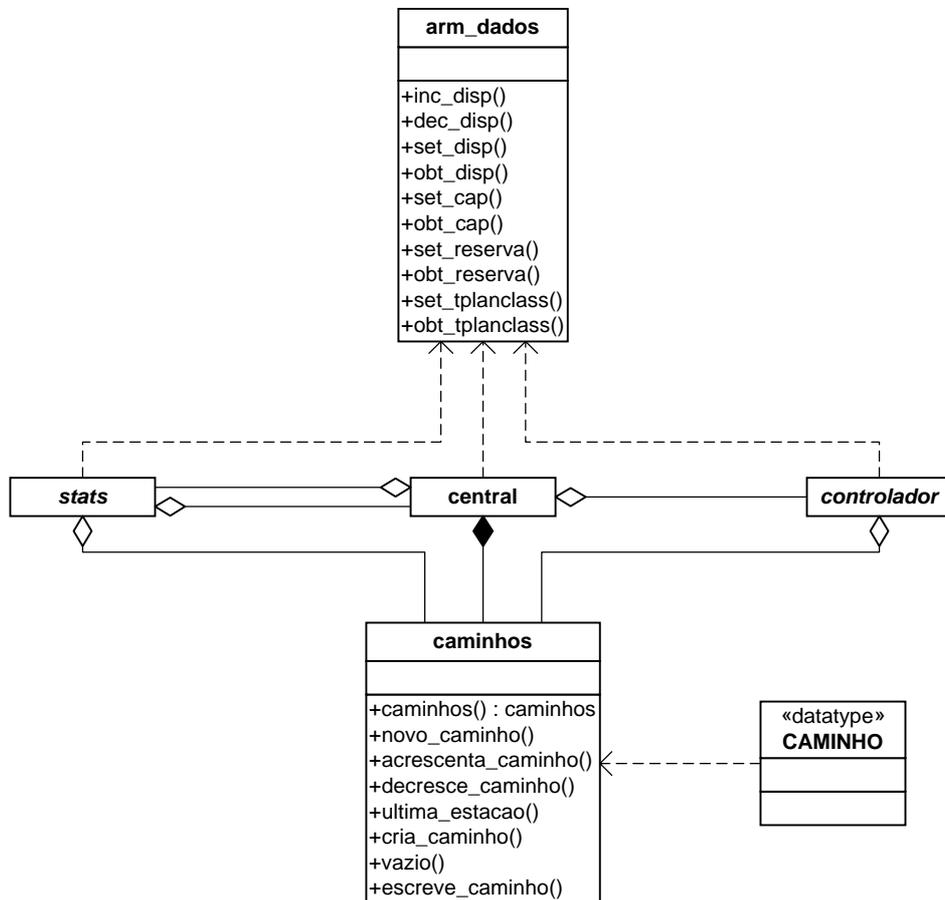


Fig. 5-18 – Diagrama das classes auxiliares do modelo, usadas pelo processo central.

### 5.5.3.2.3 Tráfego

O componente tráfego guarda a informação relativa a uma classe de tráfego.

Ao componente tráfego corresponde uma estrutura chamada `classe_trafego`. Nesta estrutura é armazenado, para cada classe, o número de circuitos necessários às chamadas, em cada feixe, a duração (tempo de serviço) das chamadas e a indicação se o tráfego é ou não de população finita (Engsetiano).

#### 5.5.3.2.4 Reserva de números aleatórios

O componente reserva de números aleatórios é utilizado para fazer a gestão dos geradores de números aleatórios independentes. Como são usados um grande número de geradores durante a simulação, foi necessário implementar este código para facilitar a tarefa de atribuição de geradores, de forma a garantir que o mesmo gerador não seja utilizado em mais do que uma situação de geração de números, causando dependências inadvertidas entre geradores supostamente independentes. A utilização de reserva de geradores, garante assim a atribuição de geradores de números aleatórios independentes em situações que o requeiram. A classe que corresponde a este componente é a classe `rr`.

Podem-se ver os detalhes da implementação deste componente na secção E.1.2.4 do apêndice E.

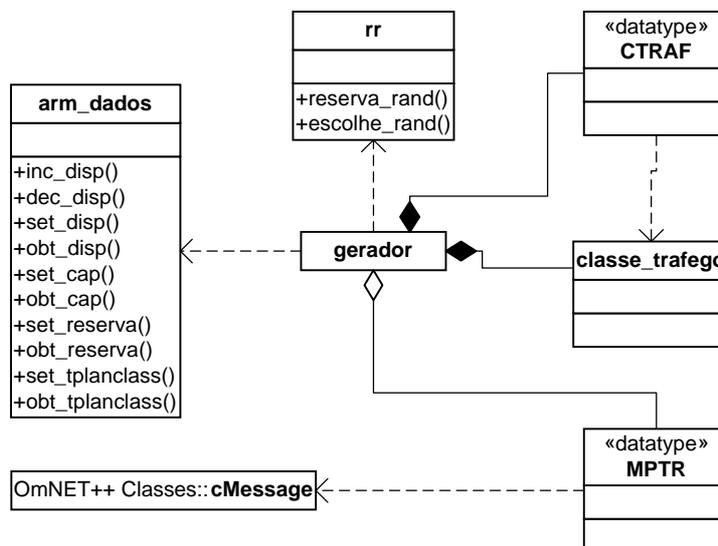


Fig. 5.19 – Diagrama das classes auxiliares do modelo, usadas pelo processo gerador.

### 5.5.4 Componentes específicos de cada encaminhamento

Pode-se consultar no apêndice E, secção E.2 a descrição detalhada dos componentes específicos de cada tipo de encaminhamento implementado:

- Directo
- FAR (Fixed Alternative Routing)
- DAR (Dynamic Alternative Routing)
- DCR (Dinamically Controlled Routing)
- RTNR (Real Time Network Routing)

Entretanto iremos referir aspectos particularmente importantes relativos à forma como foram implementados alguns destes métodos, nomeadamente o DAR, o DCR e o RTNR, envolvendo algumas diferenciações em relação à descrição original apresentada no capítulo 3 sobre métodos de encaminhamento dinâmicos.

### 5.5.4.1 DAR

Este método foi implementado de forma a permitir o encaminhamento simultâneo de vários tipos de tráfego. Para o conseguir fez-se uma generalização do método de encaminhamento original, passando o algoritmo de encaminhamento a ser o seguinte: quando uma chamada, de uma determinada classe chega ela é oferecida primeiro ao caminho directo, se este não tiver capacidade disponível suficiente para a chamada em questão, esta transborda para o caminho alternativo, de dois feixes, actualmente seleccionado para as chamadas dessa classe entre essa origem/destino. O caminho alternativo permanece inalterado a menos que a chamada seja bloqueada nele. Neste caso um novo caminho alternativo é seleccionado aleatoriamente para as chamadas subsequentes, da classe em questão. O caminho alternativo está sujeito a reserva de circuitos para o encaminhamento Directo. As diferenças em relação ao algoritmo original são: a necessidade de verificação, sempre que uma chamada chega, da classe a que pertence para determinar a quantidade de recursos de que precisa e a necessidade de armazenar tantos caminhos alternativos, por par origem/destino, como o número de classes de tráfego que existam. A implementação desenvolvida para este algoritmo tem como caso particular o algoritmo do método de encaminhamento DAR descrito no capítulo 3 (onde era considerado apenas o encaminhamento de um tipo de tráfego).

Na implementação feita, além do mecanismo de protecção de tráfego de primeiro encaminhamento, descrito no método de encaminhamento original, foi implementado um segundo mecanismo. Os dois tipos de mecanismo de protecção do tráfego de primeiro encaminhamento implementados foram:

- Reserva de Circuitos Fixa - um número de circuitos, fixo, é reservado, em cada feixe, para ser utilizado no encaminhamento Directo, tal como descrito no método de encaminhamento original.
- Reserva de Circuitos Dinâmica - um número de circuitos, variável com o tempo, é reservado, em cada feixe, para ser utilizado apenas no encaminhamento directo de chamadas, quando o número de circuitos livre for igual ou inferior a esse número. O número de circuitos reservados em cada feixe, para cada instante, é calculado usando uma expressão simplificada, obtida a partir da expressão para o cálculo do número de circuitos reservados no método de encaminhamento RTNR, tal como descrito no capítulo 3. A expressão simplificada é:

$$RBWtraf_k = \sum_{i=VN_1}^{VN_N} R_k^i \times r_i \quad (5.1)$$

onde a largura de banda total reservada  $RBWtraf_k$ , é calculada como o somatório dos níveis de reserva para cada classe  $R_k^i$  multiplicado pela largura de banda média necessária para essa classe,  $r^i$ . Este somatório soma desde a classe de serviço considerada até à classe de serviço que necessite de maior largura de banda.

Sempre que se pretende simular este método deve-se escolher qual dos dois mecanismos se pretende utilizar.

Se for escolhida reserva de circuitos dinâmica além do mecanismo usual de protecção do tráfego foi incorporado um mecanismo de protecção de modo a permitir uma utilização equilibrada dos recursos pelos vários serviços (ver capítulo 2). O tráfego das várias classes está então sujeito a um mecanismo de controlo de acesso ao caminho directo. O mecanismo de controlo de acesso implementado foi o descrito no método de encaminhamento RTNR.

O algoritmo implementado foi por isto tornado ainda mais geral do que o indicado pela descrição inicial. Se for escolhida reserva de circuitos dinâmica, o algoritmo tem que determinar periodicamente os valores para as reservas. E além disso, para encaminhar uma chamada no caminho directo não basta verificar se possui capacidade disponível suficiente para ela, devido ao mecanismo de controlo de acesso referido anteriormente.

Podem-se ver os detalhes da implementação na secção E.2.4 do apêndice E.

### 5.5.4.2 DCR

Este método de encaminhamento foi implementado tal como definido no capítulo 3. No entanto o número de circuitos gastos por uma chamada, em cada feixe do caminho, pode ser um qualquer valor que se escolha, não está portanto limitado a chamadas que ocupem apenas um circuito.

Pode ser escolhido um de três tipos de reserva de circuitos, são eles:

- Reserva fixa;
- Reserva dinâmica, conseguida utilizando a expressão:

$$m_s(t) = g \times a_s' \quad (5.2)$$

onde  $m_s(t)$  corresponde à reserva de circuitos no instante  $t$  no grupo de circuitos  $s$ ,  $g$  corresponde a um factor de escala e  $a_s'$  corresponde à quantidade corrente de tráfego, oferecido pela primeira vez, que transborda do feixe  $s$ .

- Reserva dinâmica, como definido no método de encaminhamento DAR, determinada pela expressão 5.1.

Os detalhes da implementação estão apresentados na secção E.2.5 do apêndice E.

### 5.5.4.3 RTNR

Recomenda-se que, dada a complexidade deste método de encaminhamento, antes de analisar a descrição da implementação que se segue, seja revista a apresentação feita no capítulo 3 do método de encaminhamento RTNR com várias classes de tráfego.

O RTNR foi implementado de acordo com essa descrição. No entanto é permitido, como já aconteceu nos métodos DCR e DAR, escolher outro tipo de mecanismo de protecção do tráfego de primeiro encaminhamento. O tipo permitido, além do definido no capítulo 3, é a reserva de circuitos dinâmica como a utilizada na implementação do DAR, dada pela expressão 5.1. Note que este tipo de reserva, como referido no DAR, é uma simplificação ao tipo de reserva dinâmica definida no método de encaminhamento RTNR original. Deve no entanto ser notado que a simplificação nos cálculos desta reserva não leva necessariamente a uma grande simplificação do método como um todo, uma vez que as decisões de encaminhamento fazem uso dos valores que na simplificação deixaram de ser utilizados, pelo que o seu cálculo pelas centrais continua a ser necessário.

Os detalhes da implementação deste método estão apresentados na secção E.2.6 do apêndice E.

### 5.5.5 Estrutura global da implementação (Diagrama de classes)

O diagrama de classes, da figura 5-20, serve para dar uma visão geral de todas as classes que foi necessário implementar e também algumas das ligações que foram criadas entre elas.

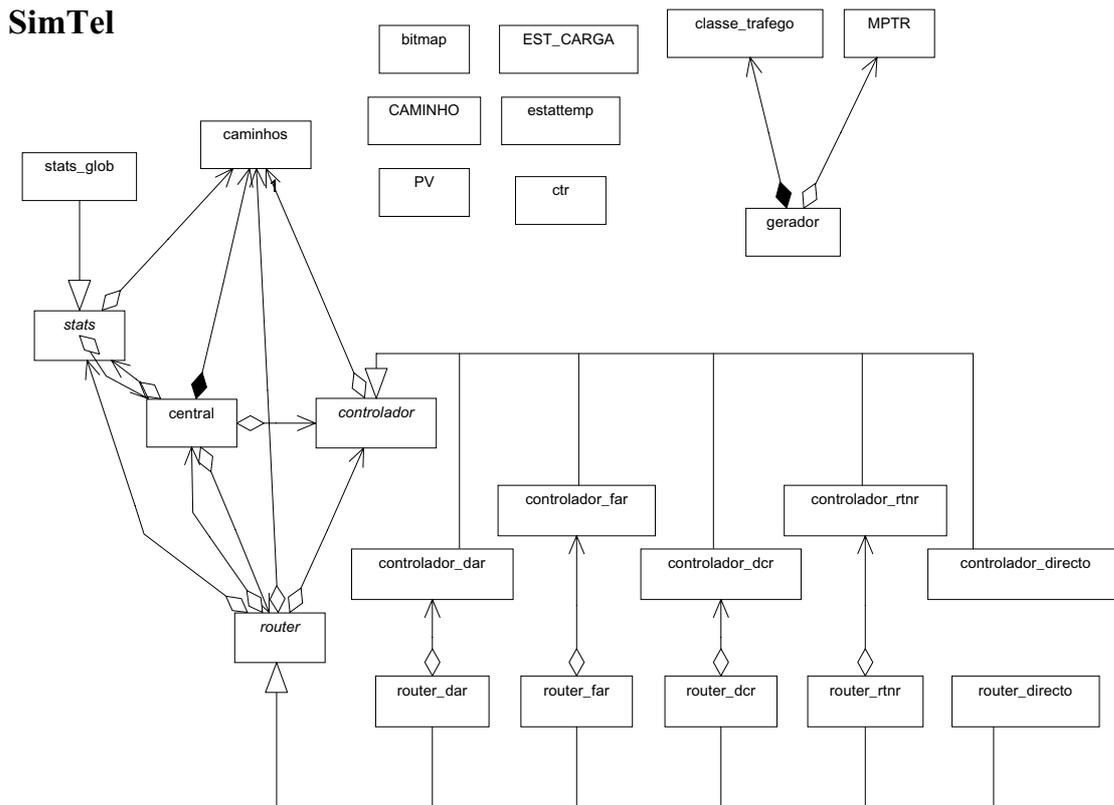


Fig. 5-20 - Diagrama de todas as classe definidas para a implementação.

Para concluir a descrição feita em relação à implementação dos métodos de encaminhamento vamos agora fazer uma analogia entre as classes principais implementadas no modelo com os componentes do sistema. Assim sendo, a classe `controlador` desempenha as mesmas funções que o processador no sistema. No caso dos métodos isolados ou distribuídos temos um `controlador` associado a cada `central` desempenhando as funções de processador local. No caso dos métodos centralizados é apenas um dos `controladores` que possui funcionalidades desempenhando esse as funções de processador central. A classe `central` juntamente com a classe `router` desempenham o papel das centrais de comutação no sistema. A determinação do caminho para encaminhar uma chamada é da competência do `router`. A função principal da classe `central` é levar as chamadas, que recebe, até ao destino se possível, para isto recorre às funções da classe `router`. Como pretendíamos ter a possibilidade de seleccionar um de vários algoritmos na determinação dos caminhos, derivamos, a partir da classe `router`, a classe `router_directo`, a classe `router_far`, a classe `router_dar`, a classe `router_dcr` e a classe `router_rtnr`. Cada uma destas classes implementa um dos algoritmos de encaminhamento. Como a informação que cada algoritmo utiliza e o modo como é determinada difere de algoritmo para algoritmo derivamos também, a partir da classe `controlador`, a classe `controlador_directo`, a classe `controlador_far`, a classe `controlador_dar`, a classe `controlador_dcr` e a classe `controlador_rtnr`. Pormenores sobre estas classes podem ser encontrados na secção E.2 do apêndice E.

No método de encaminhamento Directo não é preciso qualquer processamento de informação pois o encaminhamento é baseado apenas na capacidade disponível dos caminhos directos. Neste caso os controladores não desempenham qualquer função.

No método de encaminhamento FAR o controlador tem funções apenas na fase de inicialização (carregamento das tabelas de encaminhamento), pois a partir daí a informação a usar no encaminhamento permanece inalterada.

O método de encaminhamento DAR é isolado logo os controladores associados a cada `central` fazem a actualização da informação, a usar no encaminhamento, baseada apenas na informação armazenada nele.

No DCR, como é um método de encaminhamento centralizado, só foram atribuídas funcionalidades a um dos controladores.

Por último o RTNR é um método de encaminhamento distribuído logo temos um controlador associado a cada `central` que utiliza a informação dos outros controladores.

Por questões de simplificação ou para evitar a duplicação de informação não se manteve, por vezes, a analogia entre o local onde a informação a usar no encaminhamento é armazenada no sistema, e o local onde é armazenada no modelo; contudo os resultados não são alterados com isso. Isto passa-se tanto no RTNR como no DCR.

## **5.6 Execução de uma simulação**

### **5.6.1 Entradas para uma simulação**

A principal entrada da simulação consiste nos ficheiros que definem as variáveis descritivas dos componentes da rede (isto é, a topologia da rede). As entradas restantes consistem nos parâmetros do modelo e da corrida.

#### **5.6.1.1 Configuração da rede (ficheiro de topologia)**

A configuração da rede de comutação por circuitos, que se pretende usar, isto é, o número de circuitos do feixe que liga quaisquer duas centrais é especificado num ficheiro de texto a que chamaremos ficheiro de topologia.

No apêndice C, secção C.1, podemos encontrar um exemplo deste ficheiro, bem como uma exposição da sua estrutura.

### 5.6.1.2 Ficheiro de configuração (ficheiro *ini*)

É permitida grande flexibilidade nos vários algoritmos de encaminhamento implementados em parte pelo uso de parâmetros. O ficheiro de configuração pode descrever várias corridas de simulação, em que cada corrida pode ter os seus parâmetros. No ficheiro de configuração podem também ser seleccionadas as sementes para os geradores de números aleatórios.

Para geração do ficheiro de configuração foi desenvolvido em Perl uma ferramenta chamada `gerador_ini.pl`, que constrói um ficheiro *ini* a partir de um ficheiro mais simples. Agradeço por esta implementação ao Eng. Paulo Melo.

Exemplos do ficheiro base e do ficheiro *ini* por ele gerado, bem como uma descrição da estrutura dos mesmos podem ser encontrados no apêndice C, secção C.2.

### 5.6.2 Os vários passos para executar uma simulação

O programa de simulação foi desenvolvido em ambiente Linux. É tendo por base este ambiente que vão ser descritos os passos para executar a simulação.

Depois de termos o modelo construído, isto é, depois de termos feito a descrição da topologia do modelo (em linguagem NED) e de termos implementado em C++ os módulos simples estamos em condições de criar o programa de simulação executável. O primeiro passo para criar o programa executável é utilizar o compilador NEDC para compilar o ficheiro *ned* e então obter o ficheiro correspondente em C++. O que é feito automaticamente ao executar o comando `make` referido a seguir.

O passo seguinte é compilar todos os ficheiros C++ (ficheiro *ned* compilado e ficheiros correspondente à implementação dos módulos simples) e fazer a ligação deles com o núcleo de simulação, com a biblioteca de classes do OMNeT++ e com a biblioteca do interface com o utilizador. Deve começar-se por executar o comando:

```
makemake -u cmdenv
```

A utilização da opção `-u` é que torna possível a especificação do interface com o utilizador (`cmdenv` no exemplo) que se pretende usar. Se a opção `-u` não for usada o interface por omissão é o `Tkenv`.

O nome do programa executável que será produzido é o nome da directoria onde temos os códigos fontes, se pretendermos que o programa executável fique com outro nome deve utilizar-se a opção `-o`. Se o comando executado for por exemplo:

```
makemake -o simtel
```

Neste caso o nome do programa executável será `simtel`.

Ao executar o comando `makemake` foi produzido o ficheiro `makefile` após o que deve ser executado o comando `makedepend` e por último o comando `make`. Com o comando `make` é que será efectuada a compilação e ligação das bibliotecas de rotinas e então produzido o programa executável. Note que se depois de executado o comando `make` houver necessidade de alterar o código de algum ficheiro ao executar de novo o comando `make` só serão recompilados os ficheiros dependentes do ficheiro alterado e o próprio.

Para poder executar o programa de simulação é necessário ter o ficheiro de configuração (denominado de ficheiro *ini*) construído.

Para aliviar a fastidiosa tarefa de construção dos ficheiros *ini*, para muitas replicações de uma simulação, foi desenvolvido um programa, em Perl, chamado `gerador_ini.pl`, que constrói um ficheiro *ini* a partir de um ficheiro mais simples. A saída do programa é o ficheiro de saída standard; para obter um ficheiro *ini* a saída desse programa deve ser redireccionada.

O ficheiro de configuração *ini* deve ter o nome `omnetpp.ini`, caso contrário é necessário indicar o nome do ficheiro de configuração, ao executar o programa de simulação, utilizando para tal a opção `-f`. Se, por exemplo, o ficheiro de configuração e o programa executável tiverem o nome `teste.ini` e `simtel`, respectivamente, para correr o programa de simulação devia-se executar o comando:

```
simtel -f teste.ini
```

Após o arranque da aplicação, quando é usado o interface Tkenv, é apresentada a janela inicial do interface gráfico (figura 5-21).

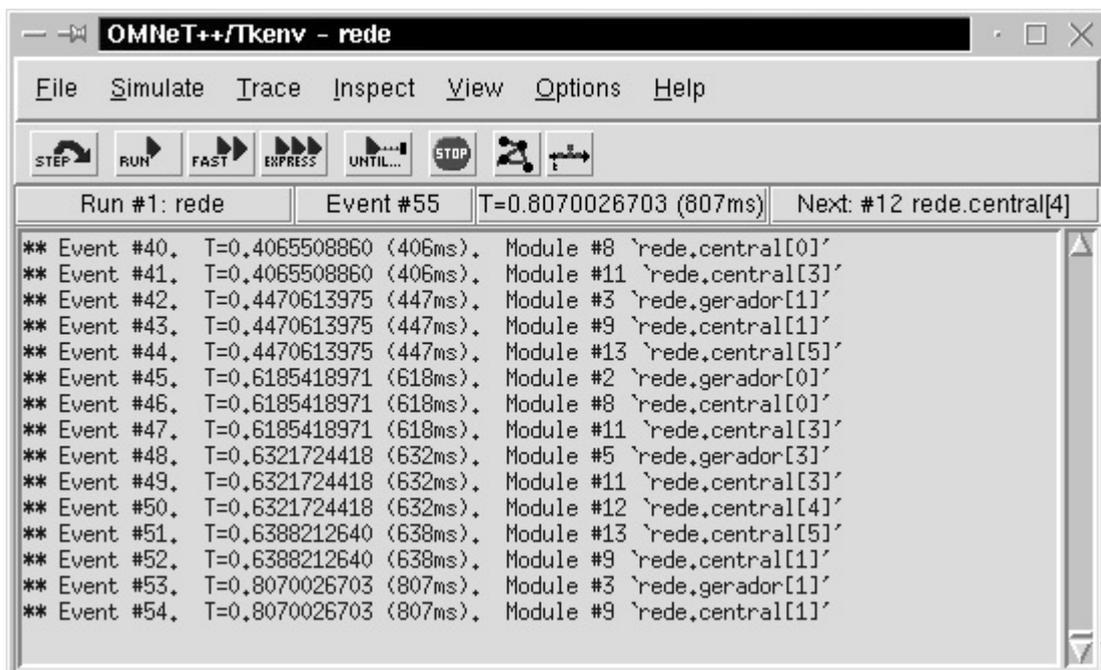


Fig. 5-21 - Janela principal da aplicação.

Este interface permite executar a simulação e verificar os valores presentes nos diversos componentes do modelo a simular. Para além dos menus, que permitem efectuar diversas operações, possui também uma barra de botões, que podem ser usados para aceder mais facilmente às funções mais usuais.

Assim, nesta barra podemos encontrar as opções de controlo da simulação que permitem a execução passo a passo (*step*), a velocidade baixa com total animação gráfica (*run*), a velocidade mais rápida com actualizações de animação pouco frequentes (*fast*) ou mesmo execução sem animação (*express*). É também possível executar a simulação até à ocorrência de determinada condição (*until*). Seja qual for a velocidade de execução seleccionada, pode ser interrompida a simulação através do botão de parar (*stop*), podendo depois prosseguir-se com a simulação a partir desse ponto.

Na barra de botões estão ainda presentes opções para a visualização da rede (que permite a representação da rede graficamente, como apresentado na figura 5-23), e para a visualização da lista de mensagens escalonadas (que corresponde aproximadamente à lista de acontecimentos futuros). O corpo da janela principal é também o local onde as mensagens de aviso (enviadas através do interface *ev*) e as marcas de acontecimento são apresentadas por omissão, embora seja possível redireccionar as mensagens geradas por cada módulo para janelas individuais.

Aquando do início da simulação, o interface permite-nos a escolha da corrida a executar, de entre todas as que estejam predefinidas no ficheiro *ini*, através da janela apresentada na figura 5-22.



Fig. 5-22 - Arranque da simulação, escolha da corrida a executar.

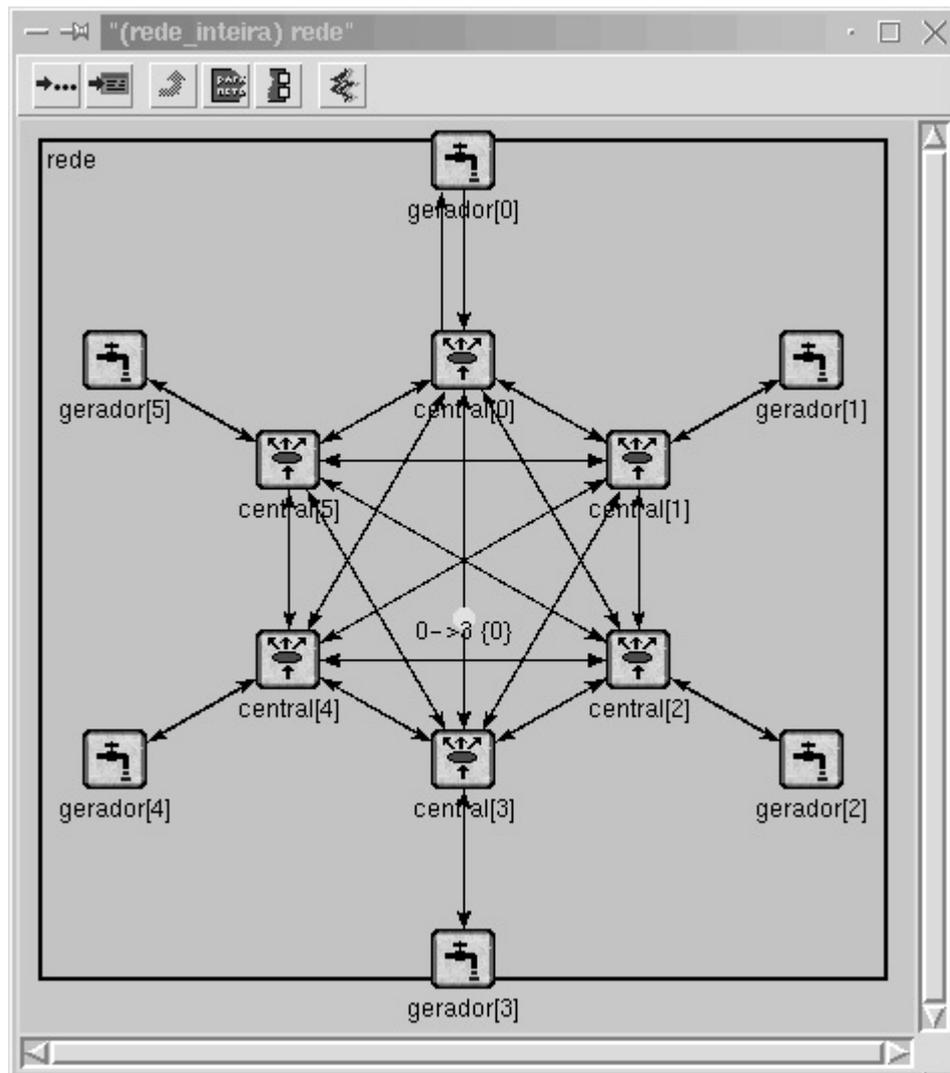


Fig. 5-23 - Um exemplo de rede a simular.

A apresentação gráfica da rede a simular, para além de permitir a visualização animada das trocas de mensagens entre módulos (notar na figura a mensagem a ser enviada da *central 0* para a *central 3*) permite-nos também acesso às informações específicas sobre cada um dos módulos e componentes nela representados, bastando para tal efectuar um duplo clique sobre o elemento que se pretenda observar, quer durante o período de corrida da simulação, quer ela tenha sido interrompida (mas não antes do início da corrida ou após o término da mesma).

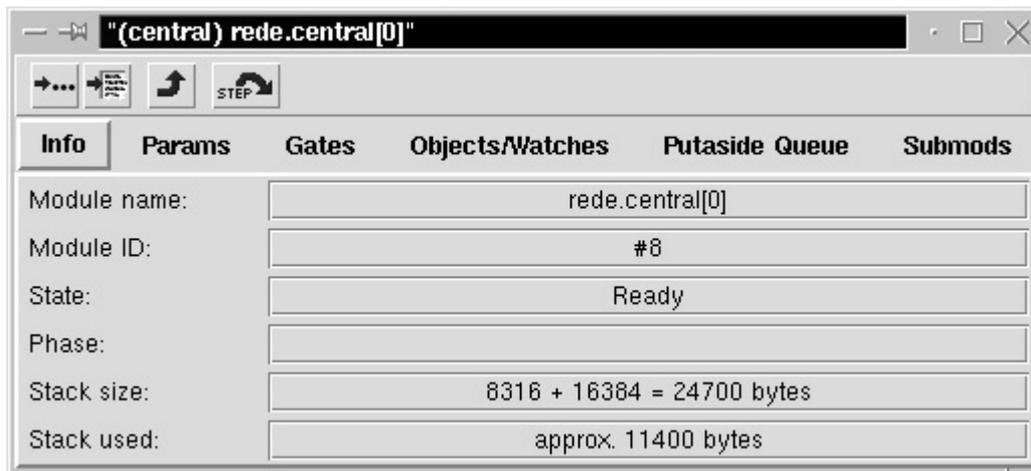


Fig. 5-24 - Observação de um módulo da simulação – informação genérica.

Cada módulo presente na simulação pode ser analisado durante a mesma quer em função das suas características básicas (figura 5-24), quer através dos valores dos seus parâmetros (que podem ser alterados durante a execução, figura 5-25), quer através da visualização das suas portas de saída e entrada (figura 5-26). Ainda outros valores podem ser analisados durante a execução (é possível definir valores a serem monitorizados (*watches*), e observar submódulos ou a fila auxiliar de mensagens do módulo, mas neste trabalho tais capacidades não foram utilizadas).

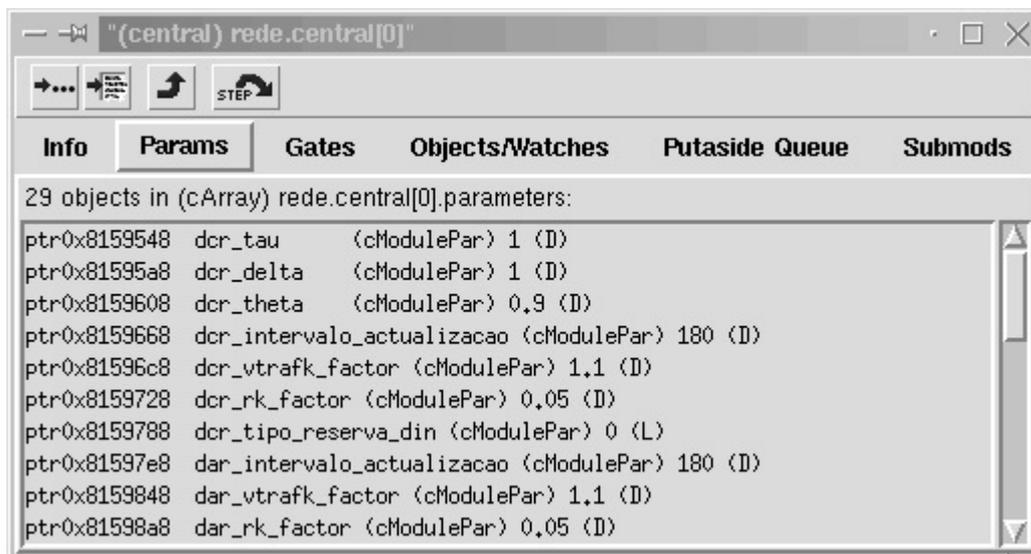


Fig. 5-25 - Observação de um módulo da simulação – parâmetros do módulo e seus valores.

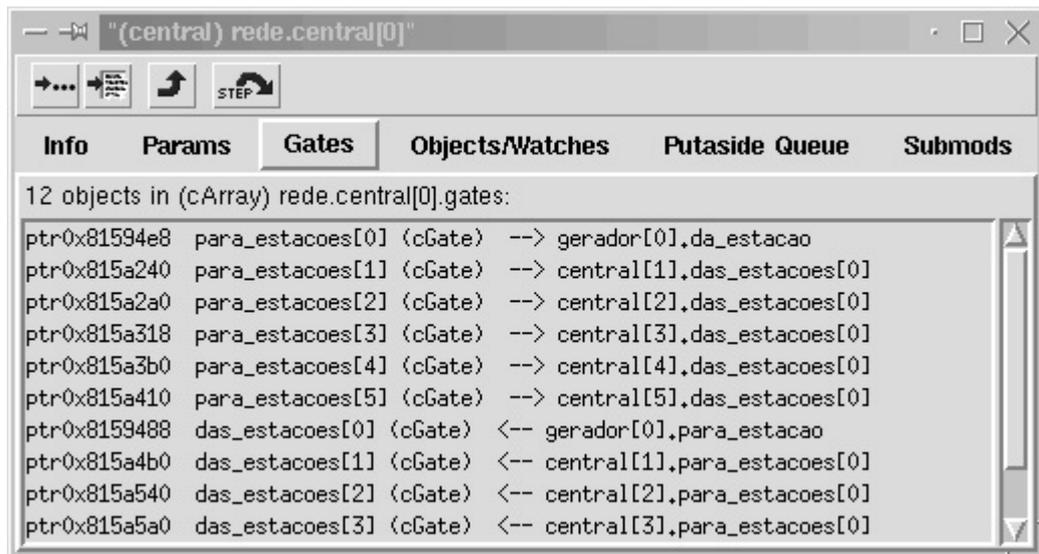


Fig. 5-26 - Observação de um módulo da simulação – portas de ligação entre módulos.

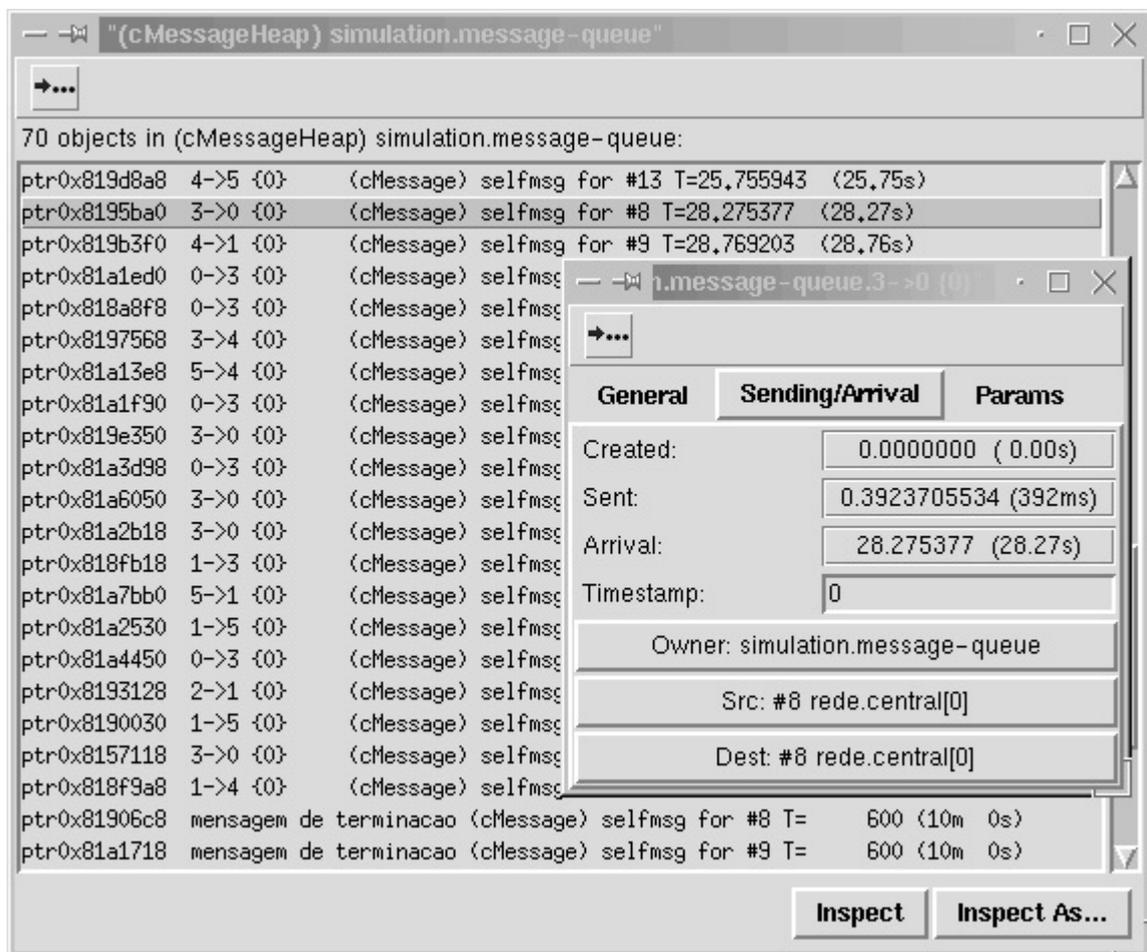


Fig. 5-27 - Execução da simulação – lista de mensagens, com pormenor de uma mensagem.

A fila de mensagens escalonadas (figura 5-27) permite-nos observar a todo o momento a totalidade das mensagens escalonadas pelo sistema. No sistema aqui descrito, as mensagens geradas podem ser mensagens correspondentes a chamadas telefónicas, ou mensagens de controlo (como as mensagens de

terminação). É possível observar em maior pormenor as mensagens trocadas pelo sistema, sendo possível observar as suas características genéricas, informação sobre as suas chegadas ou partidas (como é visível na figura 5-27) ou ainda os diversos parâmetros que tenham sido associados às mesmas (como se pode ver na figura 5-28).

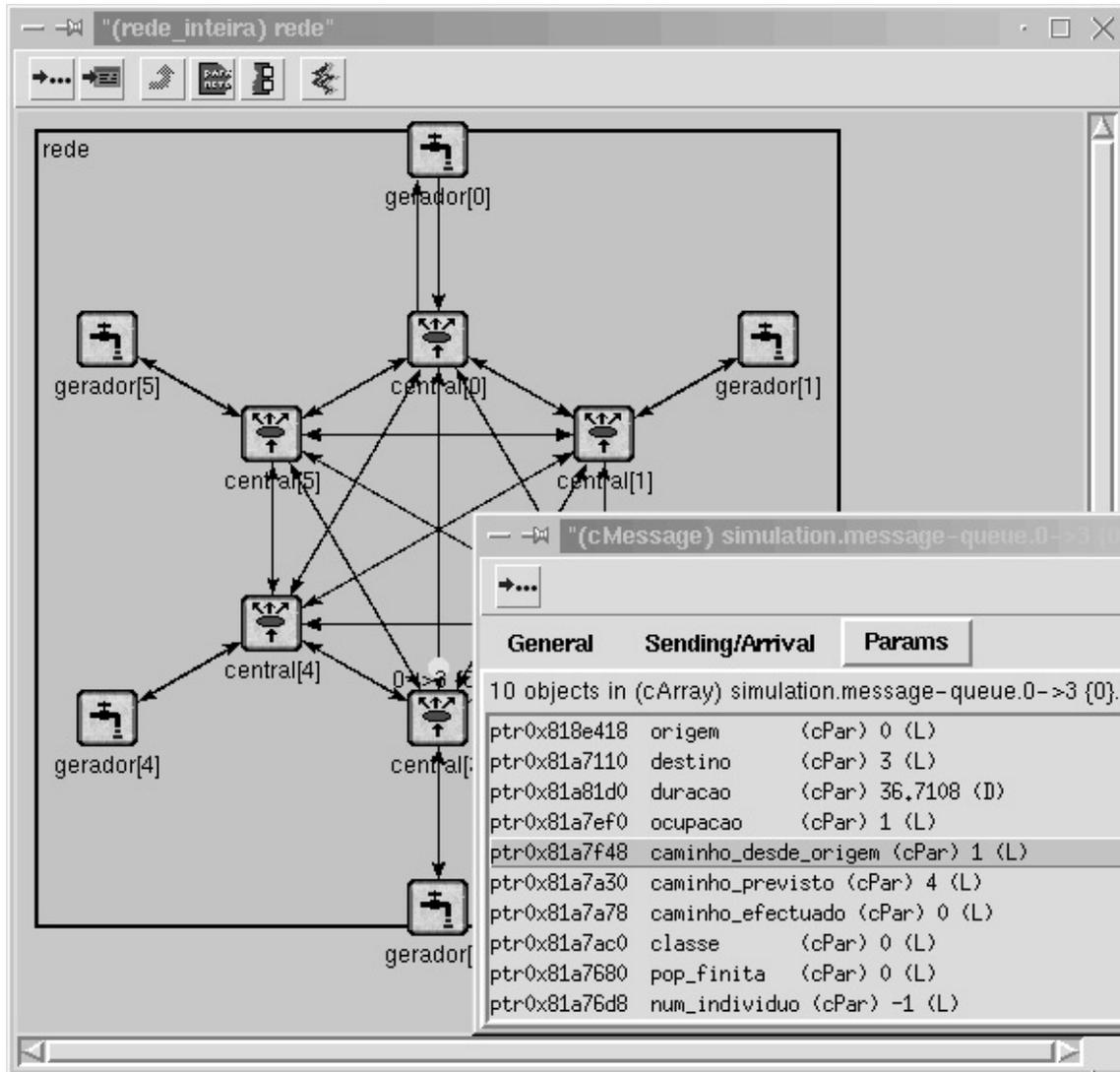


Fig. 5-28 - Execução de uma simulação – o envio de uma mensagem entre centrais, e os parâmetros que descrevem essa mensagem.

### 5.6.3 Apresentação dos resultados

As várias medidas obtidas foram armazenadas, pelo programa de simulação, em dois formatos, dando origem a dois ficheiros. Os valores armazenados num dos ficheiros de saída (ficheiro escalar) foram-no através da função `recordscalar` (função já referida e que faz parte da biblioteca de classes de simulação do OMNeT++) e por essa razão esse ficheiro é formado por várias linhas tendo um formato adequado à importação para uma base de dados ou folha de cálculo e posterior tratamento nela. No outro

ficheiro (ficheiro de tabelas) a informação é apresentada em forma de matriz o que torna a sua visualização simples.

No apêndice F podemos encontrar exemplos destes ficheiros, bem como uma exposição das suas estruturas.

## 5.6.4 Intervalos de confiança das estimativas

Para a análise de resultados de qualquer estudo simulacional, o cálculo da precisão dos mesmos é de grande importância. Para este efeito, foi decidido calcular intervalos de confiança, de acordo com um grau de precisão pré-definido, para cada uma das medidas analisadas do modelo.

Se usarmos o parâmetro  $\theta$  para designar uma das medidas do sistema, o resultado de um conjunto de simulações será uma estimativa  $\hat{\theta}$  de  $\theta$ . 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 ( $n$  corridas da simulação ou replicações) cada uma das quais usando uma sequência 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 ( $\sigma^2$ ) e média ( $\mu$ ) desconhecidas. A partir desta amostra vamos estimar o parâmetro  $\theta$  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 ([Law91]), podem-se aplicar os métodos clássicos para estimação pontual e intervalar do parâmetro  $\theta$ . A estimativa pontual de  $\theta$ ,  $\hat{\theta}$ , é igual à média aritmética da amostra,  $\bar{Y}$ , isto é:

$$\hat{\theta} = \bar{Y} = \frac{\sum_{j=1}^n Y_j}{n} \quad (5.3)$$

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  $\sigma^2$ ), define-se por ([Dudewicz88]):

$$s^2 = \frac{1}{n-1} \sum_{j=1}^n (Y_j - \bar{Y})^2 \quad (5-4)$$

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  $\varepsilon$  tal que:

$$\text{Prob}\left[|\bar{Y} - \mu| \leq \varepsilon\right] = 1 - \alpha \quad (5-5)$$

em que  $1 - \alpha$  é o grau de confiança.

Passando à variável normal reduzida  $Z$ ,  $N(0,1)$ :

$$Z = \frac{\bar{Y} - \mu}{\sigma/\sqrt{n}} \quad (5-6)$$

com

$$\varepsilon' = \frac{\varepsilon}{\sigma\sqrt{n}} \quad (5-7)$$

vem:

$$\text{Prob}\left[|Z| \leq \varepsilon'\right] = 1 - \alpha \quad (5-8)$$

Como a variância é desconhecida vamos ter que a substituir pela sua estimativa  $s^2$ . Temos que utilizar a variável aleatória  $T$  tal que:

$$T = \frac{\bar{Y} - \mu}{s/\sqrt{n}} \quad (5-9)$$

em vez da variável  $Z$ .

A variável aleatória  $T$  segue a distribuição t de Student com  $n - 1$  graus de liberdade [Kobayashi78].

Rescrevendo a equação 5-8, de acordo com as condições presentes:

$$\text{Prob}\left[\left|\frac{\bar{Y} - \mu}{s/\sqrt{n}}\right| \leq \varepsilon'\right] = 1 - \alpha \quad (5-10)$$

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, isto é,  $t_{f,\alpha}$  é o ponto definido por  $\text{Prob}[X \geq t_{f,\alpha}] = \alpha$ .

Observando agora a curva da distribuição de t de Student com  $n-1$  graus de liberdade conclui-se que  $\varepsilon'$  é o ponto  $t_{n-1;\alpha/2}$  dessa distribuição pois como:

$$\text{Prob}[|T| \leq \varepsilon'] = 1 - \alpha \quad (5.11)$$

vem que

$$\text{Prob}[|T| \geq \varepsilon'] = \alpha \quad (5.12)$$

e então

$$\text{Prob}[T \geq \varepsilon'] = \alpha/2 \quad (5.13)$$

Resolvendo a desigualdade 5.10 em ordem a  $\mu$  obtém-se:

$$\text{Prob}\left[\bar{Y} - t_{n-1;\alpha/2} \times \frac{s}{\sqrt{n}} \leq \mu \leq \bar{Y} + t_{n-1;\alpha/2} \times \frac{s}{\sqrt{n}}\right] = 1 - \alpha \quad (5.14)$$

O intervalo de confiança com grau de confiança  $1-\alpha$  é  $\left[\bar{Y} - t_{n-1;\alpha/2} \times \frac{s}{\sqrt{n}}, \bar{Y} + t_{n-1;\alpha/2} \times \frac{s}{\sqrt{n}}\right]$ .



# Capítulo 6

## Resultados da Simulação

### 6.1 Introdução

Neste capítulo são apresentados os resultados obtidos por simulação que parecem mais relevantes para os métodos de encaminhamento implementados. Estes métodos foram o DIRECTO, o FAR, o DAR, o DCR e o RTNR, sendo que destes apenas o DIRECTO, o DAR e o RTNR foram implementados de modo a suportar o transporte de várias classes de serviço na rede simultaneamente. O método de encaminhamento DIRECTO vai ser usado como termo de comparação e indicador da dimensão das vantagens conseguidas por cada um dos outros métodos de encaminhamento, em termos de desempenho da rede.

Designaremos o sistema por sistema multiclasse quando houver chamadas provenientes de vários tipos de fluxos de tráfego, associando a cada tipo de fluxo de tráfego uma classe de chamadas. Se no sistema houver apenas chamadas de uma classe será designado por sistema monoclasse. Dependendo do número de canais ocupados por uma chamada assim será designada por chamada monocanal ou multicanal. Se a chamada ocupar apenas um canal será designada por chamada monocanal, se ocupar mais será designada por chamada multicanal.

Este capítulo inicia-se com uma descrição das características comuns a todas as simulações, e está dividido em três partes principais, correspondendo cada uma a um caso particular do sistema em estudo, que foram definidos no capítulo 4. Na primeira parte é oferecido à rede apenas uma classe de chamadas, trata-se de um sistema monoclasse com apenas chamadas monocanal (*sistema 1*). Na segunda parte os recursos da rede são partilhados por duas classes de chamadas, o tipo de tráfego é de Poisson para ambas as classes e as chamadas dos dois tipos de fluxos requerem um número de circuitos diferente para o seu transporte. Temos neste caso um sistema multiclasse com chamadas monocanal e multicanal (*sistema 2*). Na terceira parte vamos considerar um sistema com integração de 4 serviços. O tipo de tráfego é de

Poisson para duas das classes de chamadas e para as duas restantes é de população finita (tráfego tipo Engset). A necessidade de recursos é dependente da classe de chamada. Consideraremos também aqui um sistema multiclasse com chamadas monocanal e multicanal (*sistema 3*).

## 6.2 Detalhes do estudo e características comuns a todas as simulações

A análise dos desempenhos dos vários métodos de encaminhamento foi feita numa rede com seis centrais, completamente emalhada. A topologia da rede utilizada foi a apresentada em [Mitra91a], como referido no capítulo 4. Para cada par de centrais (origem, destino) especificámos a capacidade do feixe e a intensidade de tráfego oferecido, no caso de tráfego de Poisson. No caso de tráfego de população finita especificámos o número de fontes e o tráfego de chamadas por fonte livre para cada uma das centrais e que proporções (%) do tráfego gerado em cada central se destina a cada uma das centrais adjacentes.

A regra (muito simples) de paragem do programa de simulação é a seguinte: cada replicação corre durante um determinado tempo de simulação (tempo simulado) pré-fixado. Desse, um tempo simulado inicial (tempo de transição) é desprezado, só passado esse tempo é que começa a ser feita a recolha de dados, para reduzir o efeito dos transitórios iniciais nas medidas de desempenho calculadas. Nas corridas efectuadas escolhemos um tempo simulado equivalente a 48 horas e um tempo de transição correspondente a 24 horas (verificou-se que passado este tempo, os resultados não apresentavam flutuações significativas, como se constata pelos resultados apresentados no gráfico 6.9). Estes valores correspondem aproximadamente à geração de desde cerca de um milhão e trezentas mil chamadas para a carga base (e para o sistema 1) até cerca de três milhões e quatrocentas mil chamadas para o factor de carga 2 (para o sistema 3). Em cada um destes casos, os valores apresentados referem-se a medidas durante o período útil, após o período de transição (logo, o número efectivo de chamadas geradas terá sido em cada caso aproximadamente o dobro). Para cada um dos casos foram executadas 5 replicações independentes 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. Os métodos de encaminhamento foram comparados sob várias condições de carga, tendo sido estudadas 11 intensidades de tráfego oferecido. Estas correspondem à intensidade de tráfego exógeno original e a acréscimos de 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90% e 100% deste. A intensidade de chegadas foi obtida a partir da matriz de tráfego exógeno considerando que o tempo de serviço médio é um minuto, no caso de tráfego de Poisson. No caso de tráfego de população finita a intensidade de chegada de chamadas por fonte livre foi obtida a partir do tráfego de chamadas por fonte livre para cada uma das centrais considerando também aqui a unidade para o tempo de serviço médio das chamadas. Tal

conduziu a que os valores obtidos para a intensidade de chegadas, no tráfego de Poisson, e para a intensidade de chegada de chamadas por fonte livre, no tráfego de população finita, sejam iguais respectivamente ao tráfego exógeno e ao tráfego de chamadas por fonte livre em Erlangs.

Para cada um dos estudos foram executadas 5 corridas com exactamente os mesmos parâmetros diferindo apenas nos números aleatórios usados. Para cada uma destas replicações foram escolhidas previamente 4096 sementes do gerador de números pseudo-aleatórios. Estas sementes encontram-se separadas entre si no ciclo gerador de números aleatórios de cerca de quinhentas mil unidades. Nas replicações com o mesmo número de corrida foi sempre utilizado o mesmo conjunto de sementes.

O programa foi executado num microcomputador com dois processadores Pentium III a 500 MHz com 512 MB de memória principal. O sistema operativo utilizado foi o Linux (RedHat). O tempo gasto por cada corrida varia de 1 a 40 minutos, sendo esta diferença resultante principalmente de dois factores, a grande diferença de complexidade na implementação dos vários métodos de encaminhamento e a carga de tráfego que é gerado.

Todos os ficheiros dos resultados obtidos nas várias corridas executadas foram consolidados em ficheiros de texto por um programa, desenvolvido em Perl (implementação feita pelo Eng.º Paulo Melo). A partir desses ficheiros de texto foram construídas bases de dados em Microsoft Access, que ocupam no total cerca de 150 MB. A análise dos resultados foi feita usando o Microsoft Excel tendo este como entrada os dados das base de dados referida anteriormente.

## 6.2.1 Alguns valores comuns a todas as simulações

Em todas as experiências realizadas com este sistema os valores para os parâmetros específicos de cada método, introduzidos no ficheiro de configuração (ficheiro *ini*), foram os apresentados na tabela 6-1.

Parâmetros	Valores
Intervalo de extrapolação ( $\tau$ )	1s
Intervalo de actualização ( $\Delta$ )	1s
Peso de média móvel ( $\theta$ )	0,9
Intervalo de actualização	3m
Factor1	1,1
Factor2	0,05
Tipo de reserva de circuitos	0, 1 ou 2

} RTNR e DAR

} DCR

Tab. 6-1 – Valores para os parâmetros específicos de cada método.

Os três primeiros parâmetros da tabela são específicos do método de encaminhamento DCR, e são usados para determinar os valores das expressões, relativas a este método, apresentadas no capítulo 3 (secção 3.5.1.4). Os restantes parâmetros da tabela são usados nos cálculos das expressões relativas ao

método de encaminhamento RTNR, apresentadas no capítulo 3 secção 3.5.1.3). No entanto, como vai ser usada reserva de circuitos dinâmica determinada pela expressão 5.1 do capítulo 5 (simplificação da reserva de circuitos original do RTNR, secção 5.5.4.1), tanto no método de encaminhamento DAR como no DCR, então estes últimos parâmetros também são necessários nestes métodos.

O parâmetro intervalo de actualização é utilizado pelo método de encaminhamento DAR e DCR como sendo a frequência com que devem ser actualizados os valores das reservas nos feixes (caso seja usado o tipo de reserva de circuitos 1, veja-se tabela 6.2).

Factor1 e Factor2 correspondem aos factores a serem usados nas seguintes expressões:

$$VTtraf_k = Factor1 \times TL_k \tag{6.1}$$

o que corresponde ao caso geral da expressão 3.3 apresentada no capítulo 3 e

$$R_k (VTs) = Factor2 \times n \times VTtraf_k \tag{6.2}$$

correspondendo esta expressão ao caso geral da expressão usada na tabela 3.1 do capítulo 3 e onde  $n$  é o nível de reserva.

A convenção para os vários valores possíveis para o parâmetro “tipo de reserva de circuitos” é indicada na tabela 6.2. Para o método de encaminhamento DAR bem como para o RTNR apenas podem ser usados os valores 0 ou 1, para o método de encaminhamento DCR podem ser usados três valores (0, 1 ou 2).

Tipo de reserva de circuitos	DAR (dar_tipo_reserva_din)	DCR (dcr_tipo_reserva_din)	RTNR (rtnr_tipo_reserva_din)
0	Reserva de circuitos fixa, original do DAR.	Reserva de circuitos fixa, original do DCR.	Reserva de circuitos dinâmica, original do RTNR.
1	Reserva de circuitos estilo RTNR – reserva de circuitos dinâmica determinada pela expressão 5.1 do capítulo 5 (simplificação da reserva de circuitos original do RTNR).		
2		Reserva de circuitos dinâmica do DCR – determinada pela expressão 5.2 do capítulo 5.	

Tab. 6.2 – Significado dos valores do parâmetro tipo de reserva de circuito, para cada um dos métodos.

A escolha do tipo de reserva de circuitos para cada um dos métodos é feita pela atribuição dos valores pretendidos aos parâmetros respectivos (*dar\_tipo\_reserva\_din*, *dcr\_tipo\_reserva\_din* e *rtnr\_tipo\_reserva\_din*), no ficheiro *cfg* (ver apêndice C). Na tabela anterior a reserva de circuitos referida diz respeito apenas à reserva de circuitos para protecção do tráfego de primeiro encaminhamento (directo). Lembremo-nos que dos três métodos de encaminhamento, como definidos originalmente, apenas o RTNR é que utilizava, além da reserva de circuitos para protecção do tráfego de primeiro encaminhamento, um mecanismo de protecção dos serviços também designado por reserva de circuitos. No RTNR, tal como implementado, é utilizado sempre esse mecanismo de protecção dos serviços, tanto para o valor 0 como para o valor 1 do parâmetro. Este mesmo mecanismo de protecção dos serviços também é utilizado pelo DAR implementado, quando este utilizar a reserva de circuitos estilo RTNR para protecção do tráfego directo.

Outros parâmetros, introduzidos no ficheiro de configuração (ficheiro *ini*), foram: o nome do ficheiro de topologia, o factor de sobrecarga, o instante em que se inicia a recolha de dados e o instante em que termina a simulação.

Foram ainda escolhidos os valores que se encontram na tabela 6-3 para cada um dos parâmetros aí referidos.

Parâmetros	Classe	Valores
Duração das chamadas	Todas	1 minuto
Número de circuitos necessários para transportar cada chamada	0	1
	1	2
	2	6
	3	24
Controlo protector do tráfego de primeiro encaminhamento - Reserva de circuitos fixa	0	4
	1	6
	2	12
	3	26

Tab. 6-3 – Valores para os parâmetros comuns a todos o métodos.

Como é obvio para cada simulação foram apenas escolhidos os valores correspondentes às classes que são usadas nessa simulação particular. O valor para a reserva de circuitos indicado na tabela 6-3 foi utilizado para todos os feixes em cada uma das classes de chamadas (apenas quando foi usada a reserva fixa).

Os valores da tabela 6-4 são utilizados, no ficheiro de topologia, para indicar se o tráfego de uma determinada classe especificada é tráfego de Poisson ou de população finita.

Parâmetros	Valores	Significado do valor
Tipo de fluxo de tráfego	0	Tráfego de Poisson
	1	Tráfego de população finita

Tab. 6-4 – Outros valores para os parâmetros comuns a todos os métodos.

Os valores das tabelas 6-3 e 6-4 foram especificados no ficheiro de topologia (ficheiro *txt*).

Em todas as experiências descritas à frente os valores dos parâmetros utilizados são os indicados nesta secção a menos que sejam indicados outros. Um parâmetro que não foi referido, e cujo valor é especificado no ficheiro de topologia, é a largura de banda mínima garantida para cada classe. Não vai ser referido aqui nenhum valor para esse parâmetro pois o seu valor é dependente do caso particular do sistema a simular, tendo sido obtido como (aproximadamente) proporcional às necessidades estimadas de circuitos para cada classe (de acordo com o tráfego de chamadas previsto para cada classe, as necessidades de cada classe e o número de circuitos existente em cada feixe). Por esta razão, é necessário especificar o seu valor para cada um dos pares de centrais.

## 6.3 Experiência computacional com o Sistema 1

Nesta secção vão ser apresentados alguns dos resultados obtidos com um dos casos particulares do sistema em estudo, definido no capítulo 4, o designado por sistema 1.

No apêndice C, na secção C.3.1, encontra-se o ficheiro de topologia (ficheiro *txt*) que foi usado nesta experiência. Nele são especificados entre outros os valores para o tráfego oferecido ponto a ponto para todos os pares de centrais e os números de circuitos entre todos os pares de centrais.

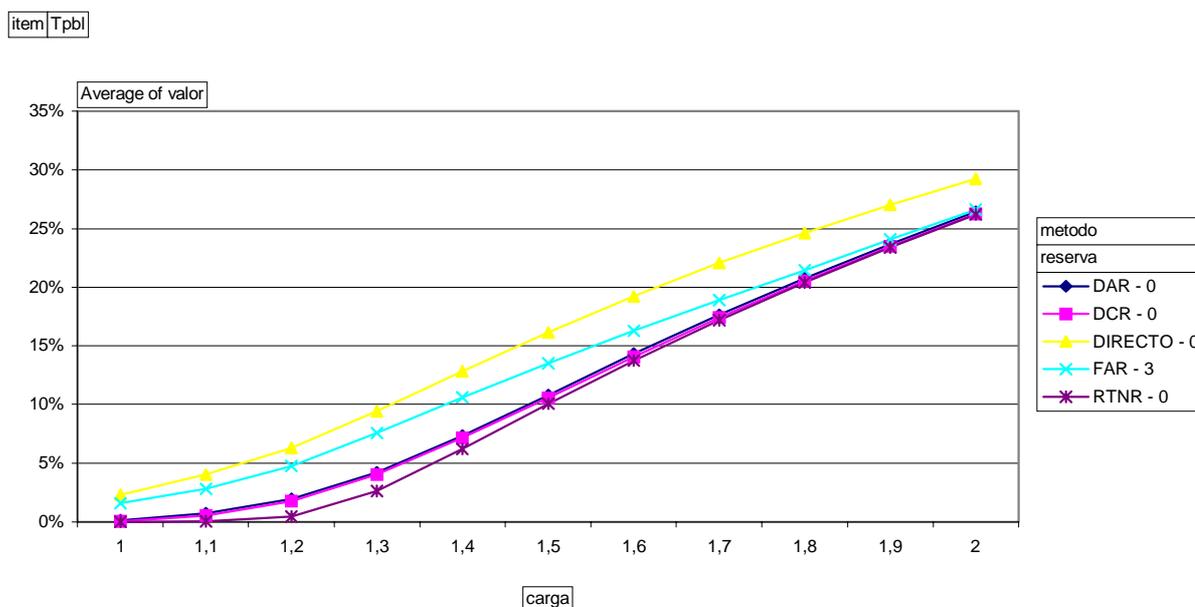
### 6.3.1 Análise da probabilidade de bloqueio global

Vamos começar por analisar os valores das probabilidades de bloqueio global que foram estimados pelo programa de simulação (no final da simulação).

Os resultados apresentados em cada um dos gráficos, neste capítulo, correspondem sempre ao valor do item particular escolhido para ser analisado, sendo o significado de cada item apresentado nas tabelas F-1 e F-2 do apêndice F, secção F.1.1. Na legenda de cada gráfico, apresentada sempre à direita do gráfico, podem aparecer vários campos. Por baixo dos campos aparece a legenda propriamente dita, sendo esta legenda constituída por tantas linhas quantas as combinações possíveis dos valores que foram

seleccionados para cada um dos campos. Os valores dos campos aparecem em cada uma dessas linhas pela ordem em que foram especificados os campos.

No gráfico 6-1 podem ser visualizadas as probabilidades de bloqueio global para todos os métodos de encaminhamentos implementados para vários factores de sobrecarga. Existe uma excepção neste gráfico em relação ao valor do campo reserva, pois o valor 3 no método de encaminhamento FAR não indica o tipo de reserva, mas sim o número máximo de caminhos alternativos entre cada par de centrais.



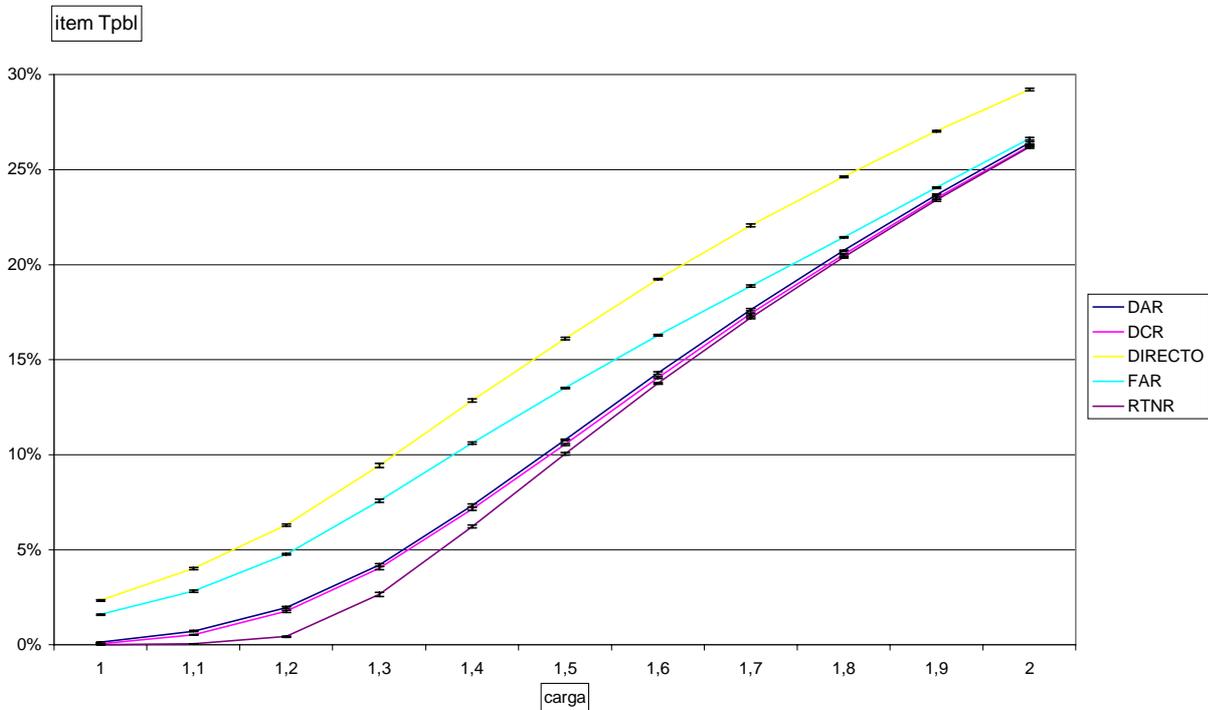
Gráf. 6-1 – Probabilidade de bloqueio global para todos os métodos de encaminhamentos implementados (sistema 1).

Em relação ao gráfico 6-1 podemos verificar que os métodos de encaminhamento dinâmicos tiveram um desempenho superior e distinto entre eles, no entanto para valores do factor de sobrecarga elevados (>1,6) o desempenho entre eles torna-se muito semelhante.

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. Para tal, fez-se uma análise estatística, utilizando intervalos de confiança para cada valor apresentado, de acordo com o descrito no final do capítulo anterior.

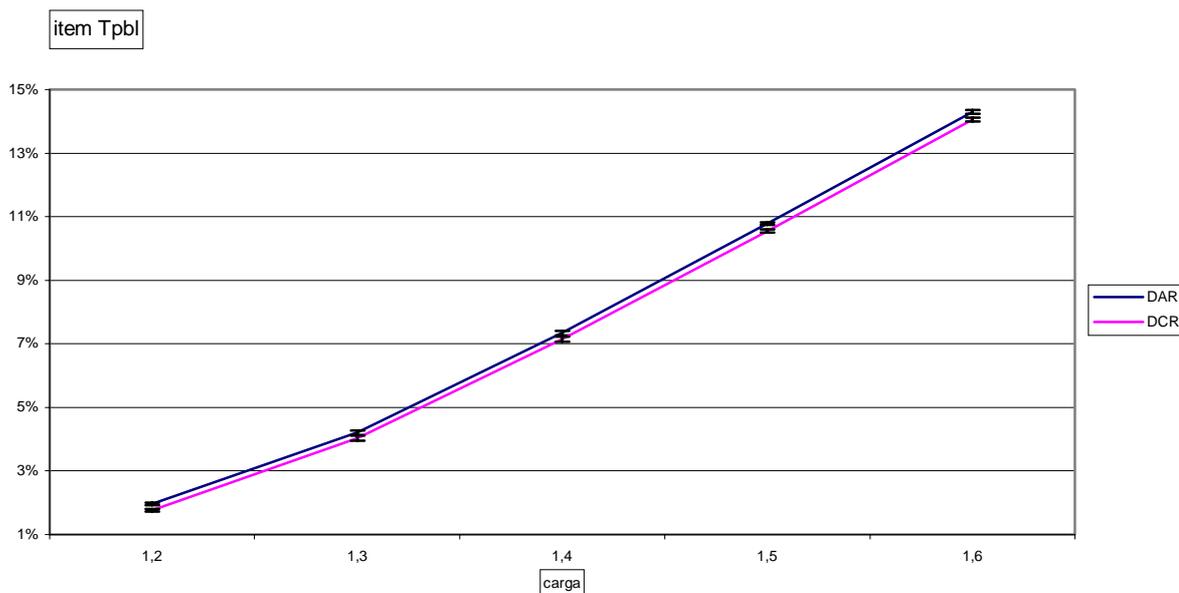
No restante deste capítulo, e de acordo com a fórmula 5-14 do capítulo 5, foi usado um grau de confiança de 95% (correspondente a um alfa de 5%) para a determinação dos intervalos de confiança. Os limites dos intervalos de confiança e a sua semi-largura relativa para cada valor apresentado, podem ser consultados no apêndice G.

No gráfico 6-2 podem ser também visualizadas as probabilidades de bloqueio global para todos os métodos de encaminhamentos implementados para vários factores de sobrecarga. No entanto, para cada valor é apresentado graficamente o intervalo de confiança correspondente a esse ponto, através da apresentação de uma barra vertical cuja dimensão corresponde a esse intervalo, e que une o limiar inferior e o superior deste (cada um dos quais correspondendo a uma linha horizontal).



Gráf. 6-2 – Probabilidade de bloqueio global para todos os métodos de encaminhamentos implementados (sistema 1) indicando intervalos de confiança.

No gráfico 6-2 a separação entre os métodos DAR e DCR é demasiado reduzida para serem perceptíveis os intervalos de confiança correspondentes a estes métodos. Por esta razão foram representados estes métodos no gráfico 6-3 de forma a tornar essa visualização possível.



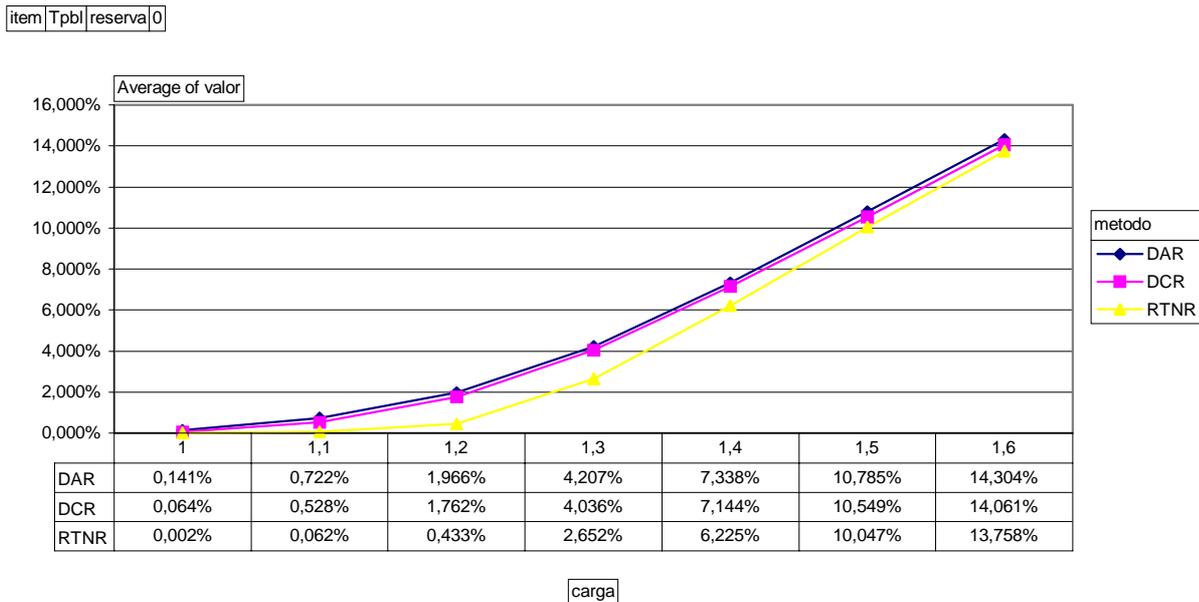
Gráf. 6-3 – Probabilidade de bloqueio global para os métodos de encaminhamentos DAR e DCR (sistema 1), indicando intervalos de confiança.

É possível agora observar no gráfico 6-3 que para cargas de 1,2 a 1,6 o método DCR é objectivamente melhor que o DAR de acordo com o grau de confiança utilizado. Note-se que embora aparentemente o gráfico indique uma sobreposição dos intervalos de confiança para cargas de 1,3 e 1,4, de acordo com a tabela G-1 do apêndice G não chega a existir esta sobreposição. De facto, como é visível nessa tabela, o bloqueio global para carga 1,3 no método DAR varia entre 4,131% e 4,284% (valor médio de 4,207%) e no método DCR entre 3,953% e 4,119% (valor médio de 4,036%).

Em qualquer caso, estes e outros resultados mostram que os valores dos bloqueios globais foram calculados com intervalos de confiança muito pequenos, o que evidencia que o dimensionamento das replicações foi plenamente ajustado aos objectivos deste estudo.

Embora a utilização de gráficos para análise dos intervalos de confiança pareça intuitivamente mais adequada, conforme se verifica nos gráficos 6-2 e 6-3, os valores objectivamente muito reduzidos destes intervalos e a sua proximidade fazem com que não se consiga visualizar as diferenças que eventualmente possam existir entre intervalos muito pequenos e próximos entre si. Por esta razão, na totalidade dos gráficos apresentados seguidamente neste capítulo, em vez da representação gráfica dos intervalos de confiança, optou-se pela sua apresentação apenas sob a forma tabular no apêndice G.

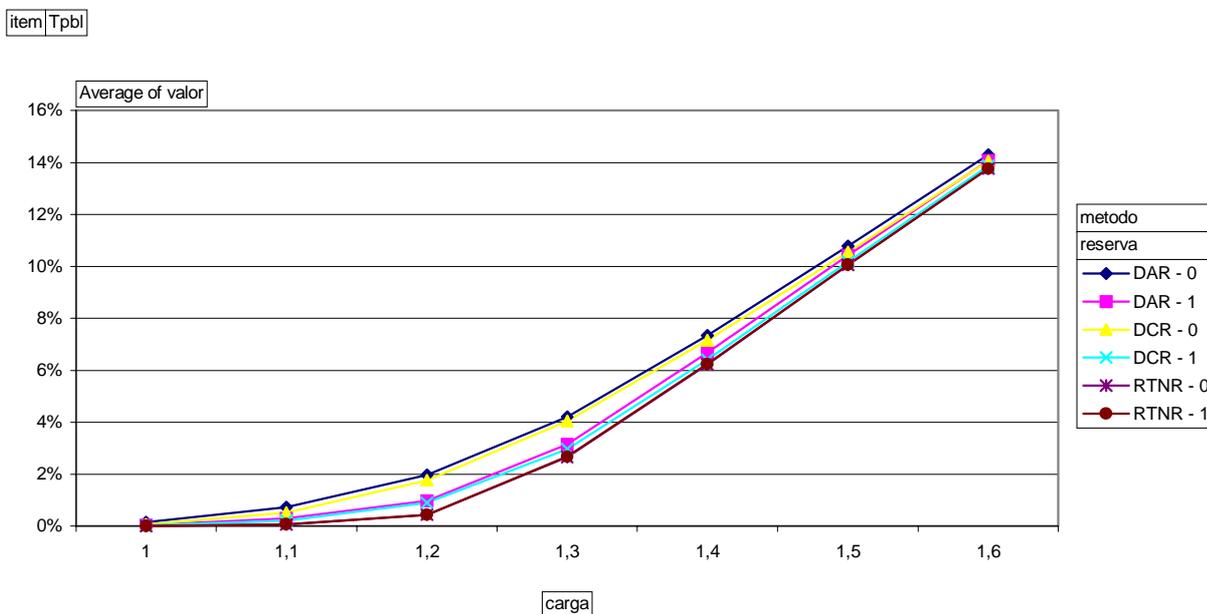
No gráfico 6-4 podem também ser visualizadas as probabilidades de bloqueio global para vários factores de sobrecarga mas apenas para os métodos de encaminhamentos dinâmicos. Este gráfico contém uma tabela com os valores das probabilidades de bloqueio.



Gráf. 6-4 – Probabilidade de bloqueio global nos métodos de encaminhamento dinâmicos (sistema 1).

Pela observação do gráfico 6-4 podemos afirmar que o método que teve um desempenho melhor (para cargas <1,6) foi o RTNR. O comportamento do DCR e do DAR foi muito semelhante, tendo sido o DCR muito ligeiramente superior, como indicado anteriormente.

No gráfico 6-5 podem também ser observados os valores das probabilidades de bloqueio global para os métodos de encaminhamentos dinâmicos mas agora para dois tipos de reserva de circuitos.



Gráf. 6-5 – Probabilidade de bloqueio global nos métodos de encaminhamento dinâmicos para dois tipos de reserva (sistema 1).

Pelos resultados apresentados no gráfico 6-5 podemos afirmar que quando foi utilizado um mecanismo de reserva de circuitos dinâmico, no método DAR e DCR, simplificação do mecanismo utilizado no RTNR, o desempenho de ambos melhorou significativamente sendo no entanto ainda inferior ao conseguido pelo RTNR. Verificamos ainda que ao ser utilizado no método de encaminhamento RTNR a simplificação no mecanismo de reserva de circuitos original proposta no capítulo 5 (equação 5-1) o seu desempenho não sofreu praticamente nenhuma alteração. O DCR continua a ser geralmente superior ao DAR neste caso, embora exista uma muitíssimo pequena sobreposição dos intervalos de confiança para a carga 1,3. Esta interpretação é válida à luz dos intervalos de confiança apresentados na tabela G-2 do apêndice G.

Os valores das probabilidades de bloqueio global apresentados nas tabelas 6-5 nas colunas centrais e na coluna mais à direita correspondem respectivamente aos resultados obtidos por nós na simulação (incluindo um intervalo de confiança correspondente a um grau de confiança de 95%) e aos obtidos por simulação em [Mitra91a].

Factor de carga	DAR			DAR (Mitra)
	Lim. Inf.	Média	Lim. Sup.	
1,0	0,13	0,14	0,15	0,14
1,5	10,74	10,78	10,83	10,48
2,0	26,35	26,41	26,48	26,08

Tab. 6-5 – Comparação da probabilidade de bloqueio global (%) obtidas por simulação com o DAR.

Na tabela 6-6 os valores das probabilidades de bloqueio global correspondem aos resultados obtidos analiticamente e por simulação por Ash [Ash93] e aos resultados obtidos por nós na simulação (incluindo um intervalo de confiança para um grau de confiança de 95%), respectivamente da coluna da direita para a esquerda.

Factor de carga	RTNR			RTNR	RTNR
	Lim. Inf.	Média	Lim. Sup.	(Ash - simulação)	(Ash - analiticamente)
1,0	0,0	0,0	0,0	0,0	0,0
1,1	0,1	0,1	0,1	0,1	0,0
1,2	0,4	0,4	0,5	0,5	0,4
1,3	2,5	2,6	2,8	2,5	2,5
1,4	6,1	6,2	6,3	5,8	5,9

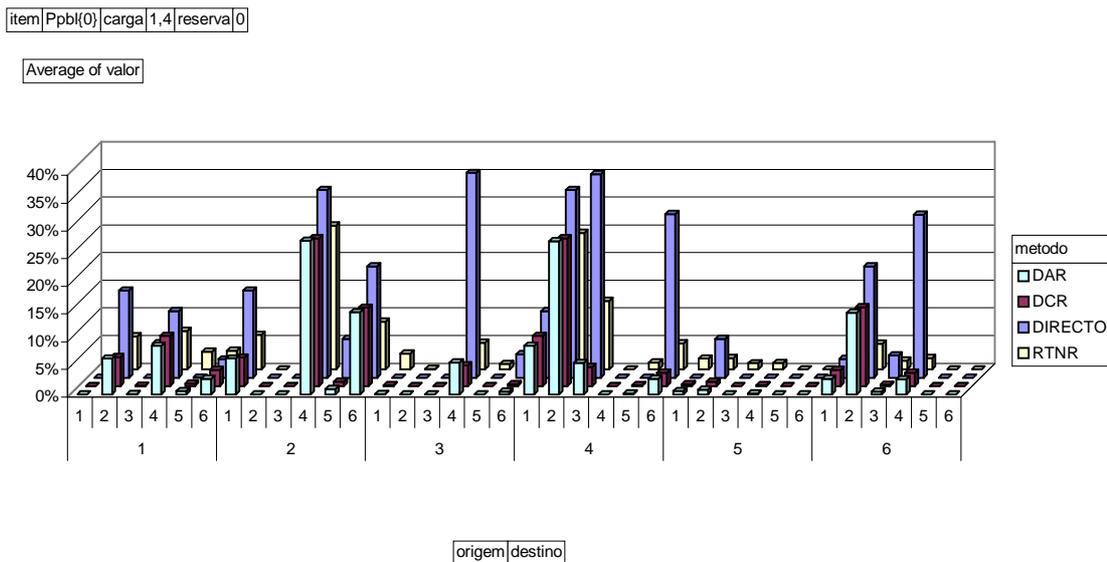
Tab. 6-6 – Comparação da probabilidade de bloqueio global (%) obtidas por simulação com o RTNR.

Os valores das probabilidades de bloqueio global apresentados nas tabelas 6-5 e 6-6 mostram a semelhança entre os resultados obtidos por nós na simulação e os resultados obtidos por simulação por

Mitra [Mitra91a] e os resultados obtidos por simulação e analiticamente por Ash [Ash93], respectivamente. Deve ser notado que embora os valores sejam do mesmo tipo de ordem de grandeza, não se pode ignorar o facto de que para a carga mais elevada no caso de RTNR, e para as cargas mais elevadas no caso do DAR, os intervalos de confiança (para alfa=5%) dos resultados obtidos não estão em concordância com os valores previstos por Ash e Mitra respectivamente.

### 6.3.2 Análise da probabilidade de bloqueio ponto a ponto

O gráfico 6-6 ilustra os bloqueios ponto a ponto com um factor de sobrecarga de 1,4 , para o método de encaminhamento DIRECTO e para os métodos de encaminhamento dinâmicos (DAR, DCR e RTNR). Foi utilizado este factor de carga, uma vez que para carga nominal, os valores dos bloqueios ponto-a-ponto são muito reduzidos, o que levaria a valores muito elevados de dimensão relativa dos intervalos de confiança (que já de si são em geral maiores que os das medidas de valores globais, uma vez que os bloqueios ponto a ponto correspondem a acontecimentos pouco (ou muito pouco) frequentes para certos pares origem-destino).

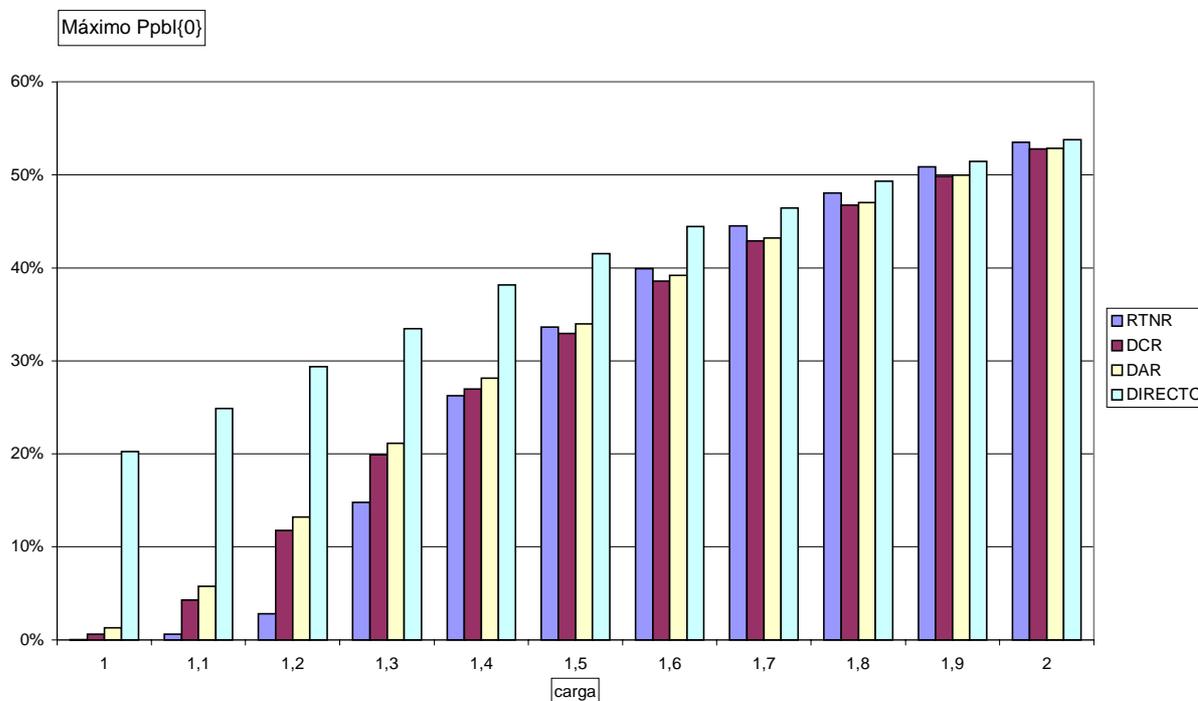


Gráf. 6-6 – Probabilidade de bloqueio ponto a ponto nos métodos de encaminhamento dinâmicos e no DIRECTO (sistema 1).

Os resultados apresentados no gráfico 6-6 mostram que com o método de encaminhamento RTNR os valores dos bloqueios ponto a ponto entre os vários pares de centrais são muito mais uniformes do que com os outros métodos de encaminhamento dinâmicos. Os outros métodos de encaminhamento dinâmicos não conseguiram distribuir tão bem o tráfego por toda a rede de modo a que as assimetrias entre os bloqueios ponto a ponto, entre os vários pares de centrais, sejam pequenas. Isto parece indicar que os mecanismos especializados de reserva de circuitos utilizados neste método são mais eficientes

que os restantes. Estes resultados estão de acordo com o apresentado nas tabelas G-3 e G-4 do apêndice G.

O gráfico seguinte apresenta os valores máximos de bloqueio ponto a ponto para diferentes métodos de encaminhamento e para várias cargas.



Gráf. 6-7 – Probabilidade de bloqueio ponto a ponto máxima nos métodos de encaminhamento dinâmicos e no DIRECTO, para cada um dos vários factores de sobrecarga (sistema 1).

Para os métodos que usam encaminhamento dinâmico, o valor máximo da probabilidade de bloqueio ponto a ponto ocorre no método RTNR para cargas elevadas (note-se que pelo contrário, para cargas reduzidas, inferiores a 1,4, já seria o RTNR o método mais vantajoso). Se se pretender que todos os valores dos bloqueios ponto a ponto estejam abaixo de um determinado valor pode acontecer que um método com bloqueio global superior seja preferido a um método com bloqueio global inferior no caso deste possuir alguns bloqueios ponto a ponto elevados. Neste caso, no entanto, podemos verificar que para cargas reduzidas o método que possui bloqueio global menor (ver gráfico 6-4) é também aquele que apresenta regularmente os menores bloqueios ponto a ponto, não sobrecarregando especialmente nenhum feixe. A tabela G-5 do apêndice G confirma a validade destas afirmações.

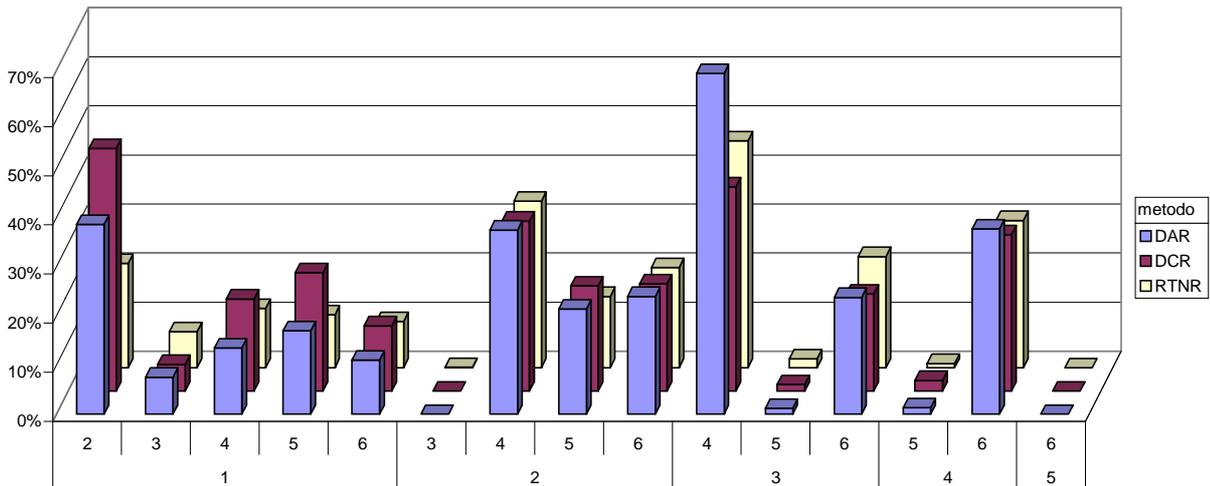
### 6.3.3 Análise da probabilidade de bloqueio nos feixes

O gráfico 6-8 ilustra as percentagens de bloqueio com um factor de sobrecarga de 1,4 para todos os feixes da rede, para os métodos de encaminhamento dinâmicos. Esta informação permite-nos saber quais

os feixes mais solicitados por cada método, mas o valor absoluto da percentagem de bloqueio em determinado feixe não é em princípio facilmente relacionável com a percentagem de bloqueio efectiva ponto a ponto ou global, pelo que apenas faz sentido comparar valores relativos tendo em vista apreciar o grau de utilização relativa dos feixes.

item | Rpb | reserva | 0 | carga | 1,4

Average of valor



origem | destino

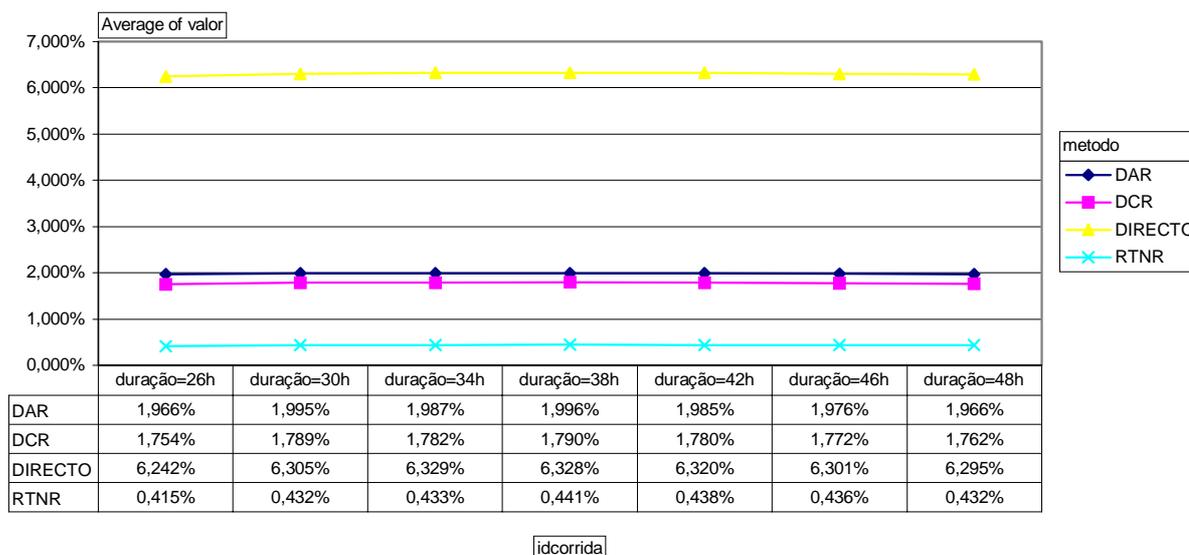
Gráf. 6-8 – Probabilidade de bloqueio nos feixes nos métodos de encaminhamento dinâmicos (sistema 1).

Os resultados apresentados no gráfico 6-8 mostram que os feixes mais congestionados são os feixes entre as centrais 1-2, 2-4, 3-4 e 4-6, com a excepção que no método RTNR o feixe entre as centrais 3-6 tem uma congestão superior ao feixe entre as centrais 1-2. Estas conclusões são suportadas pela tabela G-6 do apêndice G.

### 6.3.4 Análise ao comportamento da simulação

No gráfico 6-9 são apresentadas as probabilidades de bloqueio global nos métodos de encaminhamento dinâmicos para vários instantes ao longo do decorrer da simulação, para um factor de sobrecarga de 1,2. Nesse gráfico é também apresentada uma tabela com os valores dos bloqueios globais nos vários instantes de uma corrida, para todos os métodos, para ajudar a clarificar o gráfico.

item | Tpb | carga | 1,2



Gráf. 6-9 – Probabilidade de bloqueio global nos métodos de encaminhamento dinâmicos versus tempo simulado para um factor de sobrecarga de 1,2 (sistema 1).

Os resultados da simulação, apresentados no gráfico 6-9, indicam que o bloqueio global da rede apresenta variações muito ligeiras após o estado estacionário ser alcançado (depois de 24 horas de tempo simulado), não aparentando possuir um comportamento oscilatório.

A tabela G-7 do apêndice G indica que cada um dos valores apresentados encontra-se geralmente em intervalos de confiança de dimensão reduzida. A tabela G-8 do mesmo apêndice (criada usando todas as observações para o mesmo método, mesmo que tomadas em tempos diferentes) indica para cada método intervalos de variação extremamente reduzidos, confirmando assim a hipótese de que o tempo simulado escolhido não afecta de forma significativa os resultados obtidos.

## 6.4 Experiência computacional com o Sistema 2

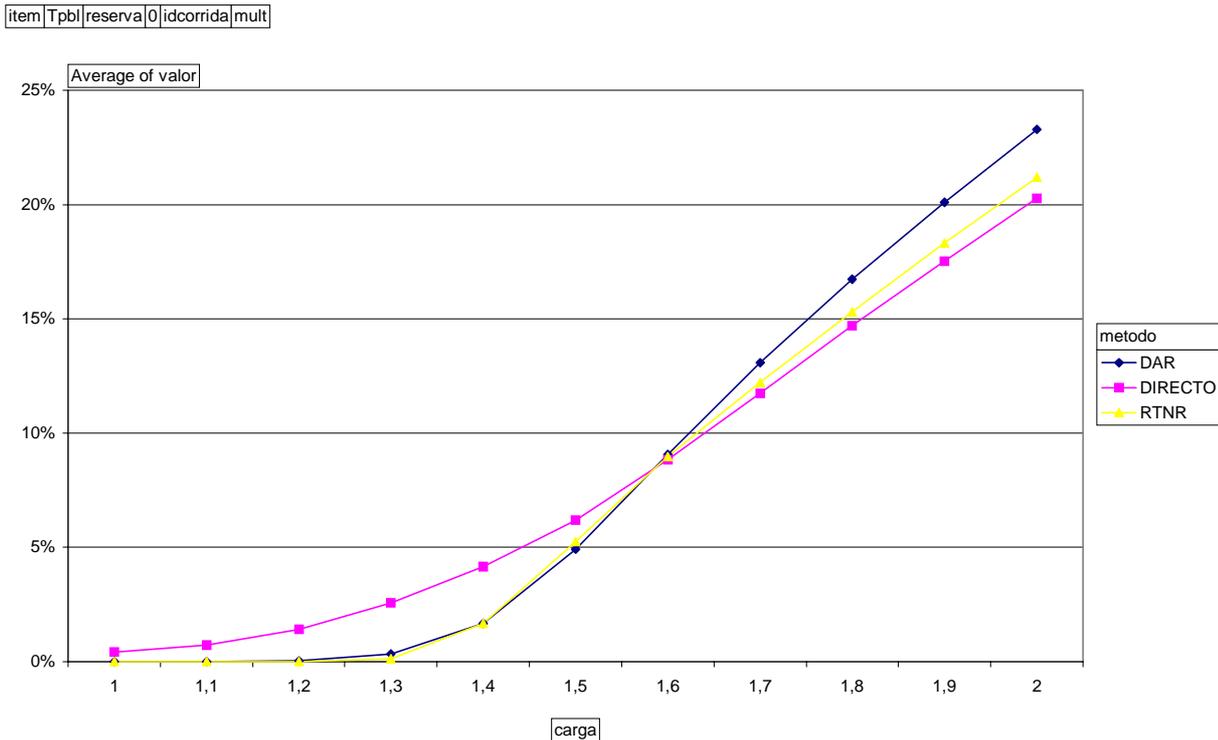
Nesta secção vão ser apresentados alguns dos resultados obtidos por simulação com o sistema 2, definido no capítulo 4.

No apêndice C, na secção C.3.2, encontra-se o ficheiro de topologia (ficheiro *txt*) que foi usado nesta experiência. Nele são especificados entre outros os valores para o tráfego oferecido ponto a ponto para todos os pares de centrais para cada uma das classes e os números de circuitos entre todos os pares de centrais.

## 6.4.1 Análise da probabilidade de bloqueio global na rede

Vamos começar por analisar os valores das probabilidades de bloqueio global na rede que foram computados pelo programa de simulação (no final da simulação).

O gráfico 6-10 ilustra esses valores para vários factores de sobrecarga, para os métodos de encaminhamento que foram implementados de modo a suportar a integração de serviços (DAR, RTNR e DIRECTO).



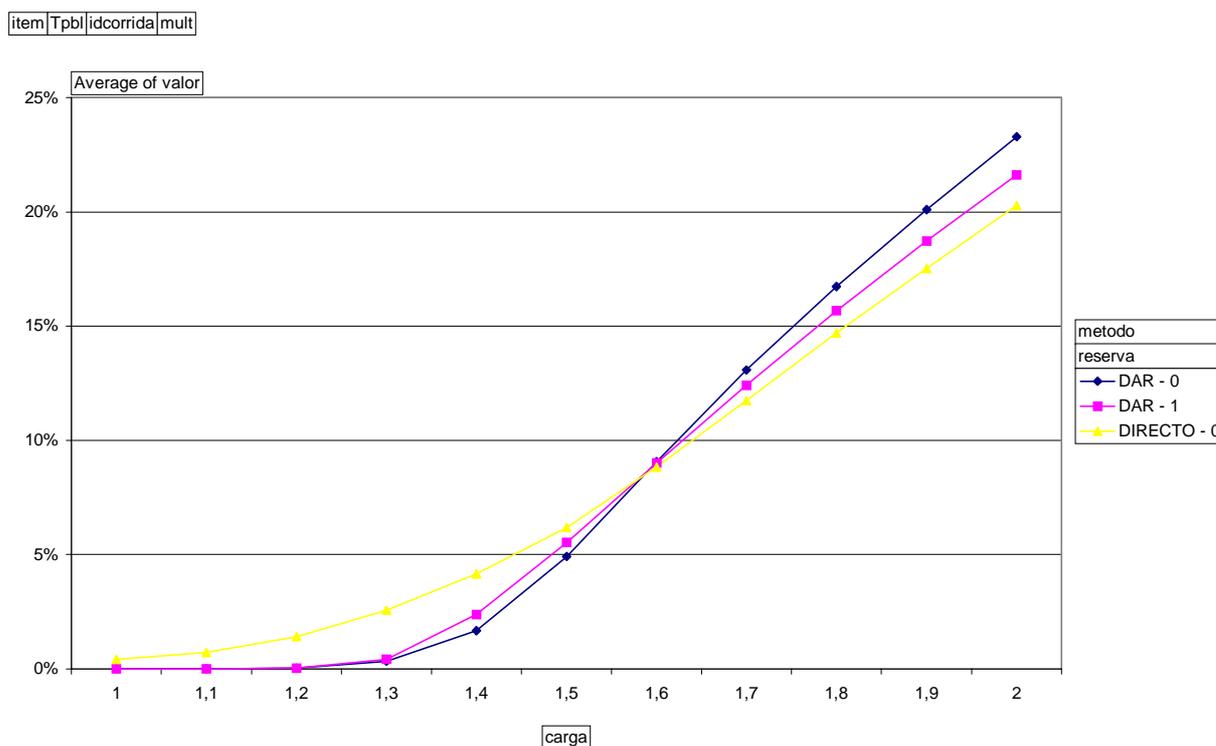
Gráf. 6-10 – Congestão de chamadas na rede (sistema 2).

Os resultados apresentados no gráfico 6-10 mostram que para factores de sobrecarga inferiores a 1,5 o método de encaminhamento DAR e RTNR têm um desempenho global bastante semelhante e significativamente superior ao método de encaminhamento DIRECTO (para factores inferiores a 1,4 o desempenho do RTNR é melhor que o do DAR, mas possui geralmente intervalos de confiança bastante amplos, como visível na tabela G-9 do apêndice G ). Para factores de sobrecarga superiores a 1,5 tanto o DAR como o RTNR têm um desempenho inferior ao DIRECTO. Pode-se explicar isto pelo facto de haver nesta situação grande escassez de recursos face aos tráfegos oferecidos, e portanto os métodos que usam encaminhamento alternativo, (e para tal consomem mais recursos) se tornam ineficientes, tratando-se duma rede totalmente emalhada.

Repare no ficheiro de topologia, utilizado nesta experiência, em que foi usado o valor 0 para a reserva de circuitos fixa em todos os feixes para as duas classes. No método de encaminhamento DAR, isto

corresponde a utilizar reserva de circuitos fixa (original do DAR), mas com um valor de reserva nulo em todos os feixes (o método RTNR não utiliza reserva fixa, pelo que esses valores são ignorados pelo método).

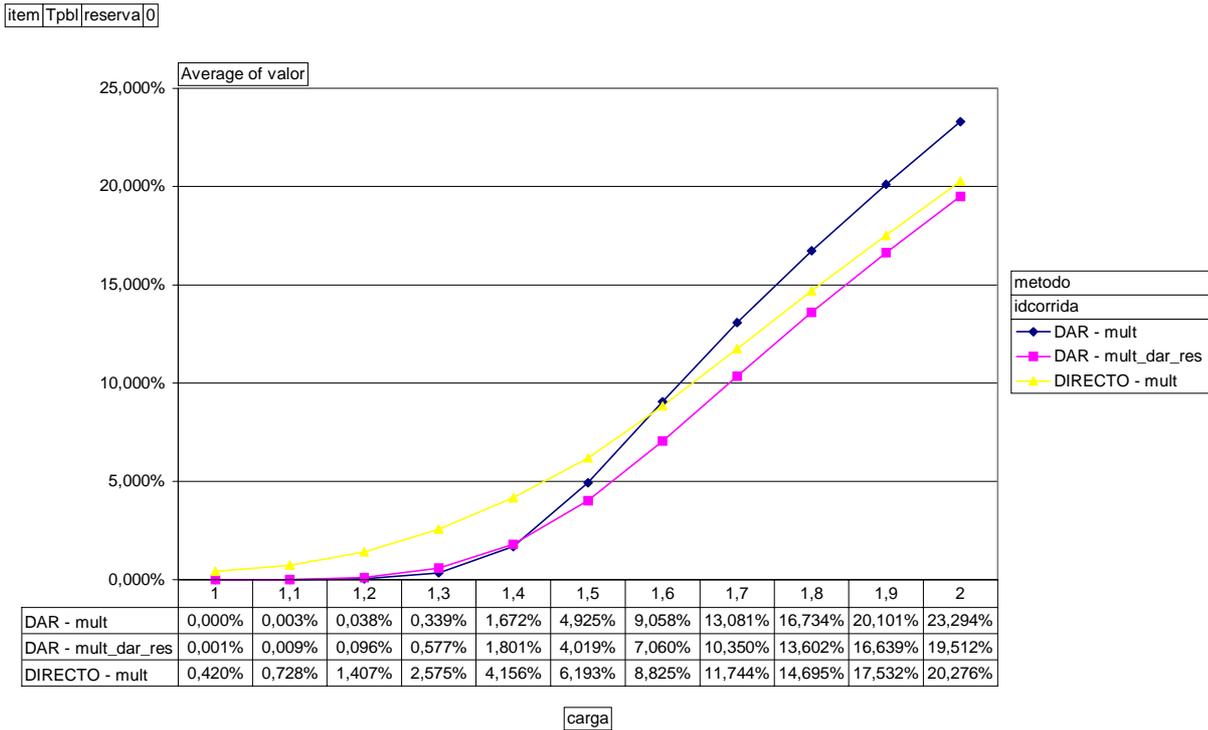
O gráfico 6-11 ilustra também os valores das probabilidades de bloqueio global na rede para vários factores de sobrecarga para o método de encaminhamento DAR mas agora para ambos os tipos de reserva de circuitos que ele pode utilizar.



Gráf. 6-11 – Congestão de chamadas na rede, para dois tipos de reserva no DAR (sistema 2).

Também aqui, como no caso monoclasse (sistema 1), se pode observar uma melhoria no desempenho do DAR quando foi usado o tipo de reserva de circuitos 1 (reserva dinâmica estilo RTNR), como pode ser observado no gráfico 6-11, embora aqui tal apenas ocorra para cargas superiores a 1,6. No entanto tal levou a um desempenho pior para as cargas 1,3 a 1,5 inclusive, ao contrário do que se passava no caso monoclasse (sistema 1), e para cargas inferiores não é possível distinguir os desempenhos para um grau de confiança de 95%, o que é visível na tabela G-10 do apêndice G. O desempenho do DAR, mesmo com este tipo de reserva, assim como aconteceu com o RTNR, também é inferior ao DIRECTO para factores de sobrecargas superiores a 1,6.

Alterando o valor da reserva de circuitos fixo, no ficheiro de topologia, para os valores apresentados na tabela 6-3 e executando uma nova simulação, os resultados são os apresentados no gráfico 6-12. Os valores da tabela 6-3 foram escolhidos arbitrariamente, sem ter sido feita uma análise detalhada que pudesse apontar para outros valores.

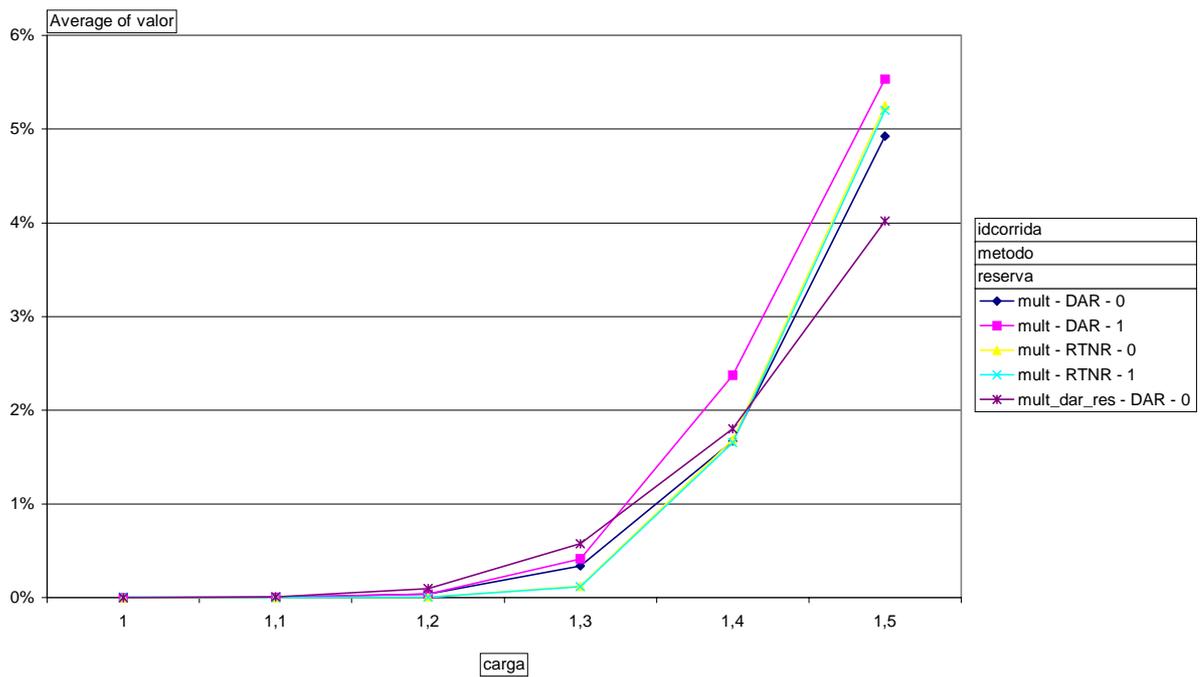


Gráf. 6-12 – Congestão de chamadas na rede, com reserva de circuitos fixa diferente de 0 (sistema 2).

No gráfico 6-12 só são apresentados os resultados para o DAR pois é apenas este que utiliza a reserva de circuitos fixa. Podemos observar que o desempenho do DAR melhorou bastante, para factores de sobrecarga superiores a 1,4, no entanto a diferença entre o seu desempenho e o do DIRECTO vai diminuindo à medida que o factor de sobrecarga aumenta. Estes resultados são bastante notórios na tabela G-11 do apêndice G.

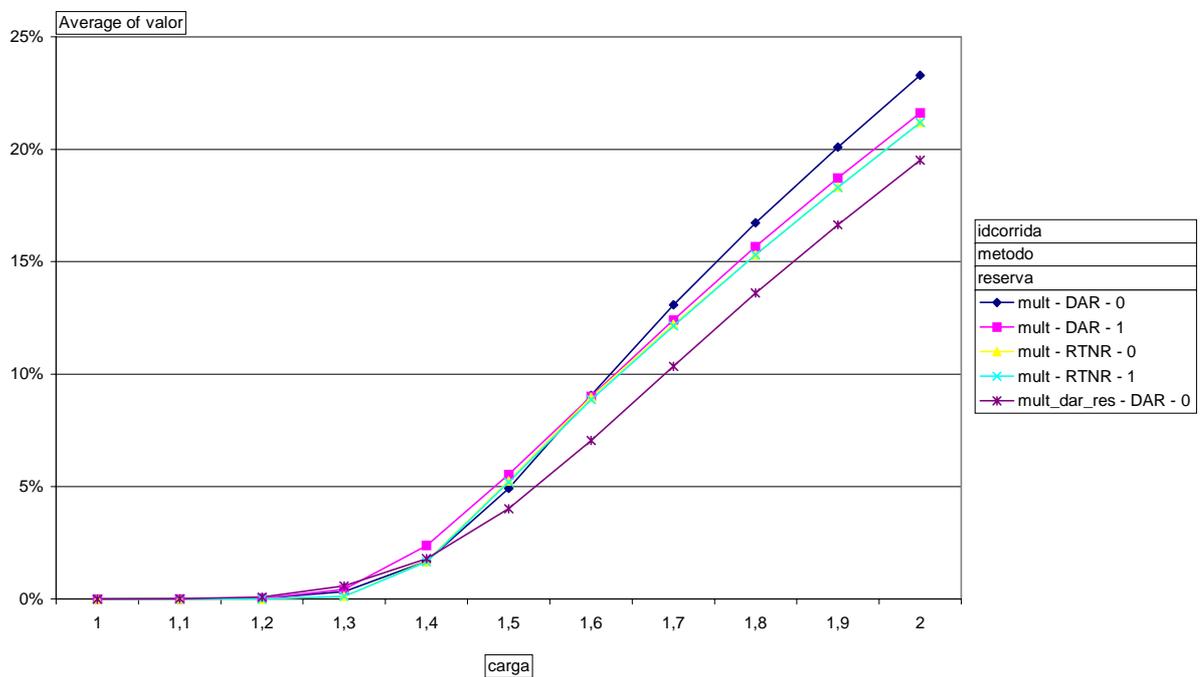
Nos gráficos seguintes, foi decidido representar o comportamento das diversas variações dos métodos DAR e RTNR para as diferentes cargas.

item | Tpb | fluxos | 2



Gráf. 6-13 – Probabilidade de bloqueio global de chamadas na rede, para os métodos DAR e RTNR (sistema 2) – cargas reduzidas e médias.

item | Tpb |



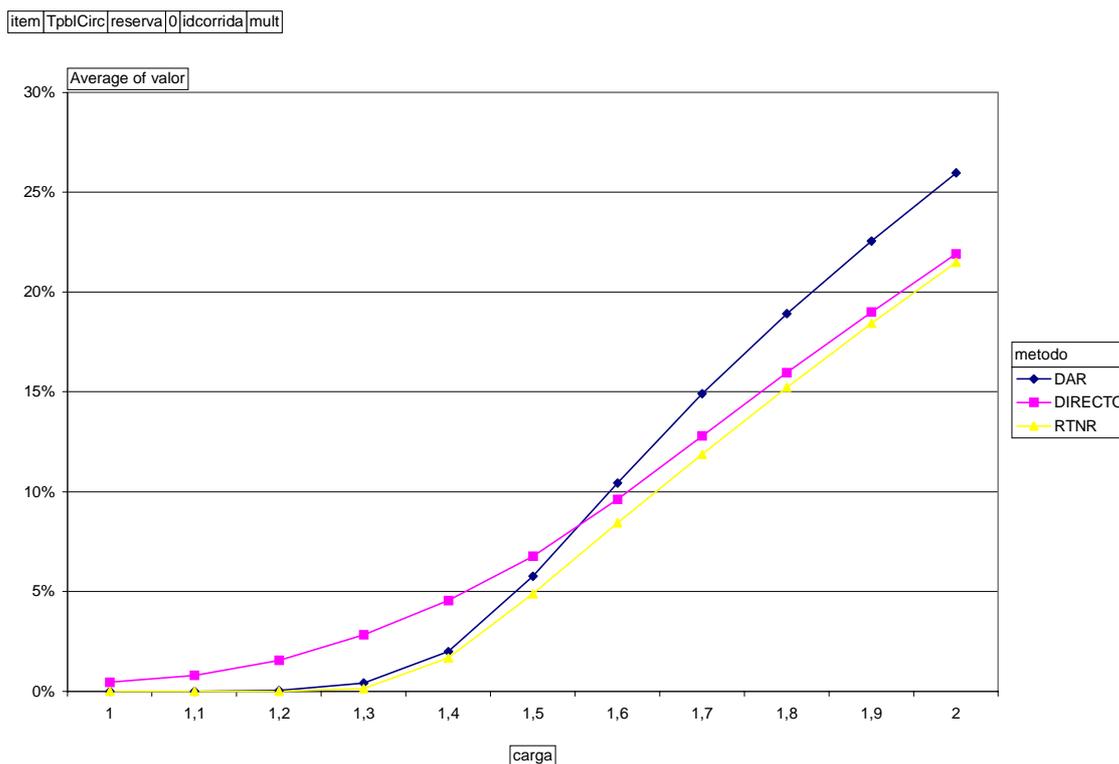
Gráf. 6-14 – Probabilidade de bloqueio global de chamadas na rede, para os métodos DAR e RTNR (sistema 2).

Como se pode verificar nos gráficos 6-13 e 6-14, o desempenho dos dois casos do método RTNR (reserva “original” e “simplificada”) é praticamente indistinguível (conforme se verifica na tabela G-12

do apêndice G, os intervalos de confiança dos dois estão quase completamente sobrepostos). Por esta razão não serão distinguidos os resultados do RTNR na análise seguinte. O mesmo já não se pode dizer do DAR, com variações significativas de desempenho de acordo com o tipo e parâmetros de reserva de circuitos usado. De acordo com os gráficos, verifica-se que o RTNR é o mais vantajoso para sobrecargas baixas (inferiores a 1,4 de factor de carga), enquanto que para sobrecargas mais elevadas (1,5 - e superiores) o DAR com reserva de circuitos fixa não nula é o melhor (embora seja nitidamente o pior de todos para cargas inferiores a 1,4). Observa-se que no ponto de inflexão 1,4 existe uma sobreposição entre os intervalos de confiança do método RTNR e do método DAR com reserva de circuitos fixa nula (e até esta carga, não existem outras variações do DAR com melhores valores de desempenho).

Note-se que a congestão de chamadas global na rede pode não ser uma boa medida de desempenho da rede pois é dada a mesma importância às chamadas de todas as classes. Provavelmente as classes com maiores necessidades trarão maiores receitas e por outro lado pode ser desejado fornecer um grau de serviço da mesma ordem de grandeza a todos os clientes dos vários tipos de chamadas. Uma medida provavelmente mais correcta será a probabilidade de bloqueio global de circuitos (equivalente de chamadas monocanal) na rede. Vamos de seguida estudar os comportamentos deste sistema de acordo com esta medida.

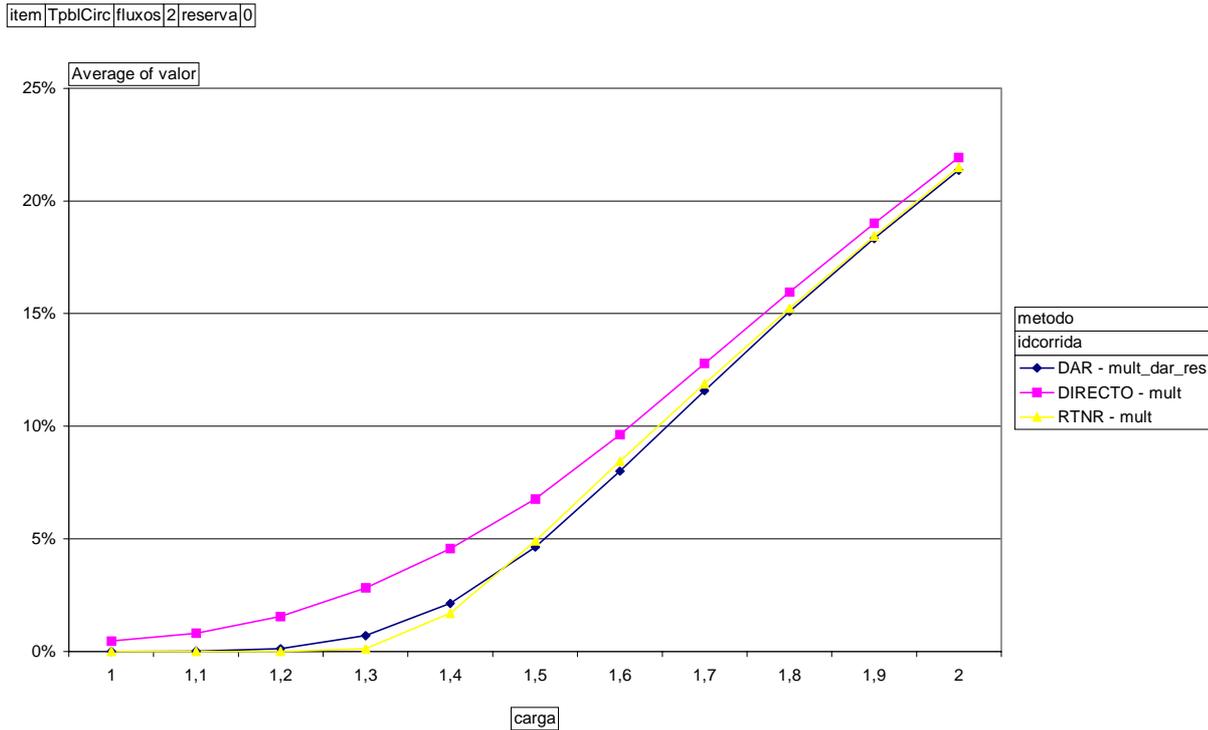
O gráfico 6-15 ilustra as probabilidades de bloqueio global de circuitos (equivalente de chamadas monocanal) na rede para vários factores de sobrecarga para cada um dos três métodos de encaminhamento (DAR, RTNR e DIRECTO), de acordo com as condições iniciais expressas no gráfico 6-10.



Gráf. 6-15 – Probabilidade de bloqueio global de circuitos na rede (sistema 2).

Ao observarmos o gráfico 6-15 podemos verificar que para todos os factores de sobrecarga o RTNR é nitidamente superior quer ao DAR quer ao DIRECTO e que para factores de sobrecarga muito altos (>1,6) o comportamento do RTNR aproxima-se ao do DIRECTO, divergindo marcadamente do DAR. Verificamos ainda que o RTNR parece bastante melhor do que fazia pensar o gráfico 6-10. Estes resultados são bastante aparentes na tabela G-13 do apêndice G.

Anteriormente, no gráfico 6-12 verificou-se que para o método DAR o melhor desempenho ocorria para cargas maiores que 1,4 quando era utilizado uma reserva de circuitos fixos diferentes de 0. Por esta razão, vamos agora observar o comportamento desse método nessa situação em relação ao RTNR e ao DIRECTO, usando como medida de desempenho a probabilidade de bloqueio global de circuitos (equivalente de chamadas monocanal) na rede.



Gráf. 6-16 – Probabilidade de bloqueio global de circuitos na rede, com reserva de circuitos fixa diferente de 0 para o DAR (sistema 2).

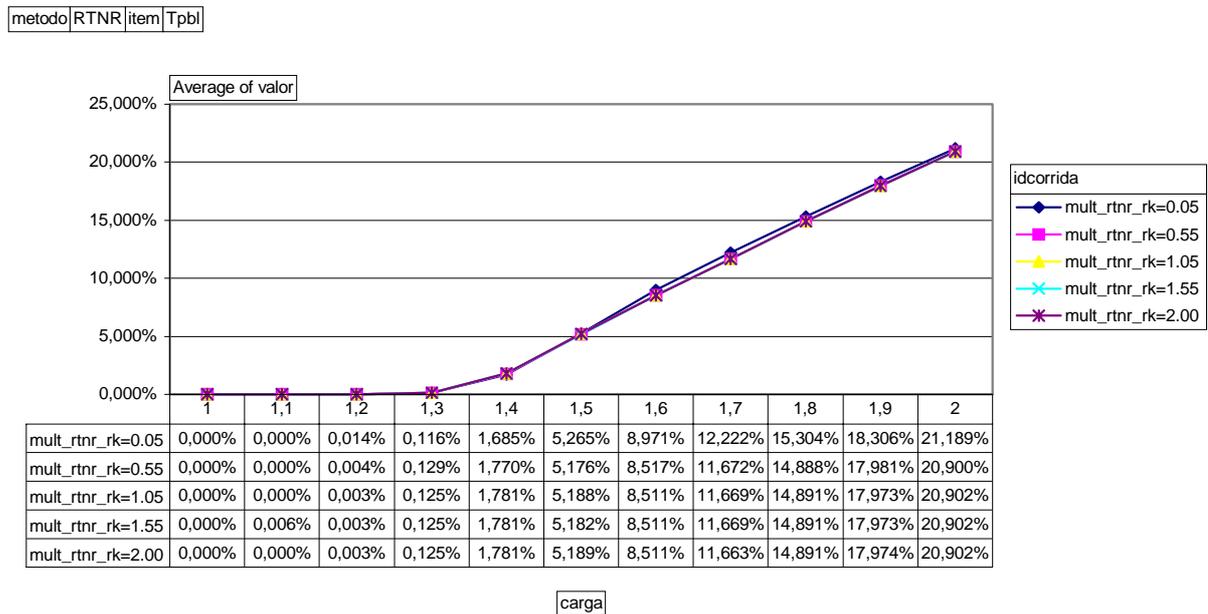
No gráfico 6-16 podemos observar que o DAR com reserva de circuitos fixa diferente de 0, tem um comportamento ligeiramente inferior ao RTNR (mas superior ao DIRECTO) para sobrecargas reduzidas (inferiores a 1,5). Para sobrecargas elevadas, e de acordo com a tabela G-14 do apêndice G, é possível verificar que o DAR é superior ao RTNR para os factores de sobrecarga de 1,5 a 1,9. Existe sobreposição dos intervalos de confiança para o factor de sobrecarga 2,0.

## 6.4.2 Estudo da variação dos valores dos parâmetros específicos do RTNR, no RTNR

O desempenho deste sistema com o método de encaminhamento RTNR deteriora-se demasiado quando o factor de sobrecarga ultrapassa o valor 1,4. Por esta razão fomos examinar o desempenho da rede sob a variação de alguns parâmetros do método. Primeiro examinámos o factor do limiar do nível de reserva- $R_k$  (ver secção 3.5.1.3.1, capítulo 3), Factor2 na expressão 6-2, secção 6.2.1, o qual foi considerado com o valor 0.05 na tabela 3-1 do capítulo 3.

Foram obtidos os valores das probabilidade de bloqueio global no RTNR, para vários factores do limiar do nível de reserva- $R_k$  e para vários valores de sobrecarga. No gráfico 6-17 estão apenas alguns dos

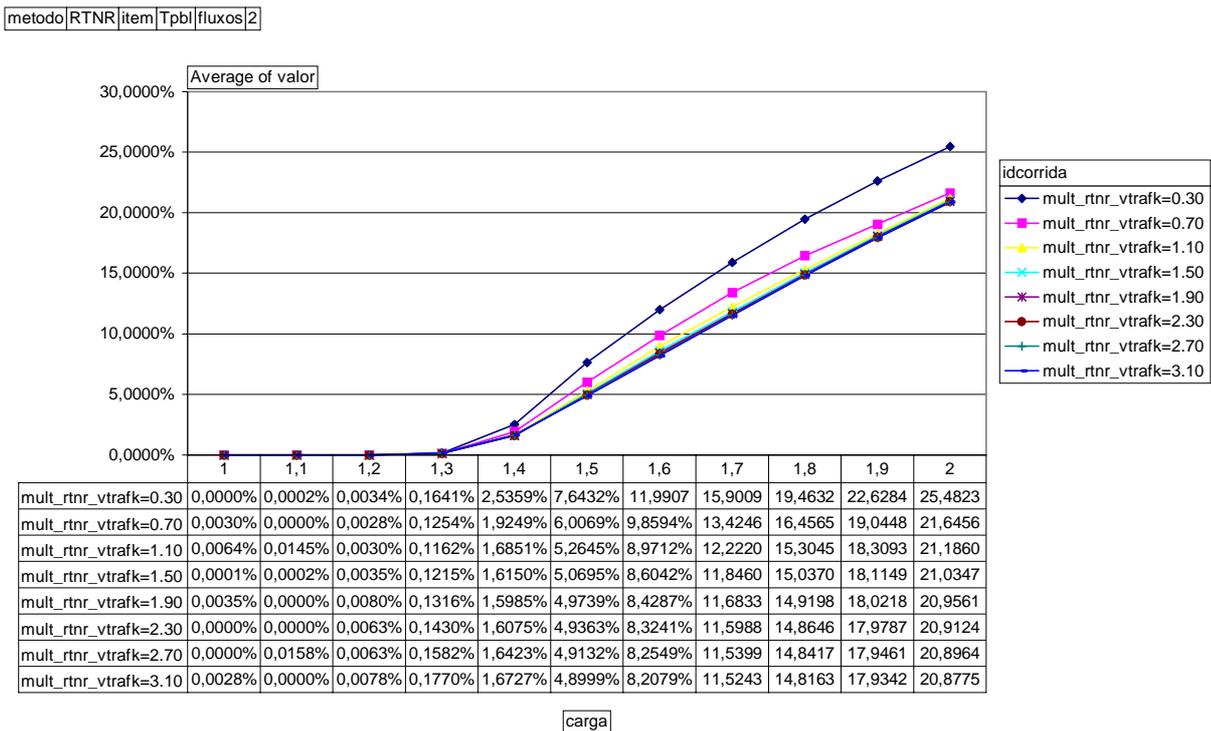
resultados obtidos pois no estudo o intervalo de variação do factor do limiar do nível de reserva-  $R_k$  foi inferior ao apresentado no gráfico.



Gráf. 6·17 – Probabilidade de bloqueio global no RTNR versus o factor do  $R_k$  (sistema 2).

Os resultados do gráfico 6·17 indicam que o nível de reserva de circuitos exacto não tem uma influência substancial no desempenho da rede (embora seja visível na tabela G·15 do apêndice G que a escolha de 0,05 é pior que as restantes para factores de carga superiores a 1,5).

Examinámos então o factor do  $VTraf_k$ , Factor1 na expressão 6·1, secção 6.2.1, o qual foi considerado com o valor 1,1 na expressão 3·3 do capítulo 3. Foram também aqui obtidos resultados para vários valores de sobrecarga, apresentados no gráfico 6·18.



Gráf. 6-18 – Probabilidade de bloqueio global no RTNR versus o factor do  $VTtrafk_k$  (sistema 2).

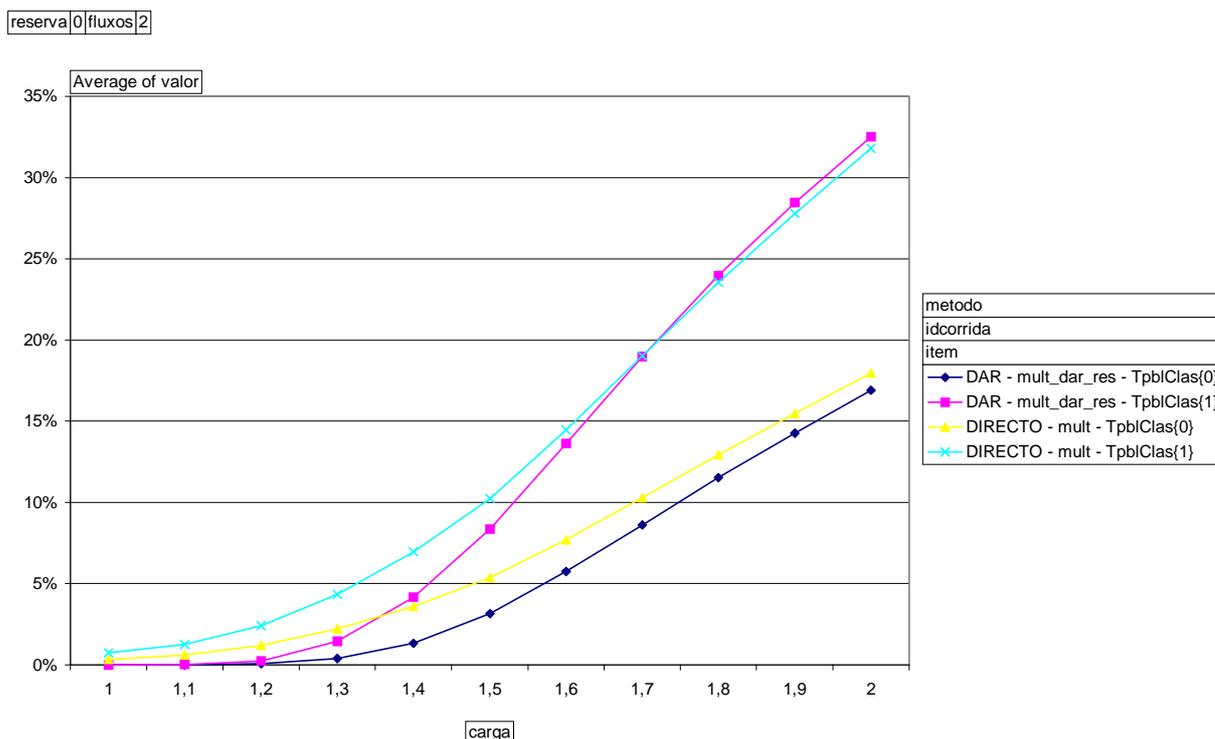
Os resultados do gráfico 6-18 e da tabela G-16 do apêndice G indicam que um factor de 1,1 é uma escolha aceitável para o sistema 2.

De entre os valores examinados para os parâmetros do método não nos foi possível arranjar valores que melhorem significativamente o desempenho da rede.

### 6.4.3 Análise da probabilidade de bloqueio global por classe

Vamos analisar agora os valores das probabilidades de bloqueio global na rede para as chamadas de ambas as classes separadamente.

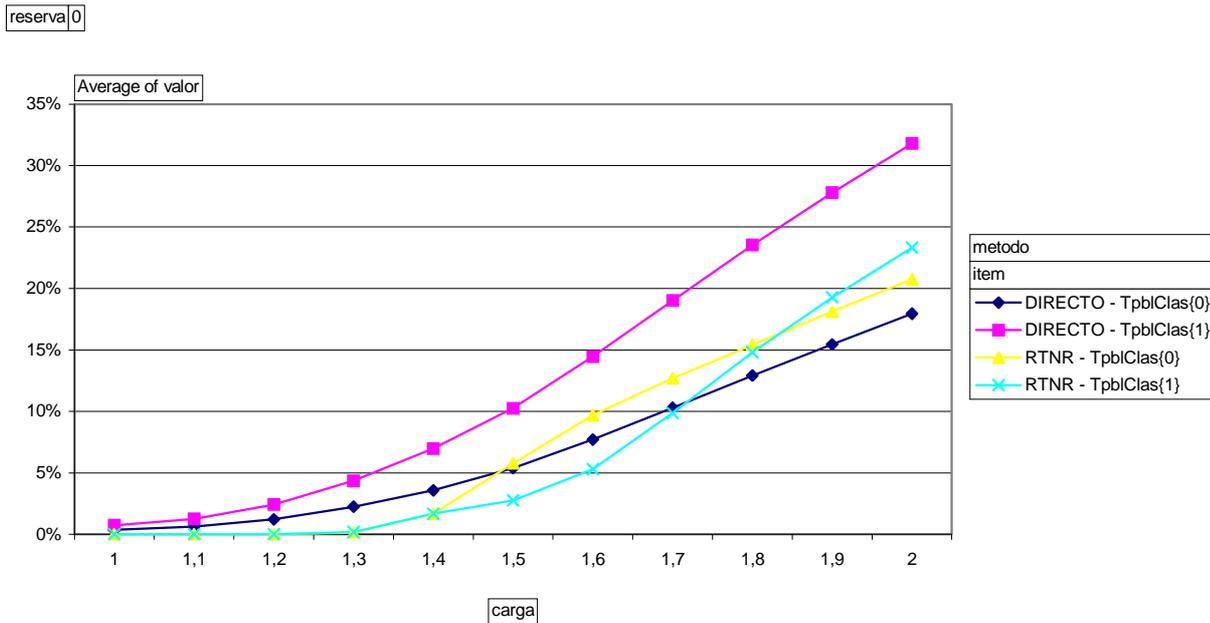
O gráfico 6-19 ilustra a congestão de chamadas na rede para ambas as classes de chamadas e para vários factores de sobrecarga para o método de encaminhamento DAR e DIRECTO.



Gráf. 6-19 – Probabilidade de bloqueio global por classe para o método DAR e DIRECTO (sistema 2).

Pode ser observado no gráfico 6-19 que a probabilidade de bloqueio global é superior para a classe com maior necessidade de recursos, o que era de esperar pois não foi usado nenhum mecanismo de protecção dos serviços. Podemos ver também que, para ambas as classes, as probabilidades de bloqueio global com o método de encaminhamento DAR são bastante inferiores às probabilidades de bloqueio global com o método de encaminhamento DIRECTO, para factores de sobrecargas inferiores a 1,5, como se atesta na tabela G-17 do apêndice G. Para factores de sobrecarga superiores a 1,5, à medida que o seu valor aumenta, aumenta também a proximidade entre os valores das probabilidades de bloqueio global, entre os dois métodos de encaminhamento, para cada uma das classes (chegando o método DIRECTO a conseguir melhores valores para o bloqueio na classe 1 do que o DAR). No entanto, conforme visível no gráfico 6-12 tal não chega para levar o bloqueio global a ser inferior para o método DIRECTO.

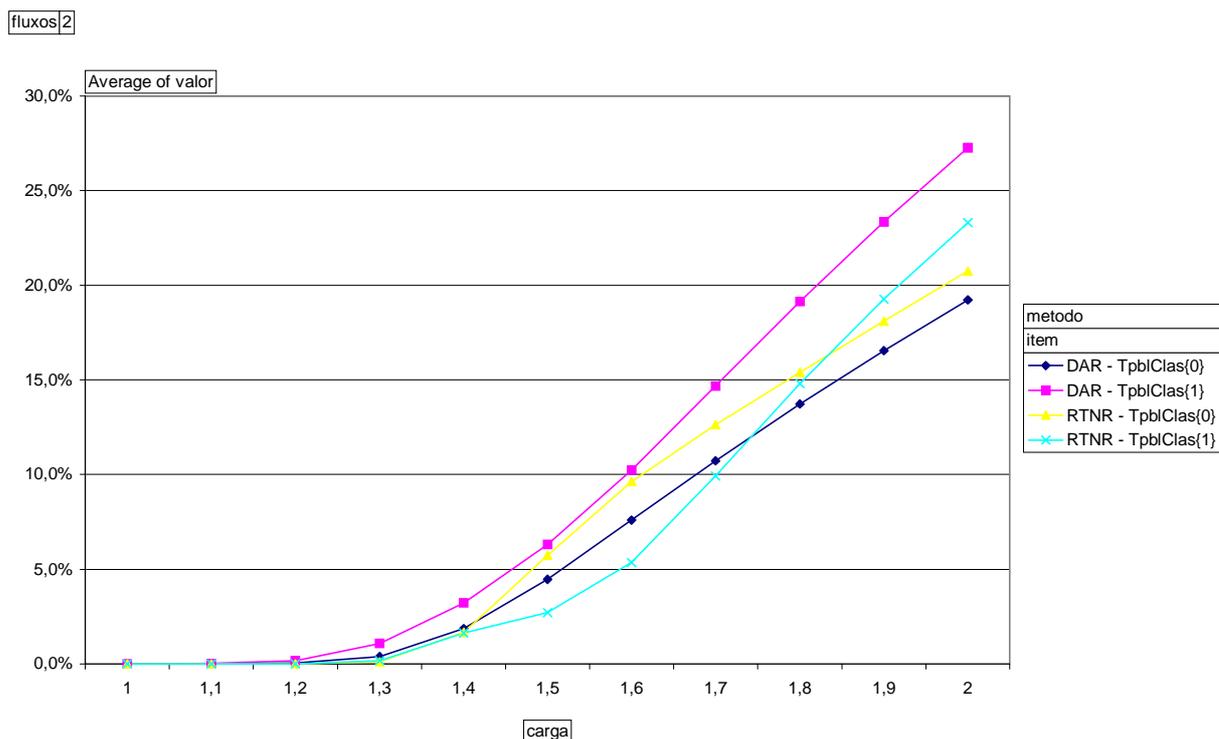
O gráfico 6-20 ilustra a congestão de chamadas na rede para ambas as classes de chamadas quando foi utilizado o método de encaminhamento RTNR e DIRECTO, para vários factores de sobrecarga.



Gráf. 6-20 – Probabilidade de bloqueio global por classe para o método RTNR e DIRECTO (sistema 2).

Pode ser observado no gráfico 6-20 que com o método de encaminhamento RTNR para factores de sobrecarga de 1,4 a 1,8, a probabilidade de bloqueio global é superior para a classe com menor necessidade de recursos. O mecanismo de protecção dos serviços usado pelo RTNR parece estar a proteger demasiado a classe com largura de banda superior. Isto, como é obvio, aumenta a congestão global de chamadas na rede, como se observou no gráfico 6-10, pois se as chamadas encaminhadas consomem mais recursos o número de chamadas que será possível encaminhar será menor. Na classe com menor necessidade de recursos, para factores de sobrecarga superiores a 1,5 as probabilidades de bloqueio global, com o método de encaminhamento RTNR são superiores às probabilidades de bloqueio global com o método de encaminhamento DIRECTO. Os intervalos de confiança para estes valores podem ser consultados na tabela G-18 do apêndice G.

Para possibilitar a comparação explícita dos dois métodos de encaminhamento alternativo dinâmico usados, no gráfico 6-21 é apresentada a congestão de chamadas na rede para as duas classes de chamadas e para vários factores de sobrecarga para os métodos de encaminhamento DAR e RTNR (escolhemos as variações destes métodos usadas nos gráficos anteriores).



Gráf. 6-21 – Probabilidade de bloqueio global por classe para o método DAR e RTNR (sistema 2).

Da observação do gráfico 6-21 verificamos que a congestão de chamadas para o método RTNR é bastante baixa e semelhante para as duas classes, para cargas baixas (inferiores a 1,4). À medida que a carga aumenta o bloqueio aumenta mais na classe de menor necessidade de recursos, aumentando porém de forma mais brusca o bloqueio na classe 1 a partir do factor de sobrecarga 1,6, chegando inclusivamente a ultrapassar a classe 0 para cargas superiores a 1,8. Por outro lado com o DAR as variações são mais suaves, e a classe 1 apresenta sempre índices de bloqueio superior à classe 0, aumentando esta diferença com a carga. Por análise dos intervalos de confiança na tabela G-19 do apêndice G, é possível verificar que o RTNR comporta-se melhor que o DAR para as duas classes em factores de sobrecarga inferiores a 1,4.

Para facilitar a comparação por classe de chamada do comportamento relativo DAR (*mult\_dar\_res*) e RTNR para diferentes factores de sobrecarga,  $\rho$ , a tabela 6-8 apresenta uma análise simplificada do comportamento destes dois métodos, indicando para que factores de sobrecarga um método possui um melhor desempenho (+), ou tem um pior desempenho (-). Estes resultados são nítidos na tabela G-19 do apêndice G, existindo sobreposição dos intervalos apenas para a classe 0 na carga nominal. O ponto de inflexão foi estimado aproximadamente, por interpolação linear.

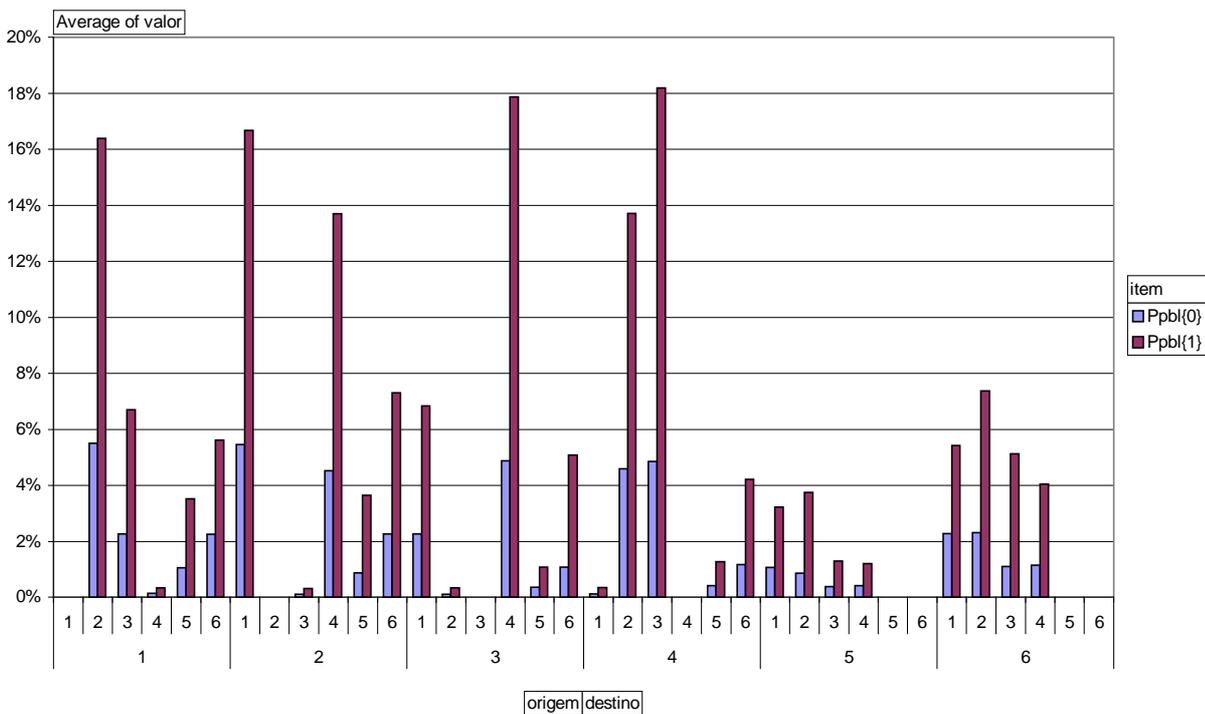
Classe	$\rho$ baixo e médio-baixo		Inflexão	$\rho$ médio-alto e alto	
	DAR	RTNR		DAR	RTNR
0	-	+	1,42	+	-
1	-	+		-	+

Tab. 6-8 – Comparação da probabilidade de bloqueio global por classe para o método DAR e RTNR (sistema 2).

### 6.4.4 Análise da probabilidade de bloqueio ponto a ponto por classe

Neste ponto vamos analisar os valores das probabilidades de bloqueio ponto a ponto para as chamadas de ambas as classes. O gráfico 6-22 ilustra esses valores para o método de encaminhamento DAR, para um factor de sobrecarga igual a 1,4.

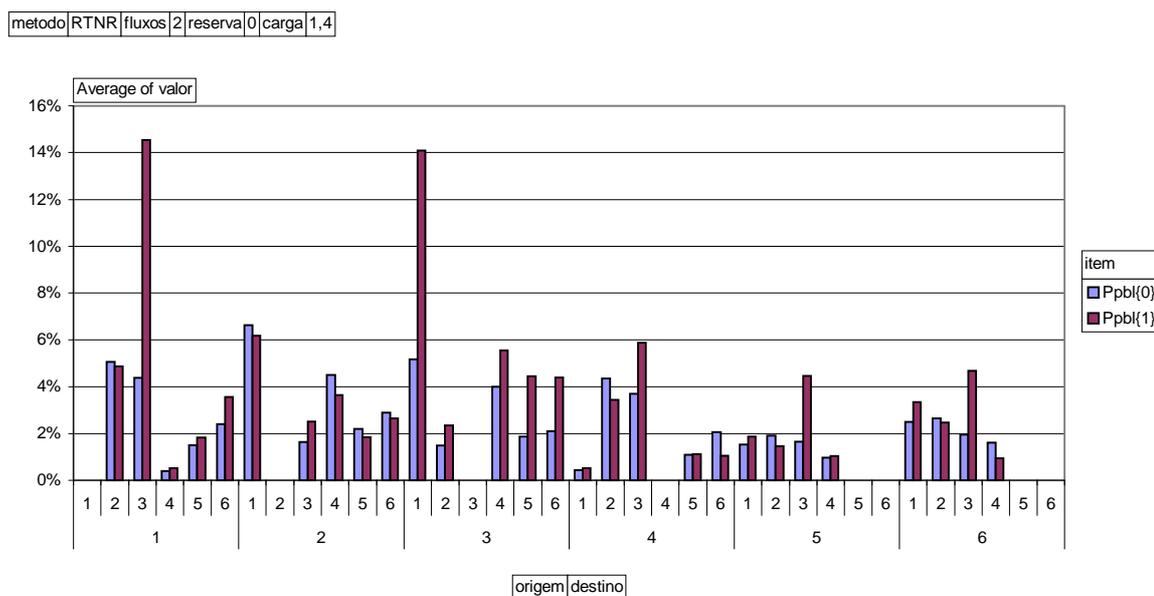
metodo DAR fluxos 2 reserva 0 carga 1,4



Gráf. 6-22 – Probabilidade de bloqueio ponto a ponto, por classe, para o método DAR (sistema 2).

Pode ser observado no gráfico 6-22 que a probabilidade de bloqueio ponto a ponto é sempre superior para a classe com maior necessidade de recursos, como verificado na tabela G-20 do apêndice G. Esta observação é válida para qualquer outro factor de sobrecarga.

O gráfico 6-23 ilustra as probabilidades de bloqueio ponto a ponto para as chamadas de ambas as classes com o método de encaminhamento RTNR, para um factor de sobrecarga igual a 1,4.



Gráf. 6-23 – Probabilidade de bloqueio ponto a ponto, por classe, para o método RTNR (sistema 2).

Pode ser observado no gráfico 6-23 que a probabilidade de bloqueio ponto a ponto para uma classe não é sempre superior à da outra classe, conforme verificado na tabela G-21 do apêndice G, o que acontece no DAR. Para qualquer outro factor de sobrecarga é frequente existir uma maioria de probabilidades de bloqueio ponto a ponto de uma classe superiores às da outra; no entanto a observação feita para o gráfico 6-23 continua válida.

Uma análise adicional deste sistema relativa ao comportamento do bloqueio nos feixes para os métodos DAR e RTNR pode ser encontrada no apêndice H, secção H.1.

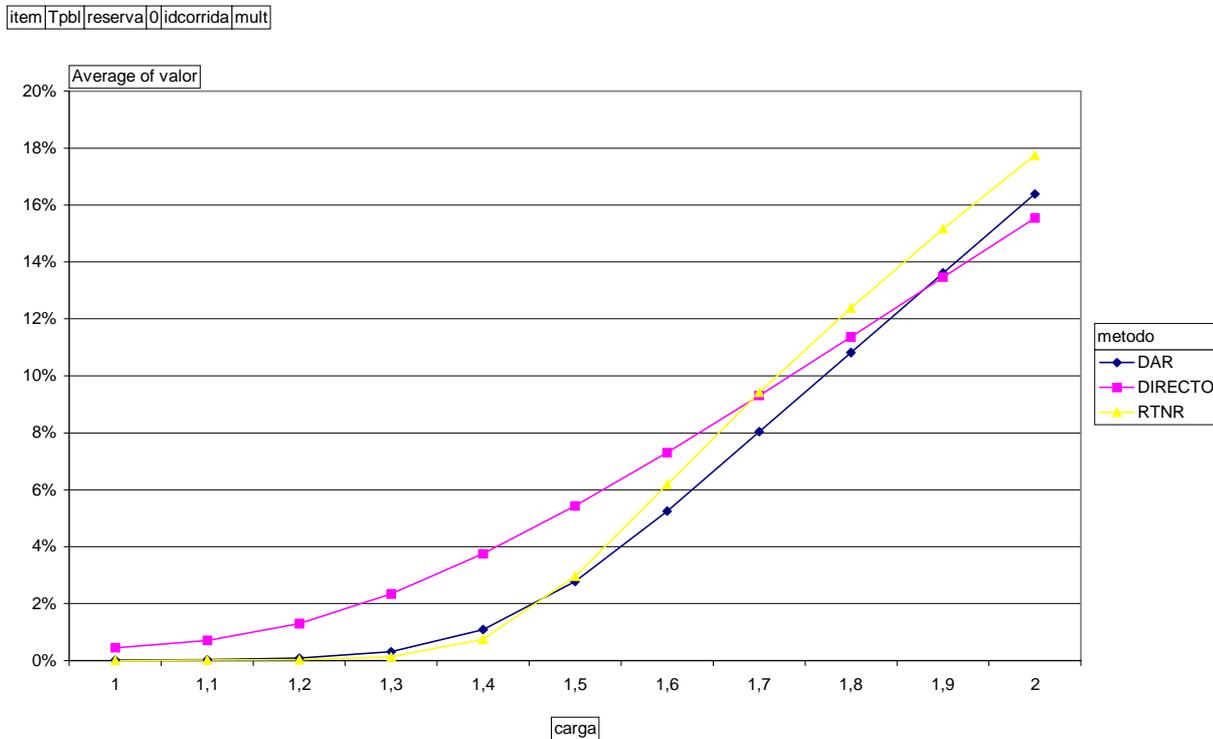
## 6.5 Experiência computacional com o Sistema 3

Por fim, vamos apresentar alguns dos resultados obtidos por simulação com o sistema 3, definido também no capítulo 4. No apêndice C, na secção C.1.2, encontra-se o ficheiro de topologia (ficheiro *txt*) que foi usado nesta experiência no qual foram introduzidas as características da rede que se pretende simular.

Nos gráficos que se seguem o campo *idcorrida* é utilizado (como foi também para o sistema 2) para fazer a distinção entre corridas onde houve alteração nos valores das reservas de circuitos. Exceptua-se o caso do estudo da variação do valor do parâmetro *Factor1* do RTNR (usado na expressão 6-1 deste capítulo), onde foi utilizado para permitir a distinção entre as várias corridas para os diferentes valores desse parâmetro. Os valores dos parâmetros, comuns a todas as classes, utilizados nesta secção continuam a ser os definidos na secção 6.2 a menos que sejam apresentados outros.

## 6.5.1 Análise da probabilidade de bloqueio global na rede

O gráfico 6-24 mostra-nos as probabilidades de bloqueio global na rede. Neste gráfico são ilustradas as probabilidades de bloqueio para vários factores de sobrecarga para cada um dos três métodos de encaminhamento (DAR, RTNR e DIRECTO).

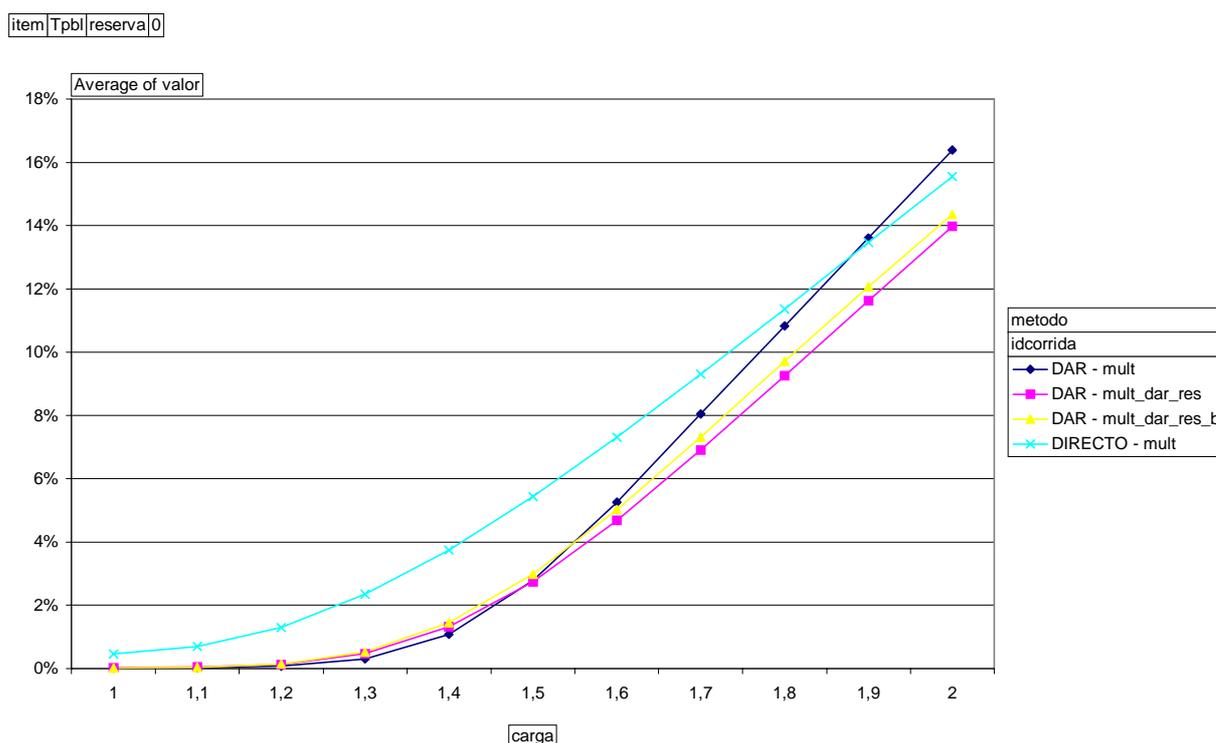


Gráf. 6-24 – Congestão de chamadas na rede (sistema 3).

Pela observação do gráfico 6-24 verificamos que os métodos de encaminhamento DAR e RTNR têm comportamentos semelhantes para factores de sobrecarga inferiores a 1,5 (embora o RTNR seja ligeiramente superior para factores de sobrecarga de 1,1 a 1,4, inclusivé e haja uma sobreposição para o factor 1,0, conforme se verifica na tabela G-22 do apêndice G) e bastante superiores ao método de encaminhamento DIRECTO. Para factores de sobrecarga superiores a 1,5, à medida que a carga aumenta, diminui o desempenho do DAR e do RTNR tornando-se, em ambos os casos, inferior ao método de encaminhamento DIRECTO. Esta situação ocorre no DAR para um factor de sobrecarga claramente superior àquele em que ocorre no RTNR. Deve ser notado que para factores de carga superiores a 1,5 o desempenho do método DAR é claramente superior ao do RTNR.

Repare-se que foi usado o valor 0 para a reserva de circuitos fixa em todos os feixes para as quatro classes. Esse valor é utilizado pelo método de encaminhamento DAR. Alterámos esses valores, no ficheiro de topologia, para os valores apresentados na tabela 6-3 e executámos uma nova simulação, o que conduziu aos resultados apresentados no gráfico 6-25 (`idcorrida=mult_dar_res`). Nesse gráfico

são também apresentados os resultados para o valor do campo `idcorrida=mult_dar_res_b` que corresponde a outra alteração dos valores da reserva de circuitos fixa. Esta alteração consistiu em manter a 0 o valor para a reserva de circuitos fixa em todos os feixes para as duas classes com maior necessidade de recursos e alterando apenas os valores das duas classes restantes para os valores respectivos apresentados na tabela 6-3. Como já referido os valores da tabela 6-3 foram escolhidos arbitrariamente, sem ter sido feita uma análise detalhada prévia. Ao manter o valor 0 para a reserva de circuitos fixa em todos os feixes para as duas classes com maior necessidade de recursos, numa das experiências, pretendíamos diminuir a probabilidade de bloqueio para essas classes.

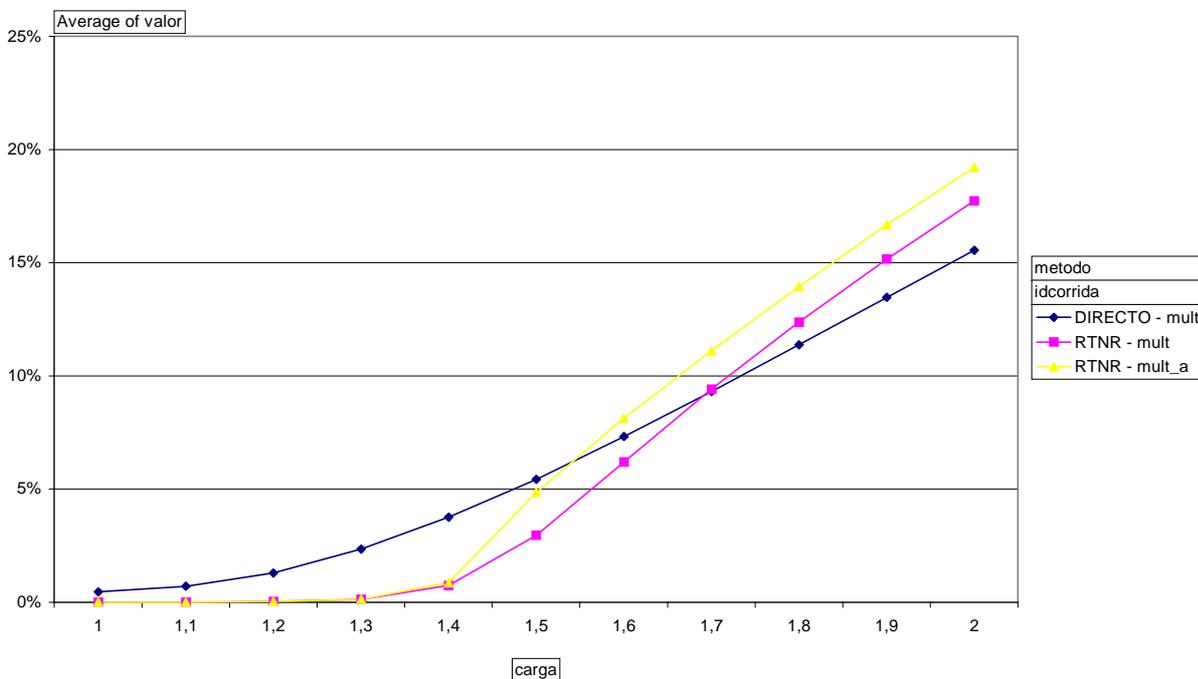


Gráf. 6-25 – Congestão de chamadas na rede, com o DAR, para reserva de circuitos fixa diferente de 0 (sistema 3).

O gráfico 6-25 ilustra uma melhoria acentuada no método de encaminhamento DAR, para as alterações feitas nos valores da reserva de circuitos fixa, para cargas superiores a 1,5. Vemos ainda que, para as alterações feitas, e para cargas superiores a 1,5 o comportamento global do DAR foi superior quando foram escolhidos valores diferentes de 0 para as reserva de circuitos fixa para todas as classes. No entanto, como se pode verificar na tabela G-23 do apêndice G, o método de encaminhamento DAR com reserva fixa de circuitos nula para todas as classes (`idcorrida=mult`) é superior para cargas inferiores a 1,5. Pode-se também verificar nessa tabela que a partir da carga 1,2, a utilização de reserva nula para as duas classes de maior necessidade de circuitos leva a um desempenho pior que o da reserva de circuitos não nula para todas as classes.

O gráfico 6-26 mostra-nos também as probabilidades de bloqueio global na rede mas apenas para o método de encaminhamento RTNR (para duas escolhas distintas de larguras de banda mínimas garantidas para as classes) e método DIRECTO. A razão do estudo que conduziu aos resultados apresentados nesse gráfico foi o termos verificado que os bloqueios por classe são superiores para as classes com maior necessidade de recursos (em geral). Por isto, o que fizemos foi aumentar os valores da largura de banda mínima garantida para as chamadas dessas classes e em contrapartida diminuir esses valores nas classes com menor necessidade de recursos. Essas alterações, feitas no ficheiro de topologia, podem ser observadas no apêndice C na secção C.3.3. A corrida que corresponde ao aumento nos valores da largura de banda mínima garantida, nas classes com maior necessidade de recursos em cada um dos pares de centrais, é a que tem o campo `idcorrida` com o valor `mult_a`.

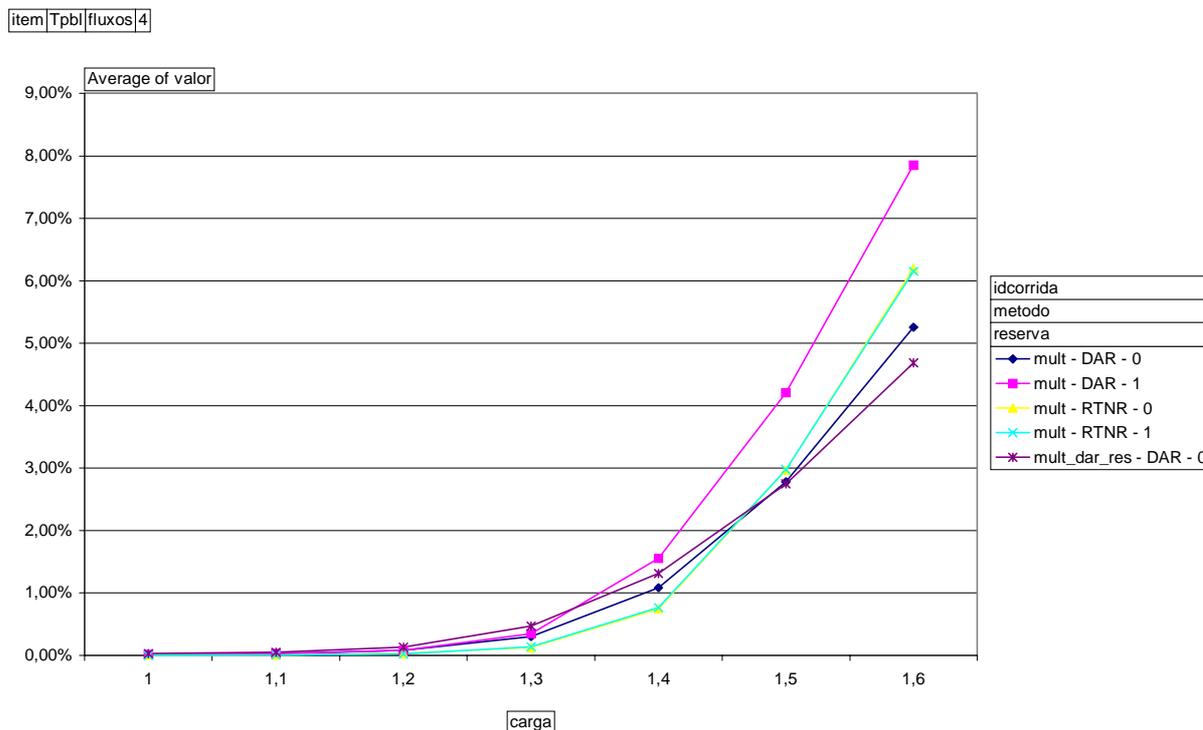
item|Tpbl|reserva|0



Gráf. 6-26 – Congestão de chamadas na rede, com o RTNR, para duas diferentes escolhas dos valores da largura de banda mínima garantida (sistema 3).

Os resultados apresentados no gráfico 6-26 mostram que a alteração feita conduziu a um aumento, bastante significativo da probabilidade de bloqueio global na rede, para factores de sobrecarga superiores a 1,4. Estes resultados assim como os do gráfico 6-25 indicam que ao tentar proteger as classes com maior necessidade de recursos a probabilidade de bloqueio global na rede aumenta, quando o tráfego em geral não é muito baixo. O comportamento da probabilidade de bloqueio global por classe será analisado numa secção seguinte. A tabela G-24 do apêndice G indica também que existe uma sobreposição significativa dos intervalos de confiança para os dois diferentes casos do método RTNR para cargas inferiores a 1,4.

No gráfico 6-27, foi decidido representar o comportamento das diversas variações dos métodos DAR e RTNR para as diferentes cargas.



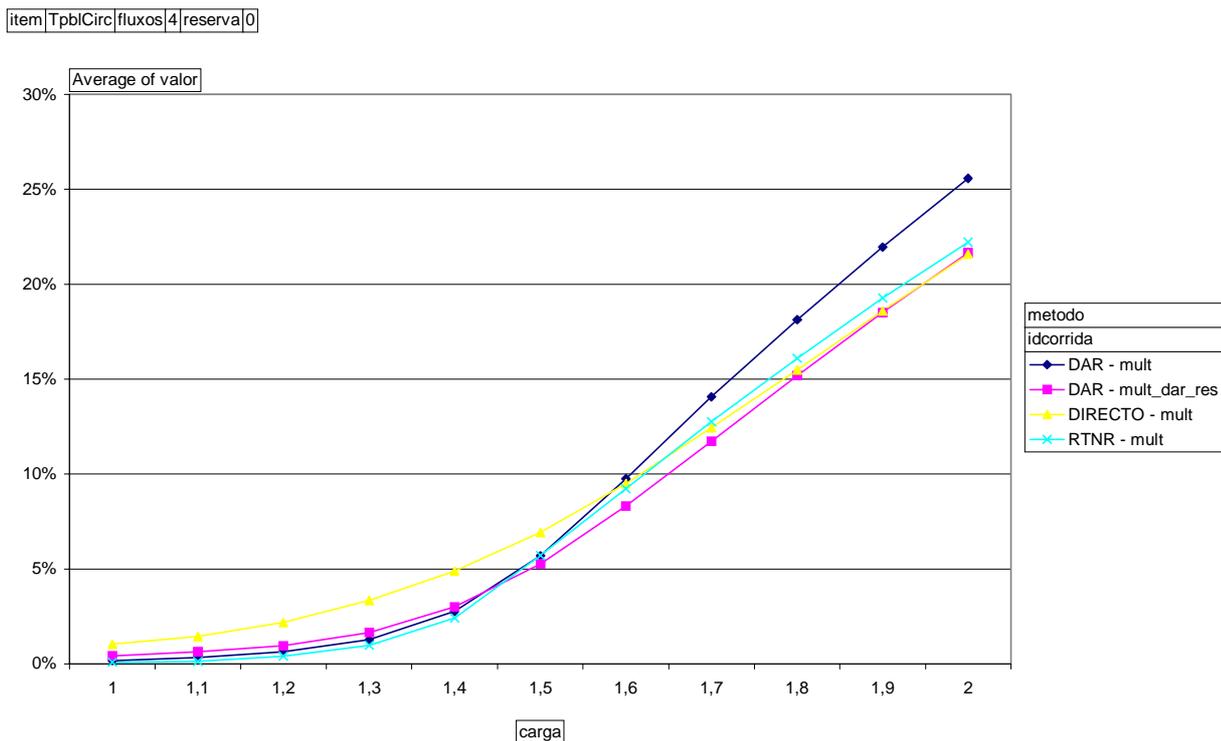
Gráf. 6-27 – Probabilidade de bloqueio global de chamadas na rede, para os métodos DAR e RTNR (sistema 3).

Como se pode verificar no gráfico 6-27, o desempenho das duas versões do método RTNR (reserva “original” e “simplificada”) é praticamente indistinguível (conforme se verifica na tabela G-25 do apêndice G, os intervalos de confiança dos dois estão quase completamente sobrepostos), tal como já ocorria no sistema 2. O mesmo já não se pode dizer do DAR, com variações significativas de desempenho de acordo com o tipo e parâmetros de reserva de circuitos fixa usada. De acordo com o gráfico, verifica-se que o RTNR é ligeiramente mais vantajoso para cargas baixas (inferiores a 1,4 de factor de sobrecarga), enquanto que para sobrecargas mais elevadas (1,5 - e superiores) o DAR com reserva de circuitos fixa não nula é o melhor. Até à carga 1,4 não existem variações DAR que apresentem melhores valores de desempenho do que o método DAR com reserva de circuitos fixa nula. Só são apresentadas as cargas até 1,6 uma vez que o comportamento posterior mantém as mesmas características deste nível de sobrecarga.

Tal como no sistema 2, consideramos que a congestão de chamadas global na rede pode não ser uma boa medida de desempenho da rede pois é dada a mesma importância às chamadas de todas as classes (o que leva a maiores distorções ainda neste sistema, com uma classe em que cada chamada ocupa 24 circuitos). Provavelmente as classes com maiores necessidades trarão maiores receitas e por outro lado pode ser desejado fornecer um grau de serviço da mesma ordem de grandeza a todos os utilizadores dos

vários tipos de chamadas. Uma medida provavelmente mais correcta será a probabilidade de bloqueio global de circuitos (equivalente de chamadas monocanal) na rede, mais ainda neste sistema pela existência de algumas classes cujas chamadas ocupam grande volume de circuitos. Vamos de seguida estudar os comportamentos deste sistema de acordo com esta medida.

O gráfico 6-28 ilustra as probabilidade de bloqueio global de circuitos (bloqueio equivalente de chamadas monocanal) na rede para vários factores de sobrecarga para cada um dos três métodos de encaminhamento (DAR, RTNR e DIRECTO). O método DAR é apresentado para dois níveis de reserva fixa – reserva fixa nula (mult) e fixa não nula (mult\_dar\_res)

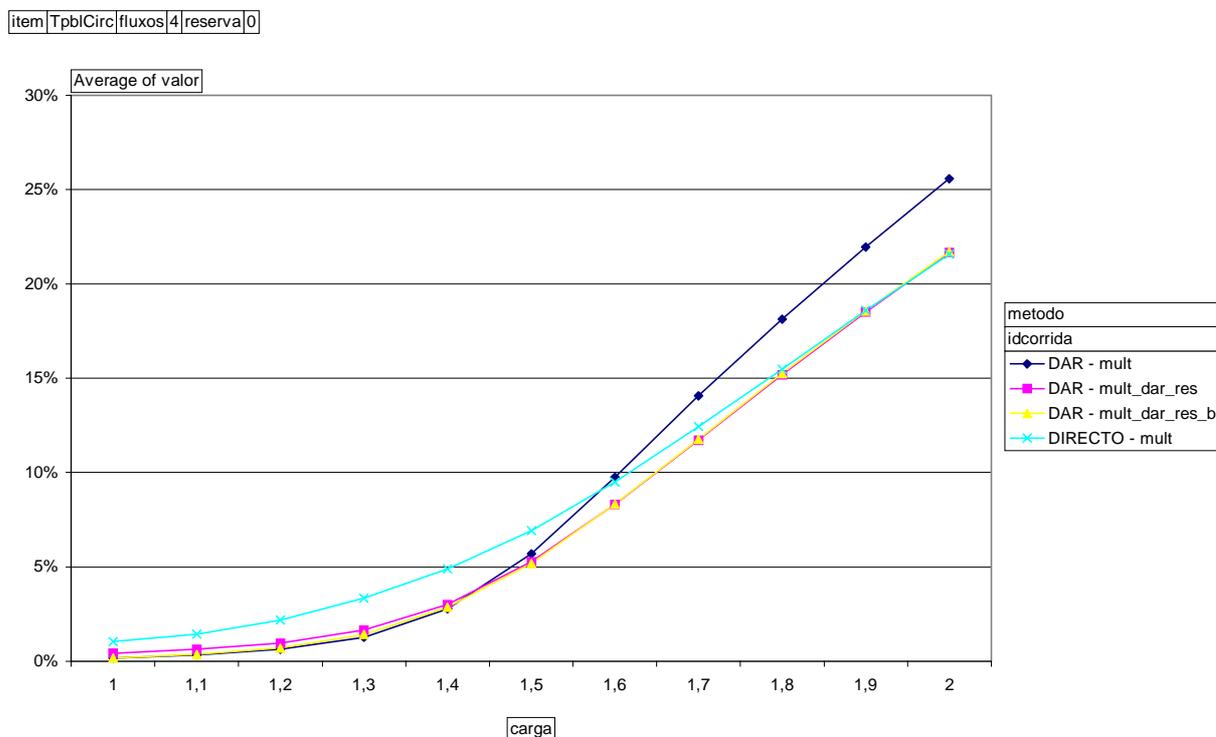


Gráf. 6-28 – Probabilidade de bloqueio global de circuitos na rede (sistema 3).

Ao observarmos o gráfico 6-28 podemos verificar que para factores de sobrecarga baixos (<1,5) o RTNR é um pouco superior ao DAR (qualquer que seja o valor de reserva fixo usado) e que para factores de sobrecarga muito altos (>1,6) o comportamento do DAR e do RTNR é muito próximo ao do DIRECTO. Pode-se verificar com maior clareza na tabela G-26 do apêndice G que o DAR com reserva não nula é superior aos restantes para cargas de 1,5 a 1,7 existindo sobreposição para os intervalos de confiança do DAR (mult\_dar\_res) e DIRECTO para cargas superiores a 1,8. Verifica-se portanto que o RTNR não parece tão mau como faria pensar o gráfico 6-24 e que o DAR não parece tão bom como parecia indicar o gráfico 6-25.

Note que devido ao facto de se estar a utilizar uma medida ponderada pelo número de circuitos ocupados por cada classe os valores das percentagens de bloqueio são superiores aos dos gráficos anteriores.

No gráfico 6-29 podemos observar as probabilidades de bloqueios globais de circuitos (equivalente de chamadas monocanal) na rede, com o DAR, para diferentes valores da reserva de circuitos fixa, para vários valores do factor de sobrecarga.

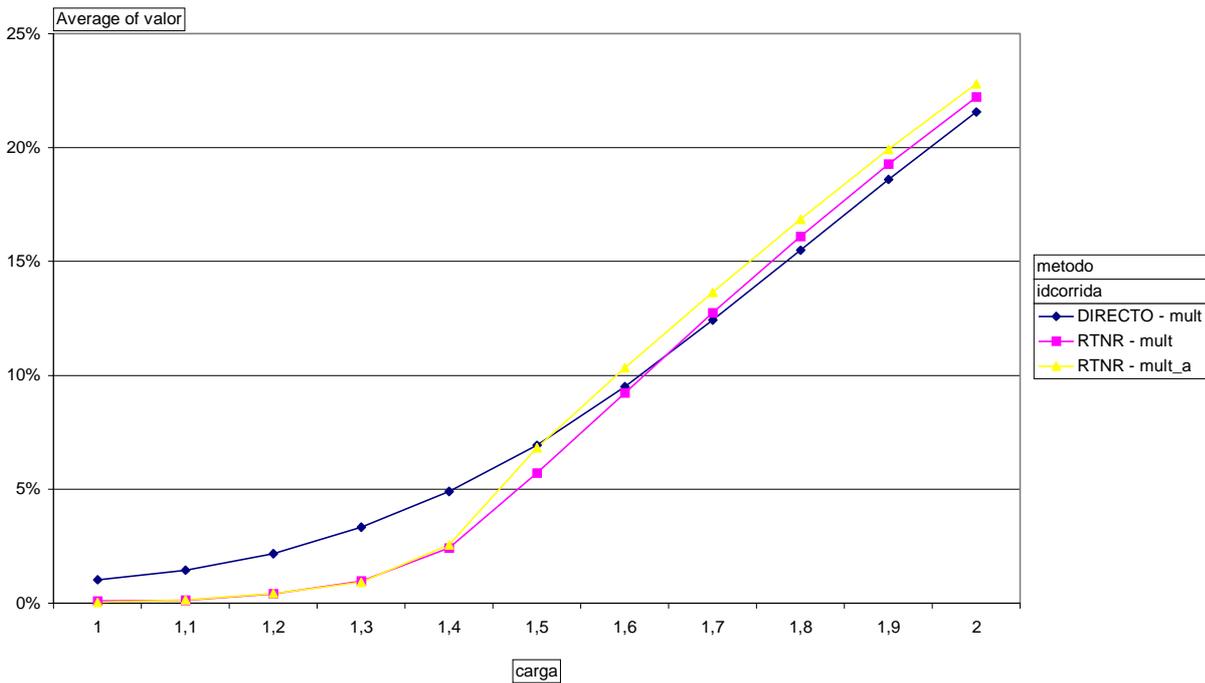


Gráf. 6-29 – Probabilidade de bloqueio global de circuitos na rede, com o DAR, para diferentes valores da reserva de circuitos fixa (sistema 3).

Ao comparar o gráfico 6-29 com o gráfico 6-25 a diferença mais significativa é que a probabilidade de bloqueio global de circuitos na rede é quase a mesma quer se utilize valores diferentes de 0 (mult\_dar\_res) ou iguais a zero (mult\_dar\_res\_b) para a reserva de circuitos fixa para as classes com maior necessidade de recursos, o que se confirma pela sobreposição quase completa dos intervalos de confiança associados a estes valores para factores de sobrecarga superiores a 1,4, visíveis na tabela G-27 do apêndice G. Uma outra diferença é que a diferença entre as probabilidades de bloqueios globais de circuitos na rede entre o DAR e o DIRECTO é muito inferior à diferença entre as probabilidades verificada no gráfico 6-25 o que é particularmente visível para sobrecargas elevadas (>1,7).

As probabilidades de bloqueio global de circuitos (equivalente de chamadas monocanal) na rede, com o RTNR, para duas diferentes escolhas dos valores da largura de banda mínima garantida e para vários valores do factor de sobrecarga são apresentadas no gráfico 6-30.

item | Tpb | Circ | fluxos | 4 | reserva | 0



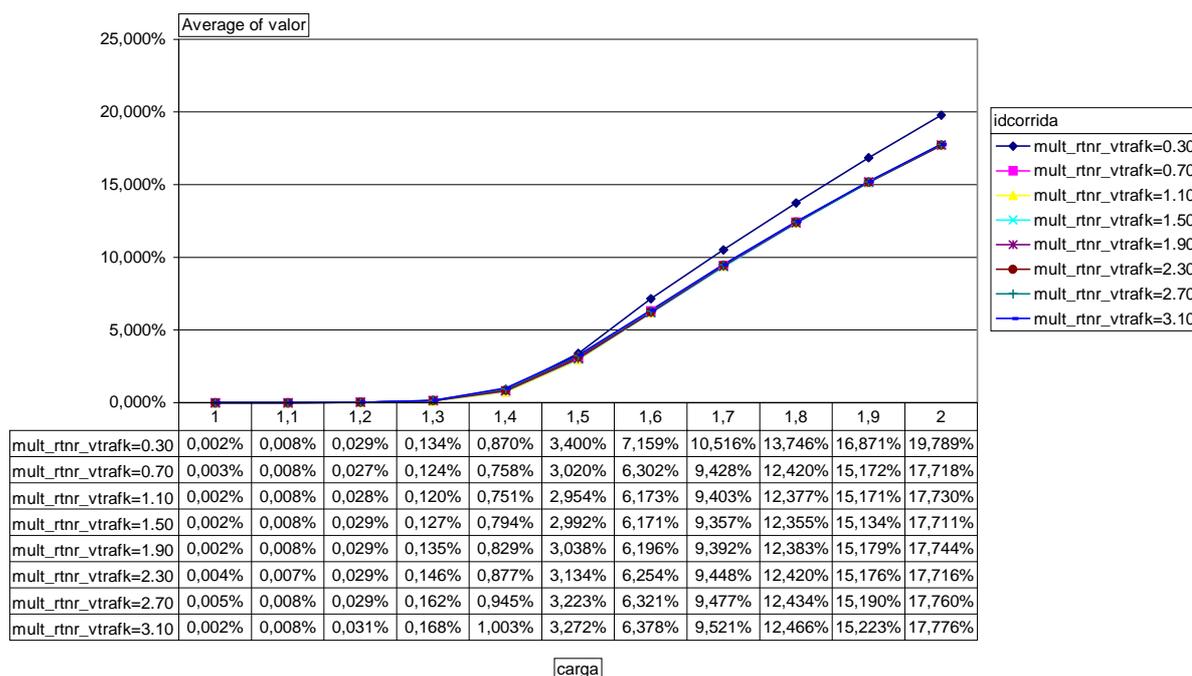
Gráf. 6-30 – Probabilidade de bloqueio global de circuitos na rede, com o RTNR, para duas diferentes escolhas dos valores da largura de banda mínima garantida (sistema 3).

Tudo o que foi dito para o gráfico 6-26 para a probabilidade de bloqueio global de chamadas continua a aplicar-se no gráfico 6-30 para a probabilidade de bloqueio global de circuitos na rede, sendo a única alteração visível a maior proximidade dos bloqueios, no gráfico 6-30, quando a carga aumenta. Também neste caso a tabela G-28 do apêndice G indica que existe uma sobreposição significativa dos intervalos de confiança para os dois diferentes casos do método RTNR para cargas inferiores a 1,4.

## 6.5.2 Estudo da variação do valor do parâmetro $Factor1$ no RTNR

O desempenho deste sistema com o método de encaminhamento RTNR deteriora-se demasiado quando o factor de sobrecarga ultrapassa o valor 1,5 , tornando-se mesmo pior que o encaminhamento DIRECTO. Por esta razão examinámos o desempenho da rede sob a variação do parâmetro  $Factor1$  na expressão 6-1 (factor do  $VTraf_k$ ) do método, como já fizemos também com o sistema 2. Os resultados dessa análise, com o sistema 3, encontram-se no gráfico 6-31.

metodo|RTNR|item|Tpb|fluxos|4



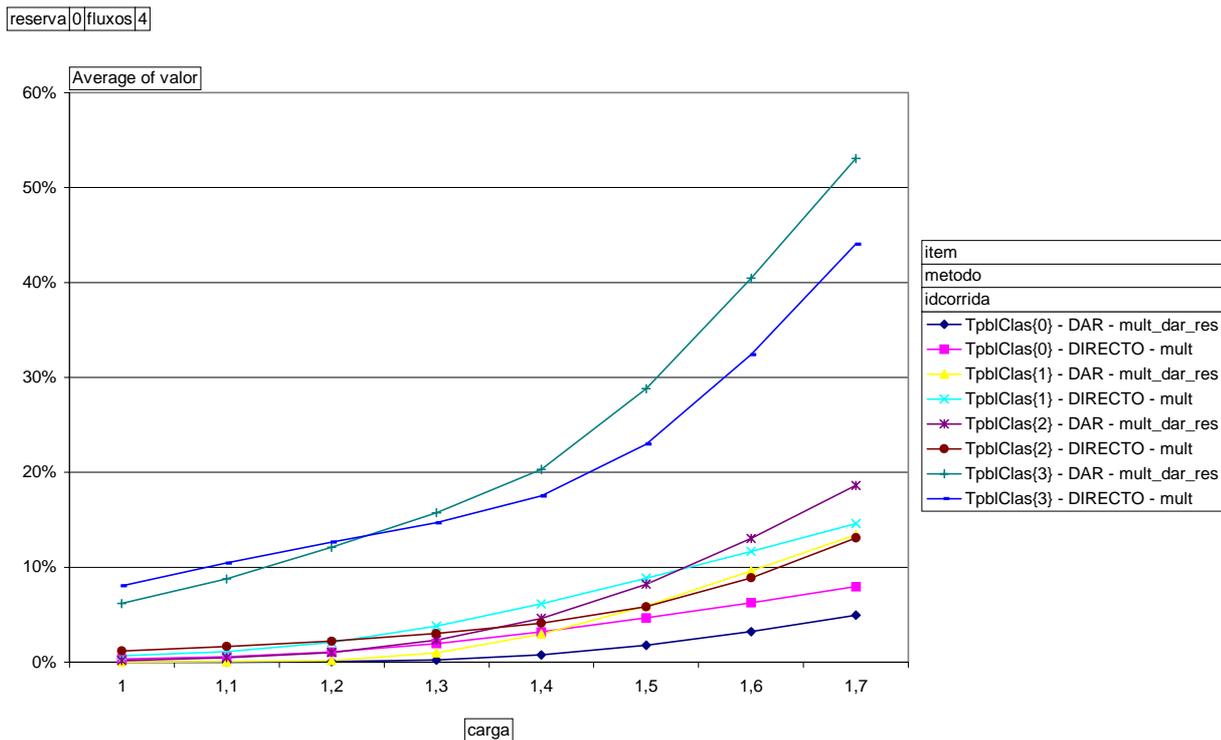
Gráf. 6-31 – Probabilidade de bloqueio global no RTNR versus o factor do  $VTtrafk_k$  (sistema 3).

Estes resultados indicam que o factor do  $VTtrafk_k$  não influencia demasiado o desempenho da rede. Os resultados do gráfico 6-31 indicam que um factor de 1,1 não é uma escolha pior que qualquer das restantes para o sistema 3, conforme é visível pela sobreposição em geral dos intervalos de confiança na tabela G-29 do apêndice G.

### 6.5.3 Análise da probabilidade de bloqueio global por classe

Vamos analisar agora os valores das probabilidades de bloqueio global na rede para as chamadas de cada uma das classes.

O gráfico 6-32 ilustra a congestão de chamadas na rede para as quatro classes de chamadas e para vários factores de sobrecarga para os métodos de encaminhamento DAR e DIRECTO.

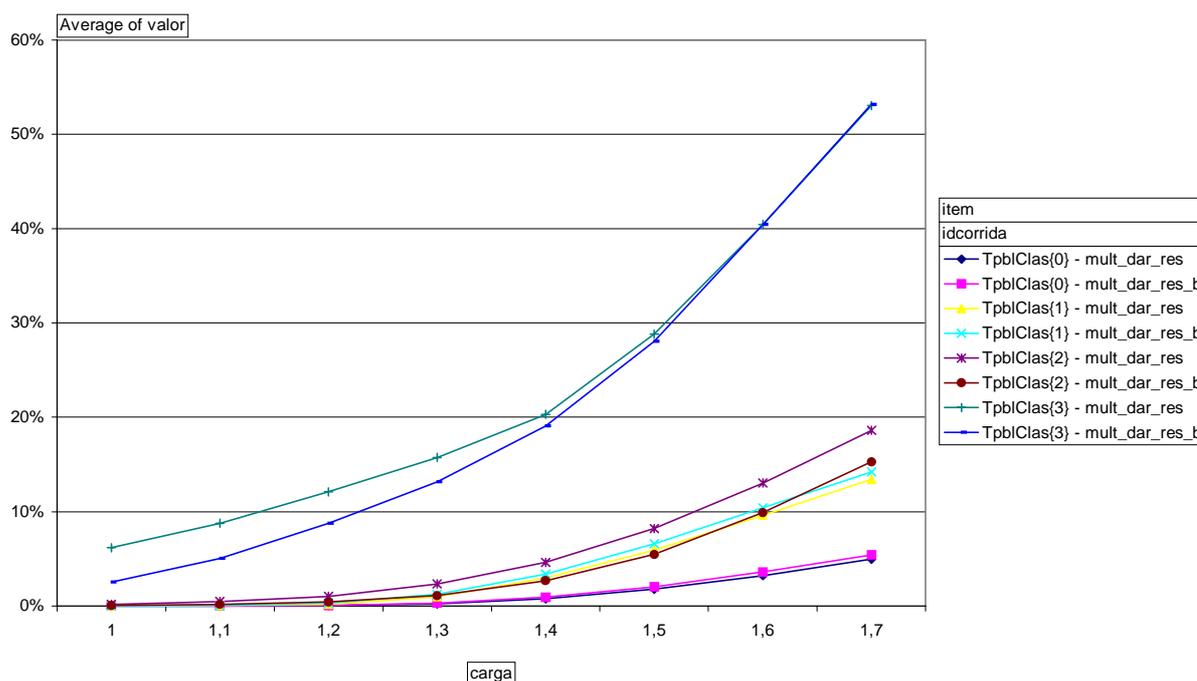


Gráf. 6-32 – Probabilidade de bloqueio global por classe para o método DAR e DIRECTO (sistema 3).

Pela observação do gráfico 6-32 podemos afirmar que, para factores de sobrecarga não muito baixos (>1,3), o comportamento do DAR é superior ao DIRECTO para as duas classes com menor necessidade de recursos. Para as outras duas classes o comportamento é o inverso, ou seja, o comportamento do DAR é inferior ao DIRECTO. No DAR quando a rede não está sobrecarregada é encaminhado tráfego de todas as classes (mais eficientemente para todas que no DIRECTO, como é visível na tabela G-30 do apêndice G) mas quando a rede começa a ficar sobrecarregada o seu comportamento tende a limitar mais o encaminhamento das classes com maior necessidade de largura de banda e isto faz com que a congestão de chamadas na rede com o DAR seja inferior à congestão com o DIRECTO, como se observou no gráfico 6-25.

O gráfico 6-33 ilustra também a congestão de chamadas na rede para as quatro classes de chamadas e para vários factores de sobrecarga mas agora apenas para o método de encaminhamento DAR, com duas diferentes escolhas para a reserva de circuitos fixa.

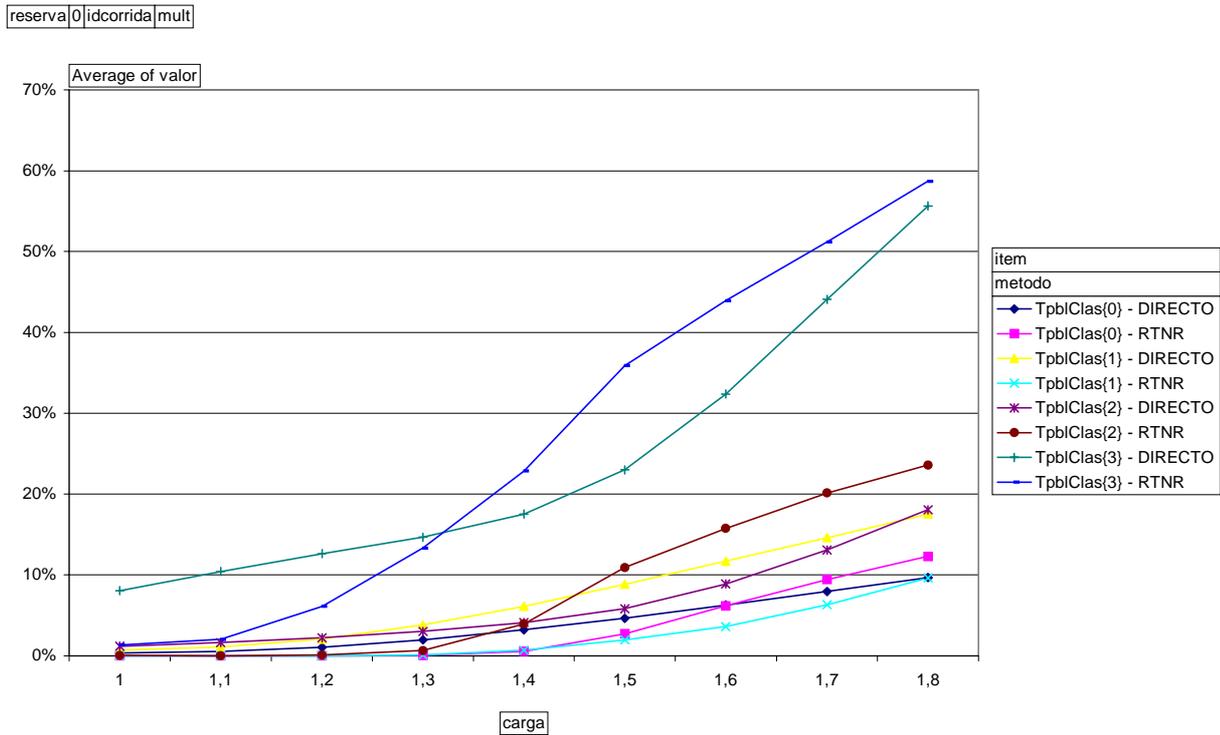
metodo|DAR|reserva|0|fluxos|4



Gráf. 6-33 – Probabilidade de bloqueio global por classe para o método DAR, para duas diferentes escolhas dos valores da reserva de circuitos fixa (sistema 3).

Quando no DAR é permitida uma maior liberdade de encaminhamento das classes com maior necessidade de recursos (pela anulação do valor da reserva fixa – `mult_dar_res_b`) a congestão de chamadas para essas classes diminui mas em contrapartida aumenta a congestão de chamadas para as classes com menores necessidades, como pode ser observado no gráfico 6-33. No entanto deve ser dito que para sobrecargas muito elevadas (maiores que 1,6) o bloqueio da classe 3 eleva-se devido à escassez generalizada de recursos, em ambos os casos. O não favorecimento das classes com necessidades mais elevadas leva a que a congestão de chamadas na rede com o DAR para `idcorrida=mult_dar_res` seja inferior à congestão para `idcorrida=mult_dar_res_b`, como se observou no gráfico 6-25 (nota-se no entanto que esta diferença é muito menos visível no gráfico 6-29). A justificação para isso resulta de que, ao dar-se uma maior liberdade de encaminhamento às classes com maior necessidade de recursos permitindo mais encaminhamento alternativo nessas classes (`mult_dar_res_b`), há um maior consumo de recursos, o que vai impedir o encaminhamento de muitas chamadas das classes com menores larguras de banda, aumentando por conseguinte o seu bloqueio. Por outro lado limitando o encaminhamento alternativo nas classes com maior necessidade de recursos (`mult_dar_res`), pela introdução de uma reserva de circuitos fixa diferente de 0, vai haver um número maior de recursos disponíveis para as classes com menor necessidade de recursos o que faz com que o bloqueio para essas classes seja mais baixo. Os dados que fundamentam esta análise encontram-se na tabela G-31 do apêndice G.

Vamos agora ilustrar, no gráfico 6-34, a congestão de chamadas na rede para as quatro classes de chamadas e para vários factores de sobrecarga para os métodos de encaminhamento RTNR e DIRECTO.

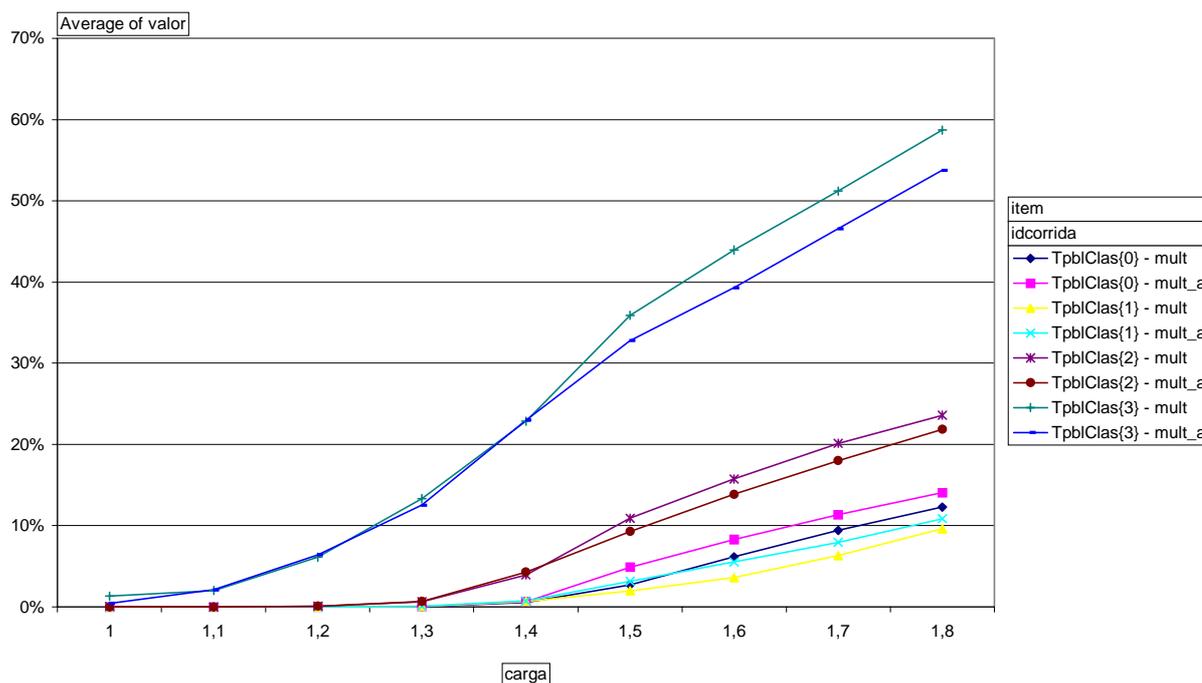


Gráf. 6-34 – Probabilidade de bloqueio global por classe para o método RTNR e DIRECTO (sistema 3).

No gráfico 6-34 vemos que para todas as classes a congestão de chamadas global por classe com o RTNR é superior à congestão com o DIRECTO, para factores de sobrecarga elevados (>1,6), com excepção da classe 1. Supõe-se por esta razão que para esses factores de sobrecarga (possivelmente devido ao mecanismo de protecção das classes) o RTNR utilize encaminhamento alternativo a proteger as classes com maior necessidade de recursos o que leva a um aumento de bloqueio em todas as classes. Esse gráfico parece ainda indicar que este mecanismo leva a existir uma protecção excessiva da classe 1 no RTNR para cargas elevadas. A análise dos intervalos de confiança da tabela G-32 do apêndice G permite ainda determinar que para sobrecargas reduzidas (<1,4) o bloqueio do RTNR é mais baixo que o do DIRECTO para todas as classes de tráfego.

Vamos agora ilustrar, no gráfico 6-35, ainda a congestão de chamadas na rede para as quatro classes de chamadas e para vários factores de sobrecarga, mas apenas para o método de encaminhamento RTNR, para duas escolhas dos valores da largura de banda mínima garantida, por classe e por par de centrais. A corrida que corresponde a maiores valores da largura de banda mínima garantida, para as classes com maior necessidade de recursos em cada um dos pares de centrais, é a que tem o campo idcorrida com o valor mult\_a, como já foi referido.

metodo|RTNR|reserva|0|fluxos|4

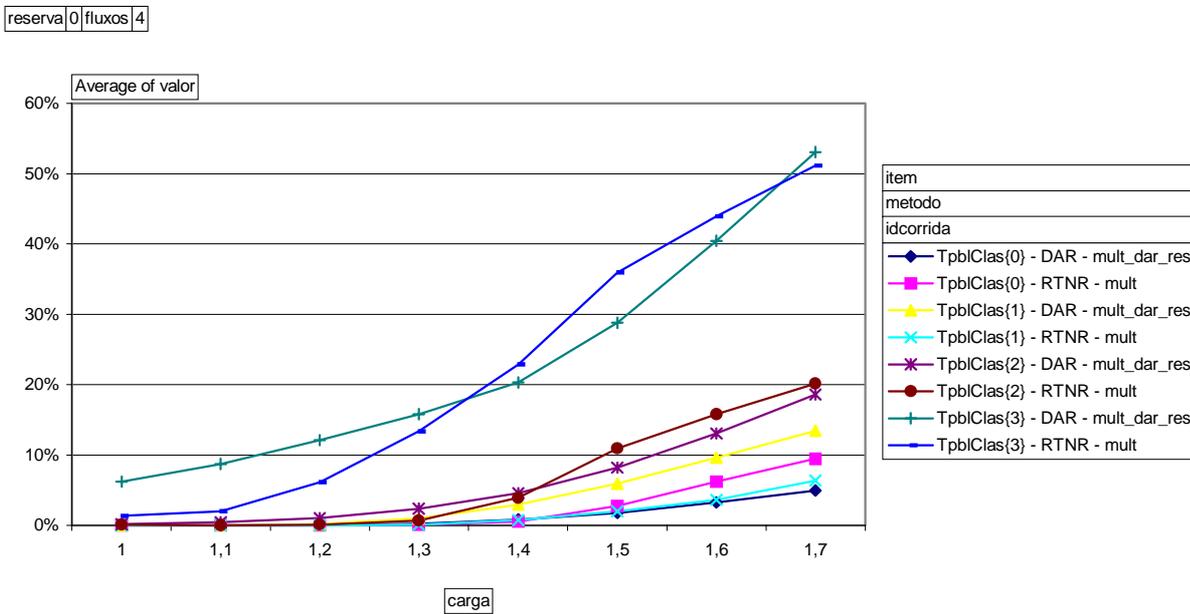


Gráf. 6-35 – Probabilidade de bloqueio global por classe para o RTNR, para duas diferentes escolhas dos valores da largura de banda mínima garantida (sistema 3).

Para cargas inferiores a 1,5 os intervalos de confiança associados aos bloqueios de cada classe são sobrepostos na sua quase totalidade, pelo que, de acordo com um grau de confiança de 95% não podemos afirmar que algum destes casos seja melhor (ou sequer distinto) do outro, como se pode ver na tabela G-33 do apêndice G. Pode ver-se no gráfico 6-35 que para factores de sobrecarga superiores a 1,4 a congestão de chamadas para as classes com maior necessidade de recursos (classe 2 e 3) é inferior na simulação que usou maiores valores da largura de banda mínima garantida para essas classes, mas em contrapartida as outras duas classes (classe 0 e 1) nessa simulação têm congestão de chamadas superiores à obtida na simulação que usou maiores valores da largura de banda mínima garantida para estas classes. Concluimos que ao estarmos a dar uma maior liberdade de encaminhamento às classes com maior necessidade de recursos há um maior consumo de recursos por chamada, o que vai impedir o encaminhamento de muitas chamadas das classes com menores necessidades, aumentando por conseguinte o bloqueio destas. Por outro lado limitando o encaminhamento nas classes com maior necessidade de recursos fica um número maior de recursos disponíveis para as classes com menor necessidade de recursos o que faz com que o bloqueio para essas classes seja mais baixo. Conseguiu-se no entanto o que se pretendia neste teste, que era diminuir o valor dos bloqueios nas classes com maior necessidade de recursos, embora por outro lado, tenha aumentado o valor da congestão global de chamadas na rede como foi observado nos gráficos 6-26 e 6-30. Como nesse caso (gráfico 6-26) as diferenças entre a congestão total de chamadas na rede é grande, tal pode indicar ser mais eficiente

limitar as classes mais altas, pois as diferenças dos bloqueios globais indicam que se está a fazer um uso muito ineficiente dos recursos globais.

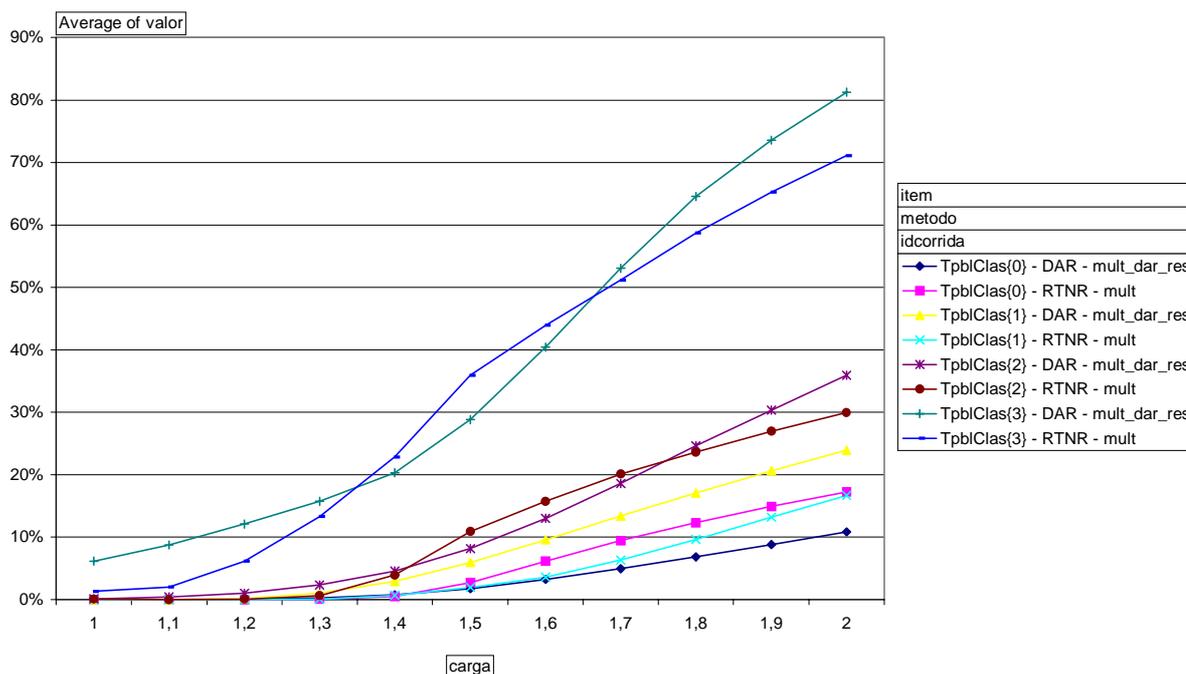
Nos gráficos 6-36 e 6-37 é também apresentada a congestão de chamadas na rede para as quatro classes de chamadas e para vários factores de sobrecarga para os métodos de encaminhamento RTNR e DAR.



Gráf. 6-36 – Probabilidade de bloqueio global por classe para o método DAR e RTNR (sistema 3) – excerto.

Pela observação dos gráficos 6-36 e 6-37 verificamos que a congestão de chamadas nas classes com maior necessidade de recursos com o RTNR tem variações mais bruscas para factores de sobrecarga à volta de 1,4 e que para factores de sobrecarga baixos e altos as variações são mais suaves. Por outro lado com o DAR as variações mais bruscas ocorrem para factores de sobrecarga superiores 1,4. Por análise dos intervalos de confiança na tabela G-34 do apêndice G, é possível verificar que o RTNR comporta-se melhor que o DAR para todas as classes em factores de sobrecarga inferiores a 1,4. Contudo, para outros valores de sobrecarga, nomeadamente na gama média-alta, tal como se mostra na tabela 6-9 o DAR tem melhor desempenho que o RTNR para quase todas as classes de chamadas.

reserva0 fluxos4



Gráf. 6-37 – Probabilidade de bloqueio global por classe para o método DAR e RTNR (sistema 3).

Para facilitar a comparação por classe de chamada do comportamento relativo DAR (mult\_dar\_res) e RTNR para diferentes factores de sobrecarga,  $\rho$ , e tal como foi já feito para o sistema 2, a tabela 6-9 apresenta uma análise simplificada do comportamento destes dois métodos, indicando para que factores de sobrecarga um método possui um melhor desempenho (+), ou tem um pior desempenho (-). Existe sobreposição dos intervalos apenas para a classe 2 na carga nominal, conforme visível na tabela G-34 do apêndice G. Note-se que tal como anteriormente, os valores dos pontos de inflexão são estimados aproximadamente através de interpolação linear dos valores obtidos.

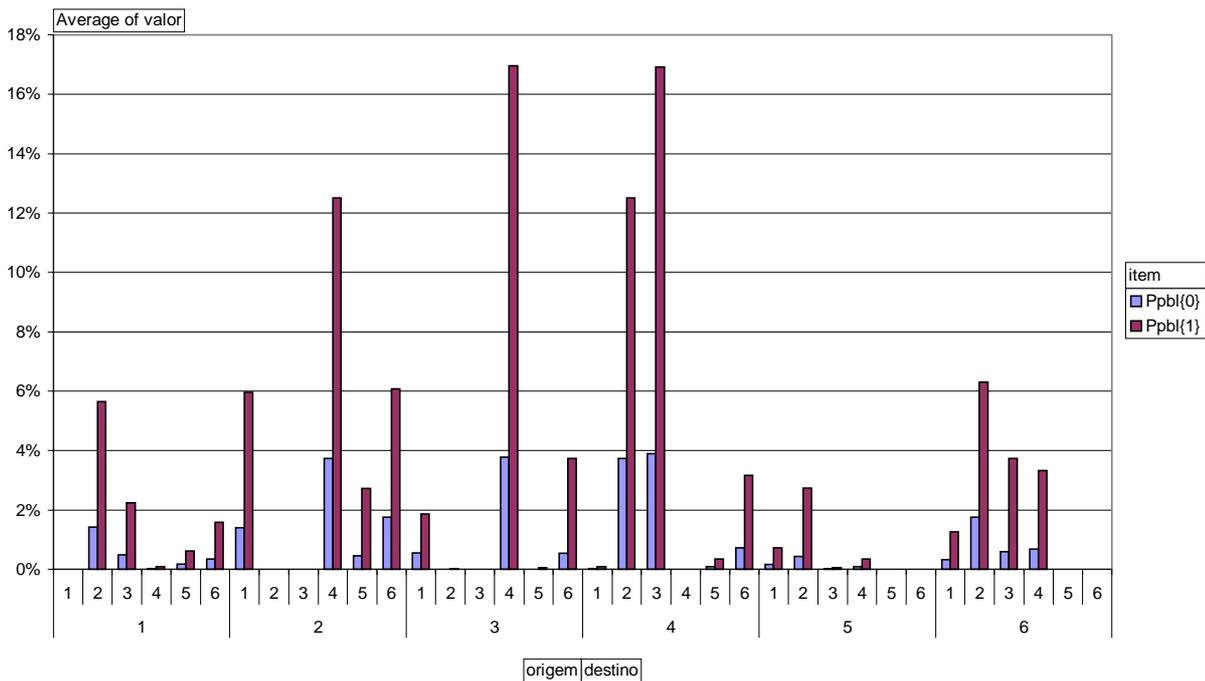
Classe	$\rho$ baixo e médio-baixo		Inflexão	$\rho$ médio-alto		Inflexão	$\rho$ alto	
	DAR	RTNR		DAR	RTNR		DAR	RTNR
0	-	+	1,45	+	-		+	-
1	-	+		-	+		-	+
2	-	+	1,45	+	-	1,75	-	+
3	-	+	1,35	+	-	1,65	-	+

Tab. 6-9 – Probabilidade de bloqueio global por classe para o método DAR e RTNR (sistema 3).

## 6.5.4 Análise da probabilidade de bloqueio ponto a ponto por classe

Neste ponto vamos analisar os valores das probabilidades de bloqueio ponto a ponto para as chamadas de várias classes. O gráfico 6-38 ilustra esses valores para o método de encaminhamento DAR, para um factor de sobrecarga igual a 1,4, mas apenas para as classes 0 e 1, uma vez que são as que geram tráfego de todas as origens para todos os destinos.

metodo|DAR|reserva|0|carga|1,4|idcorrida|mult\_dar\_res

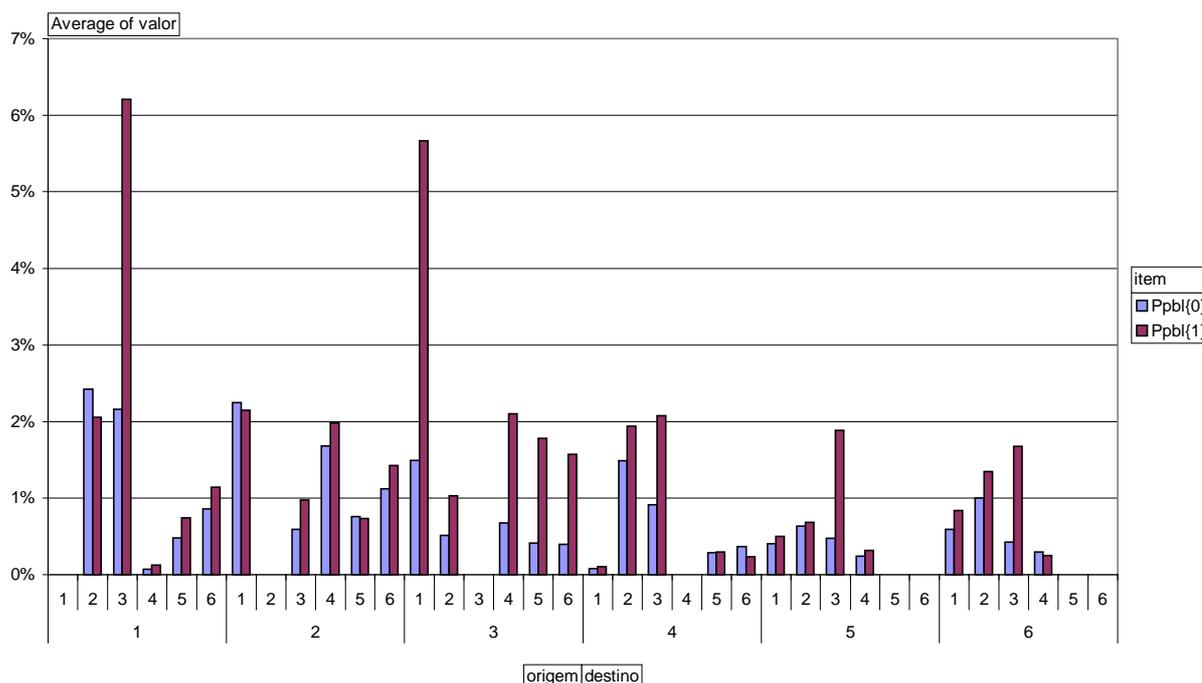


Gráf. 6-38 – Probabilidade de bloqueio ponto a ponto por classe, para as classes 0 e 1, para o método DAR (sistema 3).

Pode ser observado no gráfico 6-38 que a probabilidade de bloqueio ponto a ponto é sempre superior para a classe com maior necessidade de recursos, como pode também ser verificado na tabela G-35 do apêndice G. Esta observação é em geral válida para qualquer outro factor de sobrecarga.

O gráfico 6-39 ilustra as probabilidades de bloqueio ponto a ponto para as chamadas de duas das classes com o método de encaminhamento RTNR, para um factor de sobrecarga igual a 1,4. Também aqui apenas são apresentadas as classes 0 e 1, pela mesma razão que no caso anterior.

metodo|RTNR|reserva|0|carga|1,4|idcorrida|mult

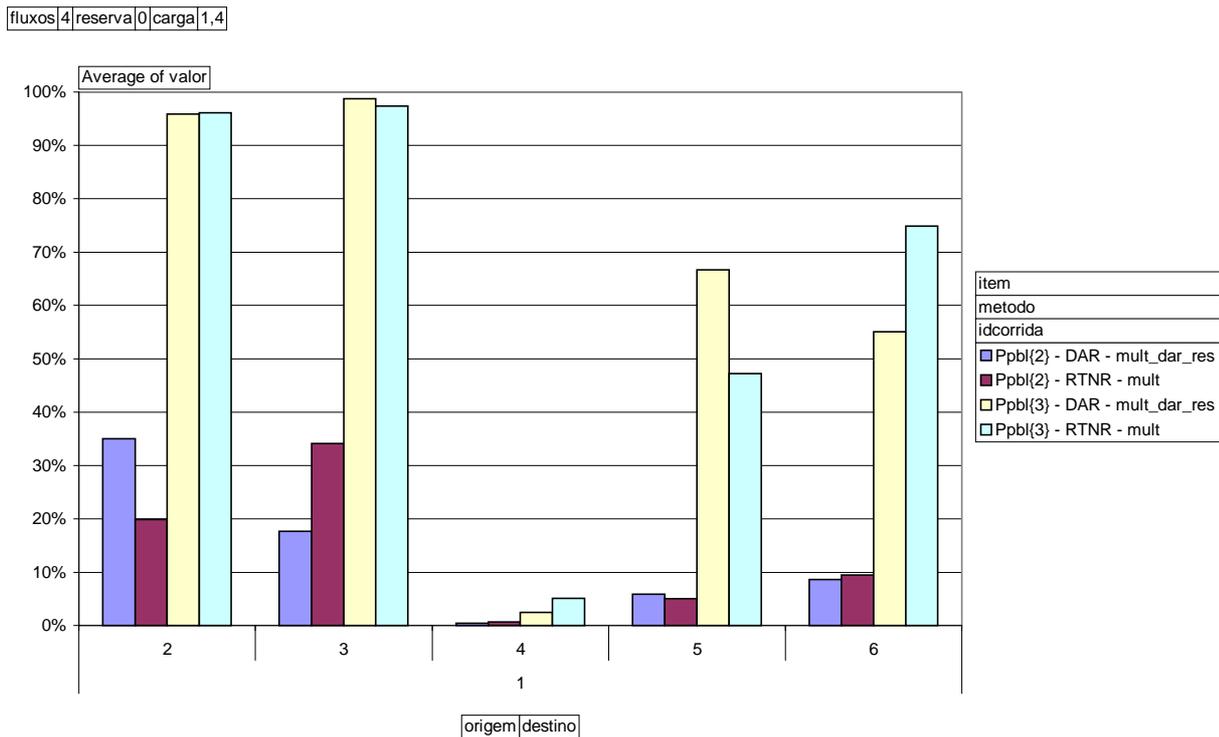


Gráf. 6-39 – Probabilidade de bloqueio ponto a ponto por classe, para as classes 0 e 1 para o método RTNR (sistema 3).

Pode ser observado no gráfico 6-39 que a probabilidade de bloqueio ponto a ponto para uma classe não é sempre superior à da outra classe (o que acontecia no DAR), conforme pode ser verificado na tabela G-36 do apêndice G.

Embora não apresentado graficamente, foi ainda observado que existem factores de sobrecarga que apresentam maioria de bloqueios ponto a ponto superiores numa qualquer das classes e outros que apresentam uma distribuição equilibrada dessas maiorias. Para sobrecargas baixas à classe 0 correspondem normalmente bloqueios ponto a ponto inferiores aos da classe 1, enquanto para sobrecargas elevadas à classe 0 correspondem frequentemente bloqueios ponto a ponto mais elevados que à classe 1.

Vamos agora analisar o comportamento ponto a ponto das classes 2 e 3. O gráfico 6-40 ilustra esses valores para os método de encaminhamento DAR e RTNR, para um factor de sobrecarga igual a 1,4. Para estas classes apenas tem sentido investigar o tráfego ponto a ponto da origem 1 para todos os destinos, uma vez que não existe geração de tráfego deste tipo nas outras centrais.



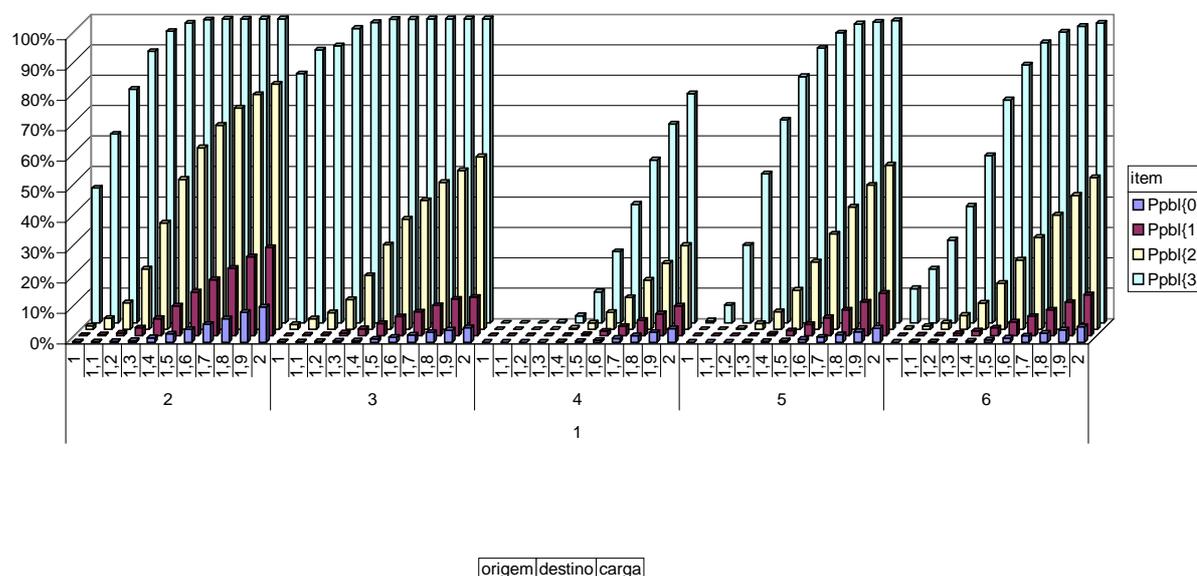
Gráf. 6-40 – Probabilidade de bloqueio ponto a ponto por classe, para as classes 2 e 3, para o método DAR e RTNR (sistema 3).

Neste gráfico, e de acordo com os dados apresentados na tabela G-37 do apêndice G é possível verificar que a classe 3 apresenta sempre um bloqueio ponto a ponto significativamente superior à classe 2, qualquer que seja o método de encaminhamento usado. Não é possível determinar qual dos métodos apresenta um comportamento ponto a ponto globalmente melhor, uma vez que este comportamento é bastante diferente para diferentes pares origem-destino. Observando o comportamento do sistema para diversos factores de sobrecarga (dos quais não se considerou necessário apresentar gráficos) verificamos que este tipo de comportamento se mantém.

Finalmente, apresentamos no gráfico 6-41 a probabilidade de bloqueio ponto a ponto da origem 1 para todos os destinos, para o método DAR para os diferentes factores de sobrecarga.

metodo|DAR|idcorrida|mult\_dar\_res|fluxos|4|reserva|0

Average of valor



Gráf. 6-41 – Probabilidade de bloqueio ponto a ponto por classe, para o método DAR para várias cargas (sistema 3).

Neste gráfico podemos verificar que as cargas crescentes implicam bloqueios ponto a ponto também crescentes para todas as classes, como seria de esperar. Para além disto, é também visível que para cada uma das cargas, o bloqueio ponto a ponto é crescente com o número de circuitos necessitados pela classe (esta afirmação apenas não pode ser feita com um grau de confiança de 95% quando os valores de bloqueio ponto a ponto são quase nulos para as diferentes classes, como é visível nas tabelas G-38, G-39 e G-40 do apêndice G).

Uma análise adicional deste sistema relativa ao comportamento do bloqueio nos feixes para os métodos DAR e RTNR pode ser encontrada no apêndice H, secção H.2.

## 6.6 Resultados mais relevantes

As seguintes tabelas apresentam o comportamento global de cada um dos sistemas estudados, apresentando para cada caso qual o método que apresenta melhor desempenho. Em cada um destes casos os pontos de inflexão foram obtidos aproximadamente por interpolação linear. O nome do método refere-se sempre à sua implementação de acordo com a definição original, uma vez que as variações não levaram em geral a atingir o melhor desempenho (mesmo que em casos pontuais tenham melhorado a prestação do método). Note-se que desde que não exista sobreposição dos intervalos de confiança não se

distingue se um método é o melhor com uma grande ou com uma pequena vantagem em relação aos restantes. Se existir sobreposição (mesmo que parcial) dos intervalos de confiança, não se pode afirmar com clareza se um método é melhor em relação aos restantes. Quando se refere o método DAR, assume-se o caso com reserva fixa diferente de zero para todas as classes existentes (caso `mult_dar_res`).

<i>Medida de Desempenho</i>	$\rho$ baixo	<i>Inflexão</i>	$\rho$ médio	<i>Inflexão</i>	$\rho$ alto
Bloqueio Global Chamadas	RTNR	-	RTNR	-	RTNR

Tab. 6-10 – Análise do Comportamento Global – Método com melhor desempenho (Sistema 1).

Na tabela 6-10 verifica-se que para o sistema 1, o método RTNR é globalmente sempre algo mais vantajoso. Note-se no entanto que existe uma sobreposição entre os métodos RTNR e DCR para o factor de sobrecarga 2, e que para todas as cargas médias e elevadas a diferença dos bloqueios entre estes dois métodos, e ainda o DAR, é relativamente pequena (e esta diferença diminui com o aumento da carga).

<i>Medida de Desempenho</i>	$\rho$ baixo	<i>Inflexão</i>	$\rho$ médio - alto
Bloqueio Global Chamadas	RTNR	1,42	DAR
Bloqueio Global Circuitos	RTNR	1,46	DAR
Bloqueio Global Classe 0	RTNR	1,42	DAR
Bloqueio Global Classe 1	RTNR	-	RTNR

Tab. 6-11 – Análise do comportamento global – Método com melhor desempenho (Sistema 2).

Na tabela 6-11 verifica-se que para o sistema 2, aparenta existir uma vantagem do RTNR para cargas baixas e médias (até um factor de sobrecarga de cerca 1,45) e a partir daí a vantagem pertence ao método DAR. Note-se no entanto que uma observação dos dados permite concluir que se esta vantagem é realmente forte em termos de bloqueio de chamadas, em termos de bloqueio de circuitos existe uma quase sobreposição entre estes dois métodos para cargas elevadas, existindo uma verdadeira sobreposição para o factor de carga 2. Deve ser notado que o RTNR permite um melhor desempenho para todos os factores de carga para a classe 1.

<i>Medida de Desempenho</i>	$\rho$ baixo	<i>Inflexão</i>	$\rho$ médio	<i>Inflexão</i>	$\rho$ alto
Bloqueio Global Chamadas	RTNR	1,46	DAR	-	DAR
Bloqueio Global Circuitos	RTNR	1,46	DAR	-	DAR
Bloqueio Global Classe 0	RTNR	1,45	DAR	-	DAR
Bloqueio Global Classe 1	RTNR	-	RTNR	-	RTNR
Bloqueio Global Classe 2	RTNR	1,41	DIRECTO	-	DIRECTO
Bloqueio Global Classe 3	RTNR	1,32	DIRECTO	1,88	RTNR

Tab. 6-12 – Análise do Comportamento Global – Método com melhor desempenho (Sistema 3).

Na tabela 6-12 verifica-se que para o sistema 3, tal como para o sistema 2, aparenta existir uma vantagem do RTNR para cargas baixas e médias (até um factor de sobrecarga de cerca 1,45) e a partir daí a vantagem pertence ao método DAR. No entanto, e tal como referido anteriormente esta diferença é mais visível em bloqueio de chamadas do que em bloqueio de circuitos, só que neste caso o método que se aproxima do DAR para cargas elevadas é o método DIRECTO (que se chega a sobrepor a este para

um factor de carga 2) o que parece indicar um muito elevado grau de eficácia do DAR, com reserva de circuitos, nesta zona de muito forte sobrecarga. Deve ser notado que os diferentes métodos apresentam comportamentos em relação a cada classe bastante diferentes entre si. Deve ser no entanto referido que o método RTNR aparece como o melhor para a classe 1 em qualquer carga.

No geral, em condições de sobrecarga extremamente elevada os métodos que usam encaminhamento alternativo mostraram-se apenas ligeiramente superiores ao método DIRECTO, com algumas variações desses métodos por vezes mesmo piores que o DIRECTO (o que aponta que esses factores de carga correspondem ao nível de rotura da eficácia dos encaminhamentos alternativos). No entanto, para cargas baixas os métodos dinâmicos são muitíssimo melhores que o encaminhamento DIRECTO.

Constata-se, como indicado na literatura, que os mecanismos de reserva de circuitos fixa (controlo de protecção do tráfego de primeiro encaminhamento) utilizada ao nível do método DAR (e de outros, como o DCR) parecem ter efeitos bastante relevantes em relação ao bloqueio global; os nossos testes verificam que estes resultados podem ser estendidos a condições de tráfego multiclasse, que são de extrema relevância em redes com integração de serviços (sistemas 2 e 3).

Por outro lado, a aplicação ao nível do método DAR e DCR de mecanismos de reserva dinâmica (controlo de protecção de tráfego de primeiro encaminhamento) indicados na secção 5.5.4.1 (simplificação dos usados no RTNR) parecem levar no sistema monoclasse a uma melhoria de desempenho em relação aos métodos originais. Esta melhoria no entanto não se verifica uniformemente para os sistemas multiclasse. De facto o comportamento do método DAR com reserva dinâmica (estilo RTNR) leva a um desempenho melhor que a reserva fixa não nula para sobrecargas baixas (inferiores a 1,4), embora conduza a desempenhos piores para sobrecargas muito elevadas. No entanto para as cargas baixas, o método DAR com reserva fixa nula é melhor que ambos. O comportamento por classe aproxima-se (mas é em geral pior) ao comportamento por classe do RTNR, mais do que ao comportamento por classe das restantes variações do DAR. Em conclusão, o mecanismo de reserva dinâmica implementado no DAR leva a um comportamento em termos de bloqueio global e por classe mais próximo do método RTNR.

Como referido, o método DAR, quando é usado o mecanismo de reserva dinâmica adaptado do RTNR apresenta um comportamento no bloqueio global por classes semelhante ao do próprio RTNR. No entanto este comportamento leva em geral a desempenhos em termos de bloqueios globais inferiores às restantes variações do método DAR e a todas as do RTNR. Note-se que o comportamento referido pode também ser devido aos mecanismos de reserva de circuitos para protecção de serviços que fazem parte do mecanismo de reserva dinâmica do RTNR e que foram também implementados na simplificação deste usada no método DAR. Em geral, os mecanismos de protecção de serviços aparentemente funcionaram de forma excessiva quando o tráfego é de maior intensidade, protegendo as classes com

maiores necessidades de recursos. Porém, ao agirem desta forma, a probabilidade de bloqueio global na rede tende a aumentar (o que é mais visível em termos de bloqueio de chamadas do que em termos de bloqueio de circuitos).

A implementação da simplificação do mecanismo de reserva de circuitos de protecção do tráfego de primeiro encaminhamento do RTNR no método RTNR levou nos diferentes sistemas a resultados que se sobrepõem quase completamente ao do método RTNR “original”.

# Capítulo 7

## Conclusões e Trabalho Futuro

Neste capítulo pretende-se fazer uma apreciação breve do trabalho realizado, indicando os resultados e principais conclusões que se julgam mais relevantes do mesmo, bem como fazer uma análise breve dos próximos passos que se considerariam merecedores de estudo a partir dos resultados deste trabalho.

### 7.1 Trabalho realizado

A recolha e estudo dos métodos de encaminhamento alternativo dinâmico permitiu escolher um conjunto de métodos (nomeadamente DAR, DCR, FAR, RTNR e DIRECTO) que foram objecto de revisão e análise aprofundada (no capítulo 3) para posterior implementação. Para além destes foram também objectos de estudo outros métodos de encaminhamento alternativo dinâmico (e estático) que se apresentam também no mesmo capítulo.

A implementação do simulador foi efectuada usando a linguagem C++ e uma ferramenta de construção de simuladores (OMNet++). Foi efectuada uma análise detalhada do sistema a implementar (descrita no capítulo 5) e escolhidas as características do mesmo, de forma a suportar diversos tipos de encaminhamento e diversas classes de tráfego. Após esta análise, o sistema foi implementado de forma a corresponder aos requisitos apresentados.

Os módulos que implementam cada um dos métodos referidos foram implementados dentro do enquadramento do simulador, com capacidades multiclasse (permitindo vários tipos de tráfego com ocupações diferenciadas de recursos por chamada). Alguns dos métodos implementados foram estendidos de forma a suportar estas capacidades (e o estudo de alguns parâmetros usualmente considerados “fixos” para o método). Finalmente, foram também implementadas as capacidades adicionais para alguns métodos descritas no capítulo 5, principalmente para tentar melhorar o seu comportamento em situações de tráfego multiclasse.

Finalmente, fez-se no capítulo 6, um estudo comparativo dos resultados obtidos para os diferentes métodos implementados em situações de tráfego monoclasse e multiclasse, para o tráfego nominal e diversos factores de sobrecarga em redes dimensionadas de acordo com as aproximações apresentadas no capítulo 4.

### **7.1.1 Resultados fundamentais do estudo simulacional**

De uma forma sintética, pretendemos aqui considerar os principais resultados do estudo simulacional, para os diversos sistemas estudados.

Da análise dos três sistemas investigados, pode-se concluir que o método RTNR se apresenta mais vantajoso no sistema que apenas possui uma classe de tráfego monocanal (sistema 1), para qualquer factor de carga. Nos dois sistemas que possuem várias classes de tráfego multicanal (2 classes, tráfego de Poisson – sistema 2; 4 classes, tráfego de Poisson e de tipo Engset – sistema 3) o método RTNR apenas é mais vantajoso para cargas baixas e média-baixas (factor de sobrecarga inferior a 1,4), enquanto o método DAR com reserva fixa não nula é o mais vantajoso para sobrecargas superiores. No entanto a análise do bloqueio por classes apresenta resultados muito díspares.

No geral, em condições de sobrecarga extremamente elevadas, que só excepcionalmente têm interesse prático (nomeadamente face à ocorrência de avarias ou sobrecargas excepcionalmente elevadas) os métodos que usam encaminhamento alternativo mostraram-se apenas ligeiramente superiores ao encaminhamento directo, com algumas variações desses métodos por vezes mesmo piores que este.

Constata-se que os mecanismos de reserva de circuitos fixa utilizada ao nível de diversos métodos têm efeitos bastante significativos em relação ao bloqueio global em todos os sistemas analisados. A aplicação nestes métodos de mecanismos de reserva dinâmica parecem levar no sistema monoclasse a uma melhoria de desempenho em relação aos métodos originais. Esta melhoria no entanto não se verifica uniformemente para os sistemas multiclasse.

No método DAR, quando é usado o mecanismo de reserva dinâmica adaptado do RTNR apresenta um comportamento no bloqueio global por classes semelhante ao do próprio RTNR, embora leve a um desempenho em termos de bloqueios globais inferior às restantes variações consideradas do método DAR e a todas as do RTNR. Este comportamento pode ser devido aos mecanismos de reserva de circuitos para protecção de serviços que fazem parte do mecanismo de reserva dinâmica do RTNR e que foram também implementados. Em geral, os mecanismos de protecção de serviços funcionaram aparentemente de forma excessiva quando o tráfego é de maior intensidade, protegendo as classes com maiores necessidades de recursos o que tende a levar a que a probabilidade de bloqueio global na rede possa aumentar.

A implementação de uma simplificação do mecanismo de reserva de circuitos de protecção do tráfego de primeiro encaminhamento do RTNR neste método levou nos diferentes sistemas a resultados que se sobrepõe quase completamente ao do método RTNR “original”.

## 7.2 Contribuições desta dissertação

O trabalho descrito anteriormente pretendeu contribuir para a área da análise do grau de serviço em redes inter-centrais, nomeadamente com tráfego multi-classe. De entre as contribuições julgadas mais relevantes desta dissertação, podem salientar-se:

- Construção de um simulador de tráfego numa rede de comutação de circuitos, com capacidade de funcionar com diferentes tipos de redes, sendo que nesta se podem aplicar simultaneamente diferentes métodos de encaminhamento, diferentes classes e diferentes tipos de tráfego. Este simulador permite trabalhar com chamadas de tráfego de população finita de tipo Engsetiano (para além do tráfego de Poisson) e com chamadas com diferentes requisitos de recursos. Este simulador permite ainda a escolha dos diversos parâmetros associados aos processos estocásticos de geração das chamadas. As características das redes a analisar são completamente parametrizáveis.
- Implementação de um conjunto de métodos de encaminhamento dinâmico frequentemente aplicados na prática em redes de telecomunicações. Estes métodos foram implementados de forma parametrizável, para poder ser analisado o efeito de alterações em algumas das grandezas que se utilizam no seu funcionamento. Alguns destes métodos foram implementados de forma a suportar explicitamente diversas classes de tráfego e diferentes necessidades de recursos para o tráfego. Para além disso, o simulador foi criado de forma extensível, pelo que a introdução de novos métodos ou adaptação dos existentes pode ser feita com relativa facilidade.
- Estudo comparativo do comportamento de redes inter-centrais para vários métodos de encaminhamento dinâmico, incluindo novas variantes de alguns métodos sugeridas pelo orientador. Em alguns destes sistemas fez-se a análise do seu comportamento com diferentes classes de tráfego de população finita e tráfegos de Poisson simultaneamente (o que não é comum nas análises comparativas deste tipo presentes na literatura).
- Um conjunto de resultados relevantes quanto à importância significativa dos mecanismos de protecção de serviço no desempenho dos diferentes métodos, bem como sobre o comportamento relativo dos métodos de encaminhamento dinâmico implementados, para diferentes situações de sobrecarga.

## 7.3 Direcções posteriores de pesquisa

Do trabalho anteriormente apresentado, sugerem-se naturalmente algumas áreas onde trabalho adicional pode ser desenvolvido e contribuir para esclarecer alguns pontos que ficaram por analisar aqui. Assim, poderemos considerar:

- Como aplicação simples do simulador e métodos apresentados, é de considerar a análise de outras redes com diferentes topologias para determinar até que ponto os resultados obtidos são extensíveis para situações diferentes. Ainda dentro deste tópico, é de considerar o estudo do comportamento do DAR multiclasse para vários outros valores de reserva fixa.
- Como extensões naturais aos métodos implementados, podem ser consideradas a aplicação de reserva fixa não nula para o método RTNR (tal como descrito no capítulo 3) e a extensão da capacidade multiclasse explícita a outros métodos como o DCR. Pretende-se também utilizar o simulador construído como plataforma experimental para simular outros métodos de encaminhamento, a serem implementados futuramente, nomeadamente o método MODR (Multiple Objective Dynamic Routing) proposto em [Craveirinha01].
- Como alterações ao funcionamento do simulador, podem ser estudados mecanismos para facilitar a análise do funcionamento conjunto de diferentes métodos de encaminhamento numa mesma central e em centrais distintas. Também é de considerar a possibilidade de alterar o simulador de forma a detectar o estado estacionário e terminar uma vez que os dados correspondentes a esse estado tenham sido recolhidos (em vez de terminar num determinado ponto de tempo simulado).
- Finalmente, deve-se também considerar a forma de utilizar diversos factores (não apenas o bloqueio ponto a ponto) para comparar o desempenho dos vários métodos de encaminhamento numa dada rede. Factores como a rentabilidade dos diversos serviços prestados (a capacidade de tomar essa rentabilidade em consideração), o custo do processamento de controlo necessário (um pouco menos importante hoje devido às acrescidas capacidades tecnológicas no processamento), o tráfego de sinalização, a adaptabilidade em situações de avaria nos feixes ou centrais, e outros podem e devem ser considerados numa avaliação relativa sistemática do comportamento dos diversos métodos de encaminhamento.

# Apêndice A

## Descrição do Modelo usando a Linguagem NED

Foi definida a estrutura do modelo da rede usando a linguagem NED (NEtwork Description). A descrição da estrutura do modelo da rede que foi construído é a seguinte:

```
1 simple central
2   parameters:
3     // parametros por metodo de encaminhamento
4     // parametros para os diversos metodos de encaminhamento
5
6     // modulo DCR
7     dcr_tau : numeric const,
8     dcr_delta : numeric const,
9     dcr_theta : numeric const,
10    dcr_intervalo_atualizacao : numeric const,
11    dcr_vtrafk_factor : numeric const,
12    dcr_rk_factor : numeric const,
13    dcr_tipo_reserva_din : numeric const,
14
15    // modulo DAR
16    dar_intervalo_atualizacao : numeric const,
17    dar_vtrafk_factor : numeric const,
18    dar_rk_factor : numeric const,
19    dar_tipo_reserva_din : numeric const,
20
21    // modulo RTNR
22    rtnr_intervalo_atualizacao : numeric const,
23    rtnr_vtrafk_factor : numeric const,
24    rtnr_rk_factor : numeric const,
25    rtnr_tipo_reserva_din : numeric const,
26
27    // modulo FAR
28    far_fich_input_tabelas_routing : string,
29    far_ncaminhos : numeric const,
30
31
32    // nome dos ficheiros de output
33    // informacao propria de cada run da simulacao
34    fich_output_folha_calculo : string,
35
36    // informacao geral da simulacao, a guardar no ficheiro de saida
37    factor_carga : numeric const,
38    fich_topologia : string,
39
40    // parametros gerais a todos os encaminhamentos
41    duracao_chamada_1 : numeric const,
42    duracao_chamada_2 : numeric const,
43    duracao_chamada_3 : numeric const,
44    duracao_chamada_4 : numeric const,
```

```

45     tipo_routing : string,
46     num_estacoes : numeric const,
47     num_classes : numeric const,
48     inicio_recolha : numeric const,
49     fim_recolha : numeric const;
50   gates:
51     in: das_estacoes[];
52     out: para_estacoes[];
53   endsimple
54
55   simple gerador
56     parameters:
57       duracao_chamada_1 : numeric const,
58       duracao_chamada_2 : numeric const,
59       duracao_chamada_3 : numeric const,
60       duracao_chamada_4 : numeric const,
61       factor_carga : numeric const,
62       num_estacoes : numeric const,
63       num_classes : numeric const,
64       // ficheiros de inicializacao
65       fich_topologia : string;
66     gates:
67       in: da_estacao;
68       out: para_estacao;
69   endsimple
70
71   module rede_inteira
72     parameters:
73       // informacao generica a todos os metodos
74       duracao_chamada_1 : numeric const,
75       duracao_chamada_2 : numeric const,
76       duracao_chamada_3 : numeric const,
77       duracao_chamada_4 : numeric const,
78       tipo_routing : string,
79       factor_carga : numeric const,
80       numero_estacoes : numeric const,
81       numero_classes : numeric const,
82       fich_topologia : string,
83       inicio_recolha : numeric const,
84       fim_recolha : numeric const;
85     submodules:
86       gerador: gerador[numero_estacoes];
87       parameters:
88         num_estacoes = numero_estacoes,
89         num_classes = numero_classes,
90         duracao_chamada_1 = duracao_chamada_1,
91         duracao_chamada_2 = duracao_chamada_2,
92         duracao_chamada_3 = duracao_chamada_3,
93         duracao_chamada_4 = duracao_chamada_4,
94         fich_topologia = fich_topologia,
95         factor_carga = factor_carga;
96       display: "p=10,10,ring,200,200;i=gen";
97     central: central[numero_estacoes];
98     parameters:
99       factor_carga = factor_carga,
100      fich_topologia = fich_topologia,
101      num_estacoes = numero_estacoes,
102      num_classes = numero_classes,
103      inicio_recolha = inicio_recolha,
104      fim_recolha = fim_recolha,
105      duracao_chamada_1 = duracao_chamada_1,
106      duracao_chamada_2 = duracao_chamada_2,
107      duracao_chamada_3 = duracao_chamada_3,
108      duracao_chamada_4 = duracao_chamada_4,
109      tipo_routing = tipo_routing;
110     gatesizes:
111       para_estacoes[numero_estacoes],
112       das_estacoes[numero_estacoes];
113     display: "p=110,110,ring,100,100;i=fork";
114   connections:
115     for i=0..numero_estacoes-1 do
116       central[i].para_estacoes[i] --> gerador[i].da_estacao;
117       central[i].das_estacoes[i] <-- gerador[i].para_estacao;
118     endfor;
119     for i=0..numero_estacoes-1, j=0..numero_estacoes-1 do

```

```

120     central[i].para_estacoes[j] --> central[j].das_estacoes[i] if i != j;
121     endfor;
122 endmodule
123
124 network rede : rede_inteira
125     parameters:
126         numero_estacoes = input(5, "Numero de centrais"),
127         numero_classes = input(4, "Numero de classes de trafego");
128 endnetwork

```

Fig. A-1 – Código do ficheiro *ned*.

O código, apresentado na figura anterior, constitui o ficheiro de texto com extensão *ned*, construído para criar parte do modelo de simulação em OMNeT++.

No modelo OMNeT++ criado para simular o sistema foi definido um módulo de sistema (chamado *rede*, linhas 124 a 128), um módulo composto (chamado *rede\_inteira*, linhas 71 a 122) que tem submódulos, de dois tipos (*gerador* e *central*, linhas 85 a 113). Os submódulos deste módulo composto são todos módulos simples. Os tipos destes módulos simples foram definidos previamente (um chamado *central*, linhas 1 a 53, e outro *gerador*, linhas 55 a 69). O módulo composto tem dois vectores de submódulos, tendo tantas instâncias *central* como número de centrais (linha 97) e também tantas instâncias *gerador* como número de centrais (linha 86).

No módulo de sistema (chamado *rede*, linhas 124 a 128, instância do módulo composto *rede\_inteira*), as linhas 126 e 127 dão a possibilidade ao utilizador de introduzir valores para os parâmetros *numero\_estacoes* (número de centrais) e *numero\_classes* (número de classes de tráfego), respectivamente, se eles não forem encontrados no ficheiro de configuração (ficheiro *ini*) quando for executada a simulação. Se o utilizador não introduzir nenhum valor os parâmetros ficam com os valores por omissão que foram indicados nessas linhas, respectivamente.

Tanto o módulo *gerador* como o módulo *central* possuem um parâmetro para armazenar o número de centrais do sistema a simular (linha 62 e 46 respectivamente). Os geradores têm também um parâmetro para armazenar o nome do ficheiro de onde vai ser lida informação de inicialização de algumas variáveis e da topologia efectiva da rede (linha 65).

Além dos parâmetros já referidos do módulo simples *central* aparecem ainda outros na sua definição que se descrevem a seguir: Os parâmetros que se encontram nas linhas 3 a 29 são parâmetros que são necessários para cada um dos vários métodos de encaminhamento implementados DCR (linhas 6 a 13), DAR (linhas 15 a 19), RTNR (linhas 21 a 25) e FAR (linhas 27 a 29); Os parâmetros que se encontram nas linhas 41 a 44 são parâmetros que são usados por todos os métodos de encaminhamento. Estes quatro parâmetros servem para possibilitar a existência de chamadas com durações diferentes, nomeadamente para classes de chamadas diferentes. Para este fim, é permitido que cada classe possa ter um de entre quatro valores possíveis de durações; Os parâmetros *tipo\_routing* e *num\_classes* (linhas 45 e 47) são usados pela *central* para determinar qual o método de encaminhamento que deve

utilizar e quantas classes de serviço devem ser consideradas, respectivamente; Os parâmetros nas linhas 48 e 49 são usados para obter o tempo em que se deve iniciar a recolha de dados e o tempo em que deve terminar a simulação, respectivamente; O parâmetro `fich_output_folha_calculo` (linha 34) é usado para obter o nome do ficheiro onde devem ser guardados os dados de saída obtidos na simulação; Por último o parâmetro `factor_carga` (linha 37) e o parâmetro `fich_topologia` (linha 38) são necessários aqui para se conseguir escrever os seus valores no ficheiro de saída dos dados da simulação.

Também para o módulo simples `gerador` além dos parâmetros já referidos aparecem ainda outros na sua definição que se passam a descrever: Os parâmetros que se encontram nas linhas 57 a 60 e 63 têm a mesma função para o `gerador` como a que tinham para a `central` os parâmetros com os mesmos nomes; O parâmetro `factor_carga` (linha 61) vai ser usado para obter o factor multiplicativo da carga de tráfego, o que possibilita a criação de várias situações de sobrecarga.

Os parâmetros dos módulos simples `gerador` e `central` vão ser utilizados no desenvolvimento dos seus algoritmos respectivamente. No módulo composto `rede_inteira` (linhas 74 a 84) foram definidos alguns parâmetros sendo os valores destes parâmetros utilizados na inicialização de alguns dos parâmetros dos seus submódulos (linhas 88 a 95 e 99 a 109). Para além disto, o parâmetro na linha 80 é também utilizado na definição da topologia, como já foi referido.

Cada `gerador` tem apenas uma porta de entrada e uma porta de saída (linhas 67 e 68). Uma `central` tem um número de portas de saída e de entrada igual ao número de centrais (linhas 51, 52, 111 e 112). Uma das portas de saída de cada `central` liga a uma porta de entrada do `gerador` que lhe está associado, e uma das portas de entrada liga a uma porta de saída do `gerador` que lhe está associado (linhas 115 a 118). As portas de entrada e saída restantes são utilizadas para fazer as ligações entre as várias centrais (linhas 119 a 121 – criando uma rede completamente emalhada), como se pode ver na figura 5.1 do capítulo 5.

Os símbolos que representam as centrais e os geradores foram especificados nas linhas 96 e 113 assim como as posições que cada um ocupa. Esta informação só é utilizada aquando da simulação em ambiente gráfico.

Durante a simulação os módulos comunicam enviando mensagens através das ligações. Como não foram atribuídos parâmetros às ligações, não existe atraso na propagação das mensagens pelas ligações.

# Apêndice B

## Programação de Módulos Simples – Mecanismos de Suporte

Esta secção contém as funções e os objectos mais importantes que podem ser utilizados na implementação dos módulos simples. Para ver outras funções deve-se consultar Manual de Referência do OMNeT++ [Varga00]. Todas as classes da biblioteca de simulação do OMNeT++ são derivadas da classe *cObject*, o que faz com que as várias classes partilhem algumas características.

### B.1 Acesso aos parâmetros e portas dos módulos

Os parâmetros dos módulos, objectos *cPar*, podem ser acedidos utilizando a função membro *par()* da classe *cModule*. A classe *cPar* é um objecto genérico para o armazenamento de valores e suporta conversões para tipos numéricos.

As portas de um módulo, objectos *cGate*, são armazenadas no módulo num *array*. A posição das portas no *array* de portas é chamado o ID da porta. O envio e a recepção de mensagens aceitam tanto o nome da porta como o ID da porta, mas por questões de rapidez é preferível utilizar o ID da porta. Os IDs das portas não são alterados durante a execução, por isso podem ser obtidos inicialmente e a partir daí passar apenas a serem utilizados em vez dos nomes. Para obter o ID de uma porta deve utilizar-se a função *findGate()* da classe *cModule*. A função membro *gate()* também da classe *cModule* pode ser utilizada para obter um ponteiro de uma dada porta. O tamanho de uma porta que é um vector pode ser obtido utilizando a função membro *size()* da classe *cGate*. Se uma porta é uma porta de saída a porta à qual está ligada pode ser obtida através da função membro *toGate()*. A função membro simétrica é a função *fromGate()*. A função membro *ownerModule()* da classe *cGate* dá-nos um ponteiro para um módulo que possui uma determinada porta.

## B.2 Envio e recepção de mensagens

Toda a interacção entre diferentes módulos simples que compõe o modelo é efectuada através da troca de mensagens entre eles. Para além disso, as mensagens podem ser também usadas para implementar ciclos de atraso e sincronização de processos.

As mensagens são objectos da classe *cMessage*. Os vários membros de cada mensagem são o nome, a espécie, o comprimento, a prioridade, marcador de erro e registo temporal. Todos os membros têm valores por omissão, pelo que alguns podem não ser definidos explicitamente pelo utilizador. O membro espécie e o membro registo temporal podem ser usados livremente para qualquer fim, o nome descreve a mensagem, o comprimento e prioridade são usados em conjugação com os parâmetros de ligação entre módulos para determinar características de comunicação, como atrasos e possibilidades de perda, enquanto o marcador de erro indica se ocorreu algum erro na comunicação da mensagem entre módulos.

Depois de uma mensagem ter sido criada os seus campos de dados podem ser alterados pelas funções seguintes: *setKind()*, *setLength()*, *setPriority()*, *setError()* e *setTimestamp()*.

Os valores dos campos de dados de uma mensagem podem ser obtidos pela funções seguintes: *kind()*, *length()*, *priority()*, *isbad()*, *isgood()* e *timestamp()*.

### B.2.1 Associar parâmetros e objectos às mensagens

A cada mensagem podem ser associados vários parâmetros. No OMNeT++, parâmetros são objectos do tipo *cPar*. Para acrescentar um novo parâmetro a uma mensagem deve utilizar-se a função membro *addPar()*. Para obter a referência para um parâmetro deve usar-se a função membro *par()* da classe *cPar*, e porque esta classe suporta conversões para outros tipos e atribuições, é fácil ler e atribuir um valor a um parâmetro.

Para saber se uma mensagem possui ou não um determinado parâmetro pode-se utilizar a função *findPar()*.

Os parâmetros de uma mensagem são armazenados num objecto do tipo *cArray* que pode armazenar qualquer tipo de objecto, não apenas *cPars*. A função membro, da classe *cMessage*, *parList()* dá-nos acesso directo ao *cArray* interno, logo, chamando as funções membro da classe *cArray*, podem-se associar quaisquer objectos a uma mensagem.

Se não forem associados parâmetros à mensagem e não for chamada a função *parList()* o objecto *cArray* interno não será criado. Isto poupa simultaneamente espaço e tempo de execução.

## B.2.2 Duplicação de mensagens

É muitas vezes necessário duplicar uma mensagem (por exemplo, enviar uma e manter uma cópia). Isto pode ser conseguido por um de dois modos, sendo no exemplo seguinte são utilizadas as duas possibilidades. Considere para tal que pretende duplicar a mensagem, *msg*:

```
cMessage *copia1 = (cMessage *) msg ->dup();
```

```
cMessage *copia2 = new cMessage (*msg);
```

Quase todos os tipos de objectos podem ser duplicados deste modo, e não apenas os objectos *cMessage*.

## B.2.3 Enviar mensagens

Uma vez uma mensagem criada, ela pode ser enviada através de uma porta de saída utilizando para isso a função (função membro da classe *cModule*) *send()*. Existe também uma função, chamada *sendDirect()*, que permite enviar uma mensagem directamente para um módulo para o qual não existe ligação. Todas as funções que permitem enviar mensagens fazem um teste para determinar se a mensagem se encontra em condições de poder ser enviada. Se assim não for, obtém-se como resultado uma mensagem de erro.

## B.2.4 Atrasar mensagens

É muitas vezes necessário modelar um atraso seguido imediatamente do envio de uma mensagem. Isto é conseguido se for utilizada a função *wait()* antes da função *send()*.

A função *wait()* suspende a função do módulo por um dado tempo de simulação. As mensagens que cheguem durante a espera serão acumuladas na fila de espera.

Se o módulo precisa de reagir a mensagens que cheguem durante o atraso então a função *wait()* não pode ser utilizada. Nesse caso deve utilizar-se a função *sendDelayed()* que atrasa o envio. A função *sendDelayed()* tem os mesmos parâmetros que a função *send()* e além desses tem um parâmetro para indicar o valor do atraso.

## B.2.5 Receber mensagens

Uma mensagem pode ser recebida, por um módulo, por uma de várias funções, sendo a função *receive()* a mais geral.

Os objectos dos módulos simples contêm um objecto fila de espera, integrado neles, chamado *putAsideQueue*. Essa fila de espera é usada por algumas das funções que recebem mensagens.

Existem dois grupos de funções que recebem mensagens:

1. *receive()* e *receiveOn()*
2. *receiveNew()* e *receiveNewOn()*

As funções do primeiro grupo testam a fila de espera primeiro e tentam obter uma mensagem dela. Só se não encontrarem lá uma mensagem apropriada é que esperam por novas mensagens.

As funções do segundo grupo esperam sempre por novas mensagens, ignorando as que já se encontrem em fila de espera.

As funções com *On* no nome esperam que as mensagens cheguem numa porta específica. Mensagens que chegam por outras portas são colocadas na fila de espera. As funções que não têm *On* no nome, aceitam uma mensagem proveniente de qualquer porta.

## B.2.6 Escalonamento de mensagens

Um módulo pode enviar uma mensagem para si próprio com a função *scheduleAt()*. Esta função juntamente com a função *cancelEvent()* (função que remove uma dada mensagem do conjunto de acontecimentos futuros) pode ser usada para implementar relógios.

## B.3 Geração de números aleatórios

O gerador de números aleatórios utilizado no OMNeT++ é um gerador congruencial linear (“Linear Congruential Generator” - LCG) com um comprimento cíclico de  $2^{31} - 2$ .

Se um programa de simulação usar números aleatórios para mais do que um fim, então esses números deverão vir de geradores de números aleatórios diferentes. O OMNeT++ tem vários geradores de números aleatórios independentes (32 geradores por omissão; este número pode ser alterado no ficheiro *defs.h*).

Para evitar correlações indesejadas é importante que sejam usadas séries não sobrepostas de números aleatórios nas várias corridas de simulação e nas várias fontes de números aleatórios dentro de uma corrida de simulação. Por esta razão, os geradores devem começar com sementes bem separadas. Para seleccionar boas sementes pode ser usado o programa *seedtool*.

### B.3.1 Usar os geradores de números aleatórios directamente

O gerador é acessível directamente através da função *inrand()* e esta função dá-nos um número aleatório inteiro no intervalo  $[1, INTAND\_MAX - 1]$ . A semente pode ser especificada no ficheiro *ini*

(*random-seed*=) ou atribuída directamente dentro do módulo simples com a função *randseed()*. As funções *inrand()* e *randseed()* usam o gerador 0. Existe outra variante destas funções nas quais é especificado o gerador, são elas as funções *genk\_inrand()* e *genk\_randseed()*.

Outra função ainda é a função *dblrand()* (função baseada na função *inrand()*) que nos dá um número aleatório no intervalo  $[0.0,1.0]$ . Esta função também tem uma variante na qual é especificado o gerador utilizado; é ela a função *genk\_dblrnd()*.

## B.3.2 Geração de variáveis aleatóreas

As funções seguintes:

*uniform()*, *intuniform()*, *exponential()*, *normal()* e *truncnormal()*

devolvem uma variável aleatória de diferentes distribuições. Todas estas funções são baseadas na função *dblrand()* e podem ser usadas também no ficheiro NED.

A função *intuniform()* gera um número inteiro no intervalo compreendido entre os limites inferior e superior, inclusive, que são dados como argumentos à função.

A função *truncnormal()* é a função normal truncada para valores não negativos (ou seja os valores obtidos, com esta função, são sempre valores não negativos).

Para todas estas funções também existem variantes nas quais é especificado o gerador utilizado. As funções são as seguintes:

*genk\_uniform()*, *genk\_intuniform()*, *genk\_exponential()*, *genk\_normal()* e *genk\_truncnormal()*

Se as distribuições anteriores não forem suficientes, novas funções podem ser escritas, em C++, e incorporadas no simulador.

## B.3.3 Números aleatórios obtidos a partir de histogramas

É também possível especificar uma distribuição através de um histograma da função densidade de probabilidade a ela associado.

## B.4 Parâmetros – classe *cPar*

A classe *cPar* foi projectada para armazenar um valor. O valor armazenado é um valor numérico mas o tipo *string*, *pointer* ou qualquer outro tipo também são suportados.

A classe *cPar* é usada no OMNeT++ nos parâmetros das mensagens e nos parâmetros dos módulos.

Para armazenar um valor num parâmetro pode usar-se a função membro *setValue()*, chamando o construtor ou utilizando o operador de atribuição (operador redefinido). Para obter o valor de um parâmetro deve utilizar-se o operador de conversão de tipo (operador redefinido).

Quando se pretende usar o valor de um objecto da classe *cPar*, pela primeira vez, o seu valor será procurado no ficheiro de configuração *ini*. Se não for encontrado lá, o utilizador é solicitado para introduzir o valor interactivamente.

## B.5 Funções estatísticas

### B.5.1 A classe *cStatistic* e as suas derivadas

Existem várias classes estatísticas e várias classes para recolha dos resultados todas elas derivadas da classe abstracta *cStatistic*. Estas classes são: *cStdDev*, *cWeightedStdDev*, *cLongHistogram*, *cDoubleHistogram*, *cVarHistogram*, *cPSquare* e *cKSplit*.

*cStdDev* armazena o número de amostras, a média, desvio padrão, valor mínimo e valor máximo, etc.

*cWeightedStdDev* é semelhante a *cStdDev* mas permite atribuir um peso a cada observação.

*cLongHistogram* e *cDoubleHistogram* são derivadas de *cStdDev* e armazenam uma aproximação das distribuições das observações usando células equidistantes de histogramas.

*cVarHistogram* implementa um histograma onde as células não precisam de ser do mesmo tamanho.

Para inserir uma observação num objecto de estatísticas deve usar-se a função *collect()* ou o operador *+=*. Os métodos da classe *cStdDev* para obter estatísticas do objecto são: *sample()*, *min()*, *max()*, *mean()*, *stddev()*, *variance()*, para obter respectivamente o número de amostras, o valor mínimo, o valor máximo, desvio padrão e variância.

As classes das aproximações das distribuições (classes histogramas) são derivadas da classe *cDensityEstBase*. Essas classes assumem que as observações estão dentro de uma determinada gama. Essa gama pode ser especificada explicitamente ou pode deixar-se que seja o objecto a determinar a gama por ele próprio através do registo de algumas observações iniciais. Os métodos que nos permitem especificar gamas são: *setRange()*, *setRangeAuto()*, *setRangeAutoLower()*, *setRangeAutoUpper()* e fazem parte da classe *cDensityEstBase*.

As funções membro *underflowCell()* e *overflowCell()* dão-nos o número de observações que caíram fora da gama do histograma.

A função membro *cells()* dá-nos o número de células, *basepoint()* usa-se para obter um dos limites de uma determinada célula, *cell()* dá-nos o número de observações numa determinada célula e *cellPDF()* dá-nos o valor PDF (“Probability Density Function”) numa determinada célula.

As funções *pdf()* e *cdf()* dão-os o valor da função densidade de probabilidade e o valor da função cumulativa num dado ponto, respectivamente.

Existem dois modos de gerar números aleatórios a partir da distribuição armazenada num objecto. Um modo é através do uso da função membro *random()*, o segundo é usar um parâmetro (objecto da classe *cPar*).

As classes estatísticas têm a função membro *loadFromFile()* que lê os dados de um histograma de um ficheiro de texto.

## B.5.2 Detecção do fim do período de transição e validação dos resultados

Podem ser associados outros objectos aos objectos estatísticos. Esses objectos, do tipo *cTransientDetection* e *cAccuracyDetection*, permitem determinar o fim do período de transição e se os resultados são suficientemente válidos, respectivamente.

## B.6 Terminar uma simulação

Uma simulação pode ser terminada com a função *endSimulation()*. Existem outros modos de terminar uma simulação, por exemplo a simulação pode ser terminada depois de um certo tempo simulado ou depois de um certo tempo de CPU.

## B.7 Registo de resultados

### B.7.1 Vectores de saída– A classe *cOutVector*

Objectos do tipo *cOutVector* são os responsáveis pela escrita de uma série temporal de dados (referidos como vectores de saída) para um ficheiro. A função membro *record()* é usada para escrever um valor com o registo da altura em que a operação ocorreu, sendo então escrito um par de valores e não apenas um valor.

Todos os objectos *cOutVectores* são escritos num ficheiro comum. Esse ficheiro é um ficheiro de texto; cada chamada à função membro *record()* produz uma linha nesse ficheiro.

## B.7.2 Saída de escalares

A saída de escalares é suposta registar um único valor por variável por corrida de simulação. Uma utilização da saída de escalares é registar resultados no final de uma corrida de simulação.

As saídas de escalares são registadas com a função membro *recordScalar()*.

## B.7.3 Saída dos objectos estatísticos

Para registar todos os objectos estatísticos deve utilizar-se a função *recordStats()*. Esta função, assim como a função *recordScalar()*, são colocadas normalmente na função *finish()*, função membro dos módulos simples que deve ser redefinida e que é chamada no fim da simulação.

As chamadas às funções anteriores escrevem num ficheiro escalar de saída (ficheiro de texto). Os dados de uma corrida de simulação são acrescentados ao fim do ficheiro, sendo separados dos dados da corrida anterior por linhas especiais

## B.8 Outros

### B.8.1 Contentores

#### B.8.1.1 Fila de espera – classe *cQueue*

A classe *cQueue* é uma classe que representa uma fila de espera. Os elementos da fila de espera podem ser objectos de tipos derivados da classe *cObject* (como já referido quase todas as classes da biblioteca do OMNET++ são derivadas de *cObject*) tais como *cMessage*, *cPar*, etc..

Um membro dos módulos simples chamado *putAsideQueue* é do tipo *cQueue*. A *putAsideQueue* armazena objectos *cMessage*. Contudo a fila de espera de mensagens global, ou seja o conjunto de acontecimentos futuros, “Future Event Set” - FES, não é um objecto *cQueue* é um objecto da classe *cMessageHeap* que é um pouco mais eficiente.

As funções membro básicas da classe *cQueue* são as funções *insertHead()* e *getTail()*, que inserem e removem um objecto da fila de espera respectivamente. Uma outra função membro é a função *length()*, a qual dá o número de itens na fila de espera. Outra função é a função *empty()* que diz se há algum elemento na fila de espera. Outras funções ainda são as funções *insertBefore()* *insertAfter()* as quais inserem um novo item exactamente antes e depois de um item especificado, as funções *peekTail()* e *peekHead()* que dão um ponteiro para o objecto do início e do fim da fila de espera, as funções *getTail()* e *getHead()* que removem itens de ambas as pontas da fila de espera e a função *get()* remove qualquer item da fila de espera através do seu ponteiro.

### B.8.1.2 Array dinâmico – classe *cArray*

A classe *cArray* é uma classe que armazena objectos derivados da classe *cObject*. Um objecto da classe *cArray* armazena os ponteiros dos objectos em vez de fazer uma cópia. Um objecto da classe *cArray* funciona como um *array*, mas se ficar completo o objecto cresce automaticamente.

A função membro *add()* acrescenta um elemento, e dá-nos a posição onde o elemento foi inserido. A função membro *find()* encontra um objecto no *array*, através do ponteiro do objecto ou do seu nome. Por último, a função *remove()*, remove a referência a um objecto de um *array*, no entanto o objecto não é apagado. Para apagar o objecto deve-se utilizar a função *delete*.

Objectos da classe *cArray* são usados no OMNeT++ para armazenar parâmetros associados com as mensagens, e internamente, para armazenar os parâmetros e as portas dos módulos.

### B.8.1.3 Contentores de itens que não são objectos – classe *cLinkedList* e *cBag*

*cLinkedList* e *cBag* são duas classes que armazenam itens que não são derivados do *cObject*. A classe *cLinkedList* funciona como a classe *cQueue* e a classe *cBag* funciona como a classe *cArray*. Estas classes são necessárias quando tem que se tratar com itens que não são objectos derivados da classe *cObject*.

## B.8.2 Nome dos objectos na rede e identificadores numéricos na rede

Os objectos no OMNeT++ têm três funções membro que nos dão o nome do objecto. Essas funções são: *name()*, *fullName()* e *fullPath()*.

Se um módulo faz parte de um vector de módulos as funções membro *index()* e *size()* podem ser usadas para determinar qual o seu índice no vector e qual o tamanho do vector de que faz parte.

### B.8.3 Acedendo a outros módulos na rede

Os módulos compostos antecessores podem ser acedidos através da função membro *parentModule()*.

Para aceder a todos os módulos dentro de um módulo composto deve usar-se a classe *cSubModIterator*.

Para determinar o módulo que está do outro lado de uma ligação devem usar-se as funções membro: *fromGate()*, *toGate()* e *ownerModule()*; estas funções são funções membro da classe *cGate*. Para portas de entrada deve usar-se a função *fromGate()* em vez da função *toGate()*.

## B.8.4 Mostrar informação acerca da actividade dos módulos

O objecto global chamado *ev* pode usar-se para enviar informação para o utilizador de acordo com as características do ambiente actual de utilização (gráfico ou linha de comando). Podem enviar-se dados para o *ev* usando os operadores de entrada/saída, `>>/<<`, tal como nas streams do C++.

## B.8.5 Conselhos para acelerar uma simulaçãoC

Alguns modos de tornar uma simulação mais rápida são:

- Armazenar os parâmetros em variáveis locais e usá-las em vez de referências aos parâmetros;
- Usar o número de uma porta em vez do seu nome;
- Minimizar a criação e destruição de mensagens. Reutilizar mensagens sempre que possível.

# Apêndice C

## Entradas para a Simulação

### C.1 Ficheiro da topologia (ficheiro txt)

A rede que se pretende simular é especificada pela introdução das suas características num ficheiro de texto. O nome deste ficheiro é especificado no ficheiro de configuração (ficheiro *ini*), no exemplo do ficheiro de configuração apresentado na última secção deste apêndice, o nome do ficheiro de topologia que foi especificado foi `red20000808-4.txt`.

#### C.1.1 Descrição da Estrutura do ficheiro de topologia

Nas primeiras linhas desse ficheiro são colocados os valores das capacidades (número de circuitos) dos vários feixes da rede. Esses valores, inteiros, são dispostos num número de linhas igual ao número de centrais na rede menos um. Na primeira linha do ficheiro são colocados os valores das capacidades de todos os feixes que ligam a central 1 a qualquer outra central. O primeiro valor da linha corresponde à capacidade do feixe que liga a central 1 à central 2, depois de um *tab* segue-se o segundo valor que corresponde à capacidade do feixe que liga a central 1 à central 3, e assim por diante até se esgotarem as centrais destino. A segunda linha começa por um *tab* seguido dos valores das capacidades de todos os feixes que ligam a central 2 a qualquer outra central com número superior a 2; a terceira linha começa por dois *tabs* seguidos dos valores das capacidades de todos os feixes que ligam a central 3 a qualquer outra central com número superior a 3, e assim por diante até se completar o número de linhas. Obtém-se assim o correspondente a uma matriz triangular, que designamos por matriz das capacidades dos feixes.

Além da informação já referida, o ficheiro contém ainda um ou vários blocos de valores, formados cada um por várias linhas. O número de blocos que aparecem no ficheiro é igual ao número de classes de

tráfego que vão coexistir no modelo. Cada um destes blocos contém a informação necessária para representar uma das classes de tráfego. Cada bloco é formado por uma linha com três valores seguida do correspondente a três matrizes, no caso da informação desse bloco representar uma classe de tráfego de Poisson, ou seguida de duas linhas de valores mais três matrizes, no caso da informação desse bloco representar uma classe de tráfego de população finita.

Os três valores da primeira linha de cada bloco representam, pela ordem que são introduzidos, a classe de tráfego (0 se trata de uma classe de tráfego de Poisson e 1 se se trata de uma classe de tráfego de população finita), o número de circuitos necessários às chamadas dessa classe em cada feixe e o tipo de duração da chamada a usar (o modelo permite utilizar um de quatro valores). No caso da informação do bloco corresponder a uma classe de tráfego de Poisson, a seguir à linha dos três valores vem o correspondente a três matrizes. A primeira matriz tem tantas linhas como colunas e em número igual ao número de centrais. Esta matriz é uma matriz quadrada e corresponde à matriz do tráfego exógeno. A diagonal desta matriz é composta por zeros em todas as posições (o que indica que não existe tráfego de uma central para si própria). Na primeira linha o primeiro valor é 0 pela razão indicada anteriormente, o segundo valor é o valor correspondente ao tráfego exógeno da central 1 para a central 2, o terceiro valor é o valor correspondente ao tráfego exógeno da central 1 para a central 3, e assim por diante até serem esgotadas todas as centrais destino. Na segunda linha o primeiro valor é o valor correspondente ao tráfego exógeno da central 2 para a central 1, o segundo valor é zero pela razão já indicada, o terceiro valor é o valor correspondente ao tráfego exógeno da central 2 para a central 3, e assim por diante até serem esgotadas todas as centrais destino. Este processo repete-se até que sejam completadas todas as linhas da matriz. A segunda matriz do bloco é uma matriz triangular, que tem um aspecto análogo à matriz das capacidades dos feixes, a diferença sendo que na segunda matriz do bloco os seus valores não correspondem a capacidades dos feixes mas sim a circuitos reservados nos feixes. Nesta matriz são colocados os valores para as reservas fixas de circuitos<sup>6</sup>. A última matriz do bloco tem um aspecto análogo à primeira matriz do bloco, a diferença é que na última matriz os seus valores não correspondem a tráfegos exógenos entre pares de centrais mas sim as capacidades mínimas garantidas nos feixes, se necessárias, para encaminhamento directo para esta classe de tráfego.

No caso da informação do bloco corresponder a uma classe de tráfego de população finita a seguir à linha dos três valores vêm mais duas linhas de valores e depois destas duas linhas mais três matrizes. Cada uma das duas linhas contém tantos valores quantas as centrais na rede. Na primeira das duas linhas, os valores representam o número de fontes para as centrais correspondentes. Na segunda das duas linhas, cada valor representa o tráfego de chamadas por fonte livre da central correspondente. A primeira matriz deste bloco é uma matriz com o número de linhas igual ao número de colunas e também igual ao número de centrais na rede. Os valores da primeira linha desta matriz representam as proporções de

---

<sup>6</sup> A reserva de circuitos está definida no capítulo 2.

tráfego da central 1 para cada uma das centrais na rede, respectivamente. Os valores da segunda linha desta matriz representam as proporções de tráfego da central 2 para cada uma das centrais na rede, respectivamente. E assim por diante até se completar o número de linhas. A diagonal desta matriz é portanto preenchida por zeros. As duas últimas matrizes representam o mesmo que as duas últimas matrizes de um bloco que corresponde a uma classe de tráfego de Poisson.

## C.1.2 Exemplo de um ficheiro de topologia

81	32	988	78	115		
	96	182	161	167		
		15	48	29		
			214	151		
				464		
0	1	1				
0	13.735	3.485	128.905	10.235	14.555	
13.735	0	12.555	50.805	38.39	41.28	
3.485	12.555	0	5.96	3.43	6.625	
128.905	50.805	5.96	0	39.71	41.5	
10.235	38.39	3.43	39.71	0	63.555	
14.555	41.28	6.625	41.5	63.555	0	
0	0	0	0	0		
	0	0	0	0		
		0	0	0		
			0	0		
				0		
0	20	8	205	19	19	
21	0	34	65	56	59	
8	34	0	5	18	10	
205	65	6	0	77	53	
19	57	18	77	0	166	
20	60	11	54	166	0	
0	2	1				
0	2.747	0.697	25.781	2.047	2.911	
2.747	0	2.511	10.161	7.678	8.256	
0.697	2.511	0	1.192	0.686	1.325	
25.781	10.161	1.192	0	7.942	8.3	
2.047	7.678	0.686	7.942	0	12.711	
2.911	8.256	1.325	8.3	12.711	0	
0	0	0	0	0		
	0	0	0	0		
		0	0	0		
			0	0		
				0		
0	8	2	82	8	8	
8	0	14	26	24	24	
2	14	0	2	6	4	
82	26	2	0	30	22	
8	24	6	30	0	66	
8	24	4	22	66	0	
1	6	1				
329	0	0	0	0	0	
0.124679635	0	0	0	0	0	0

```

0      8.036158324  2.039025246  75.42053067  5.988356786  8.515928971
0      0            0            0            0            0
0      0            0            0            0            0
0      0            0            0            0            0
0      0            0            0            0            0
0      0            0            0            0            0

0      0            0            0            0
      0            0            0            0
           0            0            0
                0            0
                        0

0      24          12          294          24          36
0      0            0            0            0            0
0      0            0            0            0            0
0      0            0            0            0            0
0      0            0            0            0            0
0      0            0            0            0            0

1      24          1
82     0            0            0            0            0
0.055582114  0            0            0            0            0

0      8.036158324  2.039025246  75.42053067  5.988356786  8.515928971
0      0            0            0            0            0
0      0            0            0            0            0
0      0            0            0            0            0
0      0            0            0            0            0
0      0            0            0            0            0

0      0            0            0            0
      0            0            0            0
           0            0            0
                0            0
                        0

0      0            0            120          0            24
0      0            0            0            0            0
0      0            0            0            0            0
0      0            0            0            0            0
0      0            0            0            0            0
0      0            0            0            0            0

```

Fig. C-1 – Conteúdo do ficheiro de topologia correspondente ao sistema 3.

Este exemplo de um ficheiro de topologia representa uma rede à qual vão ser oferecidos tráfegos de 4 tipos, sendo dois tipos de tráfego de Poisson e dois tipos de tráfego de população finita. Este ficheiro de topologia corresponde ao sistema 3 apresentado no capítulo 4.

## C.2 Ficheiros de configuração (ficheiro ini)

É no ficheiro de configuração onde devem ser colocadas as opções que vão controlar como a simulação é executada. Nesse ficheiro podem também ser inicializados parâmetros do modelo.

De modo a automatizar a construção dos ficheiros *ini*, deve recorrer-se ao programa *gerador\_ini.pl*, que foi desenvolvido em Perl. Esse programa pega num ficheiro (ficheiro com extensão *cfg*) muito mais compacto que o ficheiro *ini* e a partir dele constrói o ficheiro *ini*.

## C.2.1 Descrição da estrutura do ficheiro criado para gerar o ficheiro de configuração automaticamente (ficheiro *cfg*)

As linhas iniciais do ficheiro (linhas começadas pelo caracter #) contêm comentários. As linhas seguintes contêm uma lista de parâmetros e os valores respectivos. Os parâmetros das primeiras linhas devem ser os parâmetros especiais. Como parâmetros especiais existe apenas um com o nome `routings_a_tentar`. Devem ser indicados aí, separados por vírgulas, os nomes dos métodos de encaminhamento que se pretendem simular. Depois deste aparecem os parâmetros normais os quais devem seguir a seguinte sintaxe:

```
[metodo*]nome_parametro=valor[.valor:intervalo]
```

No caso do parâmetro ser específico de um método de encaminhamento o nome do parâmetro deve ser precedido pelo nome do método correspondente. A cada parâmetro pode-se dar um ou vários valores. Devem ser dados vários valores no caso de se pretenderem efectuar várias corridas cada uma das quais com seu valor para o parâmetro. Nesse caso devem-se indicar os limites inferior e superior, bem como o intervalo entre os valores dos parâmetros. O parâmetro global `corrida` pode ser usado para criar no ficheiro *ini* tantas corridas independentes, para cada simulação, como as aí indicadas. O parâmetro global `rede` vai ser usado para indicar no ficheiro *ini* que a rede que deve ser usada é a rede com o nome `rede`. Os restantes parâmetros globais vão ser usados para criar, no ficheiro *ini*, parâmetros que são comuns a todos os métodos. Estes parâmetros estão descritos no apêndice E.

Os parâmetros dos métodos vão ser usados para criar os parâmetros relativos a cada método no ficheiro *ini*. Estes parâmetros estão também descritos no apêndice E.

Repare que não vão corresponder parâmetros no ficheiro *ini* gerado, a todos os parâmetros deste ficheiro, sendo usados apenas os realmente necessários para os métodos simulados.

## C.2.2 Exemplo de um ficheiro criado para gerar o ficheiro de configuração automaticamente

```
#parametros especiais
#devem vir antes dos parametros normais
#routings_a_tentar=primeiro_metodo,...ultimo_metodo

#os parametros que podem ser alterados tem
#a seguinte sintaxe:
#[metodo*]nome_parametro=valor[.valor:intervalo]

#parametros que nao podem ser alterados directamente:
#nome do ficheiro de resultados (ficheiro de escalares)
#nome do ficheiro de resultados (ficheiro de tabelas)

#parametros especiais
routings_a_tentar=DIRECTO,DAR,RTNR
```

```
#parametros normais
rede="rede"
prefix_fich="mult-4-"
duracao_chamada_1=1m
duracao_chamada_2=1m
duracao_chamada_3=1m
duracao_chamada_4=1m
factor_carga=1.0..2.0:0.1
inicio_recolha=24h
fim_recolha=48h
display-update=2h
numero_estacoes= 6
numero_classes= 4
mostra_tudo= 0
corrida=1
fich_topologia= "red20000808-4.txt"

#parametros dos metodos

#parametros do metodo dcr
DCR*dcr_tau= 1s
DCR*dcr_delta= 1s
DCR*dcr_theta= 0.9
DCR*dcr_intervalo_atualizacao= 3m
DCR*dcr_vtrafk_factor= 1.1
DCR*dcr_rk_factor= 0.05
DCR*dcr_tipo_reserva_din= 0..2:1

#parametros do metodo dar
DAR*dar_intervalo_atualizacao= 3m
DAR*dar_vtrafk_factor= 1.1
DAR*dar_rk_factor= 0.05
DAR*dar_tipo_reserva_din= 0..1:1

#parametros do metodo rtnr
RTNR*rtnr_intervalo_atualizacao= 3m
RTNR*rtnr_vtrafk_factor= 1.1
RTNR*rtnr_rk_factor= 0.05
RTNR*rtnr_tipo_reserva_din= 0..1:1

#parametros do metodo FAR
FAR*far_fich_input_tabelas_routing="mitra_far%d.txt"
FAR*far_ncaminhos=1..3:1
```

Fig. C-2 – Código para geração do ficheiro de configuração (*ini*).

### C.2.3 Descrição da estrutura do ficheiro de configuração (ficheiro *ini*)

As linhas iniciais do ficheiro (linhas começadas pelo caracter #) contêm comentários. Estes comentários foram criados automaticamente pelo programa de conversão do ficheiro *cfg* para o ficheiro *ini* e indicam de modo resumido todos os valores que vão ser usados nas várias simulações.

A primeira secção do ficheiro *ini* (secção General) é criada automaticamente pelo programa de conversão e contém as seguintes entradas:

- `network` – especificação da rede que deve ser usada tal como definido no ficheiro *cfg*;
- `ini-warnings` – com o valor `yes` para mostrar avisos em relação ao ficheiro *ini*;
- `warnings` – com o valor `yes` para mostrar avisos em tempo de execução;

- `snapshot-file` – indica o nome do ficheiro `telefone.sna` como sendo o nome do ficheiro de saída de `snapshots` (imagens do sistema - valor dos parâmetros numa determinada altura);
- `log-parchanges` – é uma opção obrigatória, sem interesse no nosso caso por isso definida com o valor `no`;
- `total-stack-kb` – definição do tamanho da pilha que vai ser usada para as variáveis locais dos módulos;
- `distributed` – definida com o valor `false` para indicar que não há execução em paralelo;
- `include meusseeds` – para que seja incluído o ficheiro `meusseeds.ini` o qual contem os `seeds` iniciais;
- `module-messages` – definida com o valor `no` para que não sejam mostradas as mensagens criadas pelos módulos;
- `verbose-simulation` – definida com o valor `no` para que não sejam mostradas as mensagens criadas pelos `ev`;
- `display-update` – de 1 em 1 horas mostra o número de acontecimentos que já ocorreram.

Os valores para todas estas entradas, com excepção da opção `network`, são fixas. Se não se pretenderem esses valores devem ser feitas as alterações deles directamente no ficheiro `ini`.

Depois da secção `General` aparece a secção `Parameters` onde são definidos os valores por omissão para todos os parâmetros. Nas secções `Run` são definidos valores para alguns dos parâmetros, específicos para a corrida particular, que substituem os valores dos parâmetros definidos na secção `Parameters`. Os valores dos parâmetros definidos na secção `Parameters`, bem como os dos parâmetros das várias corridas, foram criados directamente a partir dos parâmetros normais do ficheiro `cfg`.

Repare que o nome dos ficheiros de saída não foi totalmente escolhido, apenas foi escolhido, no ficheiro `cfg`, o seu prefixo. A parte restante do nome é construída pelo programa de conversão de modo a aparecer no nome dos ficheiros o método de encaminhamento a que dizem respeito bem como a que corrida correspondem e ainda qual foi o factor de carga que foi utilizado.

## C.2.4 Exemplo de um ficheiro de configuração

```
###PARAMETROS GLOBAIS###
#parametro= duracao_chamada_1
#      valor= 1m
#parametro= duracao_chamada_2
#      valor= 1m
#parametro= duracao_chamada_3
#      valor= 1m
#parametro= duracao_chamada_4
#      valor= 1m
```

```
#parametro= factor_carga
#   valores= 1 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2
#parametro= inicio_recolha
#   valor= 24h
#parametro= fim_recolha
#   valor= 48h
#parametro= display-update
#   valor= 2h
#parametro= numero_estacoes
#   valor= 6
#parametro= numero_classes
#   valor= 4
#parametro= mostra_tudo
#   valor= 0
#parametro= corrida
#   valor= 1
#parametro= fich_topologia
#   valor= "red20000808-4.txt"
###PARAMETROS METODOS###
###DIRECTO###
###DAR###
#parametro= dar_intervalo_atualizacao
#   valor= 3m
#parametro= dar_vtrafk_factor
#   valor= 1.1
#parametro= dar_rk_factor
#   valor= 0.05
#parametro= dar_tipo_reserva_din
#   valores= 0 1
###RTNR###
#parametro= rtnr_intervalo_atualizacao
#   valor= 3m
#parametro= rtnr_vtrafk_factor
#   valor= 1.1
#parametro= rtnr_rk_factor
#   valor= 0.05
#parametro= rtnr_tipo_reserva_din
#   valores= 0 1
#Comeca aqui o verdadeiro .ini

[General]
network = rede
ini-warnings = yes
warnings = yes
snapshot-file = telefone.sna
log-parchanges = no
total-stack-kb=2048 ;2MByte -- increase if necessary
distributed=false

include meusseeds.ini

module-messages = no
verbose-simulation = no
display-update = 1h

[Parameters]
rede.duracao_chamada_1=1m
rede.duracao_chamada_2=1m
rede.duracao_chamada_3=1m
rede.duracao_chamada_4=1m
rede.inicio_recolha=24h
rede.fim_recolha=48h
rede.display-update=2h
rede.numero_estacoes=6
rede.numero_classes=4
rede.mostra_tudo=0
rede.corrida=1
rede.fich_topologia="red20000808-4.txt"
rede.central[*].dar_intervalo_atualizacao=3m
rede.central[*].dar_vtrafk_factor=1.1
rede.central[*].dar_rk_factor=0.05
rede.central[*].rtnr_intervalo_atualizacao=3m
rede.central[*].rtnr_vtrafk_factor=1.1
rede.central[*].rtnr_rk_factor=0.05
#estes vao ser substituidos nos runs
```

```
#metodo DCR
rede.central[*].dcr_tau=1s
rede.central[*].dcr_delta=1s
rede.central[*].dcr_theta=0.9
rede.central[*].dcr_intervalo_atualizacao=3m
rede.central[*].dcr_vtrafk_factor=1.1
rede.central[*].dcr_rk_factor=0.05
rede.central[*].dcr_tipo_reserva_din=0
#metodo DAR
rede.central[*].dar_intervalo_atualizacao=3m
rede.central[*].dar_vtrafk_factor=1.1
rede.central[*].dar_rk_factor=0.05
rede.central[*].dar_tipo_reserva_din=0
#metodo RTNR
rede.central[*].rtnr_intervalo_atualizacao=3m
rede.central[*].rtnr_vtrafk_factor=1.1
rede.central[*].rtnr_rk_factor=0.05
rede.central[*].rtnr_tipo_reserva_din=0
#metodo FAR
rede.central[*].far_fich_input_tabelas_routing="mitra_far%d.txt"
rede.central[*].far_ncaminhos=1

[Run 1]
rede.factor_carga=1.0
rede.tipo_routing="DIRECTO"
output-scalar-file="mult-4--1.0-DIRECTO.sca"
rede.central[*].fich_output_folha_calculo="mult-4--1.0-DIRECTO.txt"

[Run 2]
rede.factor_carga=1.1
rede.tipo_routing="DIRECTO"
output-scalar-file="mult-4--1.1-DIRECTO.sca"
rede.central[*].fich_output_folha_calculo="mult-4--1.1-DIRECTO.txt"

[Run 3]
rede.factor_carga=1.2
rede.tipo_routing="DIRECTO"
output-scalar-file="mult-4--1.2-DIRECTO.sca"
rede.central[*].fich_output_folha_calculo="mult-4--1.2-DIRECTO.txt"

[Run 4]
rede.factor_carga=1.3
rede.tipo_routing="DIRECTO"
output-scalar-file="mult-4--1.3-DIRECTO.sca"
rede.central[*].fich_output_folha_calculo="mult-4--1.3-DIRECTO.txt"

[Run 5]
rede.factor_carga=1.4
rede.tipo_routing="DIRECTO"
output-scalar-file="mult-4--1.4-DIRECTO.sca"
rede.central[*].fich_output_folha_calculo="mult-4--1.4-DIRECTO.txt"

. . .

[Run 10]
rede.factor_carga=1.9
rede.tipo_routing="DIRECTO"
output-scalar-file="mult-4--1.9-DIRECTO.sca"
rede.central[*].fich_output_folha_calculo="mult-4--1.9-DIRECTO.txt"

[Run 11]
rede.factor_carga=2.0
rede.tipo_routing="DIRECTO"
output-scalar-file="mult-4--2.0-DIRECTO.sca"
rede.central[*].fich_output_folha_calculo="mult-4--2.0-DIRECTO.txt"

[Run 12]
rede.factor_carga=1.0
rede.tipo_routing="DAR"
rede.central[*].dar_tipo_reserva_din=0
output-scalar-file="mult-4--1.0-DAR0.sca"
rede.central[*].fich_output_folha_calculo="mult-4--1.0-DAR0.txt"

[Run 13]
rede.factor_carga=1.0
rede.tipo_routing="DAR"
```

```
rede.central[*].dar_tipo_reserva_din=1
output-scalar-file="mult-4--1.0-DAR1.sca"
rede.central[*].fich_output_folha_calculo="mult-4--1.0-DAR1.txt"

[Run 14]
rede.factor_carga=1.1
rede.tipo_routing="DAR"
rede.central[*].dar_tipo_reserva_din=0
output-scalar-file="mult-4--1.1-DAR0.sca"
rede.central[*].fich_output_folha_calculo="mult-4--1.1-DAR0.txt"

[Run 15]
rede.factor_carga=1.1
rede.tipo_routing="DAR"
rede.central[*].dar_tipo_reserva_din=1
output-scalar-file="mult-4--1.1-DAR1.sca"
rede.central[*].fich_output_folha_calculo="mult-4--1.1-DAR1.txt"

...

[Run 32]
rede.factor_carga=2.0
rede.tipo_routing="DAR"
rede.central[*].dar_tipo_reserva_din=0
output-scalar-file="mult-4--2.0-DAR0.sca"
rede.central[*].fich_output_folha_calculo="mult-4--2.0-DAR0.txt"

[Run 33]
rede.factor_carga=2.0
rede.tipo_routing="DAR"
rede.central[*].dar_tipo_reserva_din=1
output-scalar-file="mult-4--2.0-DAR1.sca"
rede.central[*].fich_output_folha_calculo="mult-4--2.0-DAR1.txt"

[Run 34]
rede.factor_carga=1.0
rede.tipo_routing="RTNR"
rede.central[*].rtnr_tipo_reserva_din=0
output-scalar-file="mult-4--1.0-RTNR0.sca"
rede.central[*].fich_output_folha_calculo="mult-4--1.0-RTNR0.txt"

[Run 35]
rede.factor_carga=1.0
rede.tipo_routing="RTNR"
rede.central[*].rtnr_tipo_reserva_din=1
output-scalar-file="mult-4--1.0-RTNR1.sca"
rede.central[*].fich_output_folha_calculo="mult-4--1.0-RTNR1.txt"

...

[Run 54]
rede.factor_carga=2.0
rede.tipo_routing="RTNR"
rede.central[*].rtnr_tipo_reserva_din=0
output-scalar-file="mult-4--2.0-RTNR0.sca"
rede.central[*].fich_output_folha_calculo="mult-4--2.0-RTNR0.txt"

[Run 55]
rede.factor_carga=2.0
rede.tipo_routing="RTNR"
rede.central[*].rtnr_tipo_reserva_din=1
output-scalar-file="mult-4--2.0-RTNR1.sca"
rede.central[*].fich_output_folha_calculo="mult-4--2.0-RTNR1.txt"
```

Fig. C-3 – Conteúdo de um ficheiro de configuração (*ini*).

Este exemplo representa uma parte do ficheiro *ini* que se obteve a partir do exemplo do ficheiro *cfg* apresentado na secção anterior. Neste exemplo apenas foram retirados algumas das corridas. Com este ficheiro *ini* são executadas 55 corridas sendo o método de encaminhamento utilizado em cada uma dessas corridas o método de encaminhamento DIRECTO, DAR ou o RTNR.

## C.3 Outros ficheiros de topologia utilizados na simulação

Esta secção contém os ficheiros de topologia utilizados na simulação, com a excepção do apresentado como exemplo na primeira secção deste apêndice, que representa o sistema 3 descrito no capítulo 4.

### C.3.1 Ficheiro que representa o Sistema 1

As seguintes matrizes correspondem ao conteúdo do ficheiro de topologia utilizado na simulação dos métodos de encaminhamento com apenas uma classe de tráfego, sendo esse tráfego monocal. Este ficheiro de topologia representa o sistema 1 descrito no capítulo 4. É especificado também neste ficheiro o número de circuitos reservados em cada feixe. Aqui o número de circuitos reservados para tráfego de primeiro encaminhamento foi de 4 para todos os feixes. A descrição do conteúdo deste ficheiro foi feita na primeira secção deste apêndice.

```

36      24      324      48      48
        96      96      108      96
                12      48      24
                        192      84
                                336

0 1 1

0      13.735  3.485  128.905  10.235  14.555
13.735  0      12.555  50.805  38.39  41.28
3.485   12.555  0      5.96   3.43   6.625
128.905 50.805  5.96   0      39.71  41.5
10.235  38.39  3.43   39.71  0      63.555
14.555  41.28  6.625  41.5  63.555  0

4      4      4      4      4
        4      4      4      4
                4      4      4
                        4      4
                                4

0      18      12      162      24      24
18      0      48      48      54      48
12      48      0      6      24      12
162     48      6      0      96      48
24      54      24      96      0      168
24      48      12      48      168      0

```

Fig. C-4 – Conteúdo do ficheiro de topologia correspondente ao sistema 1.

### C.3.2 Ficheiro que representa o Sistema 2

As matrizes que seguem correspondem ao conteúdo do ficheiro de topologia utilizado na simulação dos métodos de encaminhamento com duas classes de tráfego sendo tráfego de Poisson para ambas mas monocal para uma e multicanal para outra. Este ficheiro de topologia representa o sistema 2 descrito no capítulo 4. O número fixo de circuitos reservados para tráfego de primeiro encaminhamento, em cada feixe, foi 0 para todos os feixes, para ambas as classes. É também especificado neste ficheiro as

capacidades garantidas nos feixes, se necessárias, para encaminhamento directo para cada uma das classes de tráfego. Para uma descrição detalhada do conteúdo deste ficheiro ver a primeira secção deste apêndice.

47	24	570	48	68	
	96	182	161	167	
		15	48	29	
			214	151	
				464	
0	1	1			
0	13.735	3.485	128.905	10.235	14.555
13.735	0	12.555	50.805	38.39	41.28
3.485	12.555	0	5.96	3.43	6.625
128.905	50.805	5.96	0	39.71	41.5
10.235	38.39	3.43	39.71	0	63.555
14.555	41.28	6.625	41.5	63.555	0
0	0	0	0	0	
	0	0	0	0	
		0	0	0	
			0	0	
				0	
0	17	8	203	18	24
18	0	34	65	56	59
8	34	0	5	18	10
203	65	6	0	77	53
18	57	18	77	0	166
24	60	11	54	166	0
0	2	1			
0	2.747	0.697	25.781	2.047	2.911
2.747	0	2.511	10.161	7.678	8.256
0.697	2.511	0	1.192	0.686	1.325
25.781	10.161	1.192	0	7.942	8.3
2.047	7.678	0.686	7.942	0	12.711
2.911	8.256	1.325	8.3	12.711	0
0	0	0	0	0	
	0	0	0	0	
		0	0	0	
			0	0	
				0	
0	6	4	82	6	10
6	0	14	26	24	24
4	14	0	2	6	4
82	26	2	0	30	22
6	24	6	30	0	66
10	24	4	22	66	0

Fig. C-5 – Conteúdo do ficheiro de topologia correspondente ao sistema 2.

### C.3.3 Alterações feitas no ficheiro de topologia que representa o sistema 3

O ficheiro de topologia que representa o sistema 3, descrito no capítulo 4, foi apresentado como exemplo na secção C.1. A partir deste ficheiro de topologia foi criado um segundo ficheiro que difere deste apenas nas matrizes que representam as capacidades mínimas garantidas para encaminhamento directo para a classe de tráfego correspondente. Por esta razão vamos apresentar aqui apenas essas

matrizes, começando com a matriz correspondente à classe com menor necessidade de recursos e terminando com a matriz correspondente à classe que tem maior necessidade de recursos.

0	10	8	181	9	9
11	0	34	65	56	59
8	34	0	5	18	10
181	65	6	0	77	53
9	57	18	77	0	166
10	60	11	54	166	0
0	6	2	70	6	6
6	0	14	26	24	24
2	14	0	2	6	4
70	26	2	0	30	22
6	24	6	30	0	66
6	24	4	22	66	0
0	24	12	294	24	36
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	24	0	192	24	48
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0

Fig. C-6 – Alterações ao conteúdo do ficheiro de topologia correspondente ao sistema 3.

O sistema 3 foi simulado com os dois ficheiros de topologia referidos.



# Apêndice D

## Estudo Comparativo da Congestão através de Método Analítico e Simulação

### D.1 Cálculo da Congestão no Tempo através da fórmula de Engset

Vamos calcular a congestão no tempo através da fórmula de Engset, sendo conhecida a intensidade de chamadas por fonte quando a fonte está livre.

A fórmula de Engset é:

$$E_{n,N}(\beta) = \frac{\binom{N}{n} \times \beta^n}{\sum_{r=0}^n \binom{N}{r} \times \beta^r} \quad (\text{D}\cdot 1)$$

para  $N > n$  e  $n = 1, 2, 3, \dots$ ; onde  $N$  é o número de fontes,  $n$  o número de circuitos no feixe e  $\beta$  a intensidade de chamadas por fonte quando a fonte está livre, e sendo o tempo de ocupação médio a unidade de tempo.  $E_{n,N}(\beta)$  é a probabilidade de encontrar todos os circuitos ocupados num instante de tempo escolhido aleatoriamente.

A fórmula anterior pode ser transformada na fórmula recursiva seguinte [Olsson75]:

$$\begin{cases} E_{n,N}(\beta) = \frac{(N-n+1) \times \beta \times E_{n-1,N}(\beta)}{n + (N-n+1) \times \beta \times E_{n-1,N}(\beta)} \\ E_{0,N}(\beta) = 1 \end{cases} \quad (\text{D}\cdot 2)$$

A recursão é feita relativamente ao número de circuitos  $n$  (começando com  $n = 0$ ) sendo  $n$  o número de recursões.

Como já referido no capítulo 4 (secção 4.1.1.2) uma característica nos modelos de população finita é a diferença entre a probabilidade de bloqueio vista por uma chamada recém chegada (congestão de chamadas -  $B$ ) e probabilidade do feixe ter um número de circuitos inferior aos necessários por uma chamada (congestão no tempo -  $E$ ). Pois, para populações finitas, uma chegada implica que menos uma fonte esteja disponível para gerar chamadas. E por isso após a primeira chegada a população livre é reduzida de uma unidade passando a ser igual a  $N - 1$ . Para calcular a congestão de chamadas, num sistema em que as chamadas são geradas por uma população finita de  $N$  fontes, basta calcular a congestão no tempo de um modelo em que a população tem  $N - 1$  fontes.

Temos então:

$$B(N) = E(N - 1) \quad (D.3)$$

## D.2 Resultados analíticos

A tabela seguinte apresenta os valores da congestão obtidos pela fórmula de Engset quando o número de fontes é 34, para vários valores de capacidades e intensidades de tráfego.

<i>Capacidades</i> \ <i>Intensidades</i>	10	15	20	25	30
1	62,8%	35,6%	9,2%	0,3%	0,0%
2	80,6%	64,5%	41,1%	12,4%	0,3%
3	86,9%	75,8%	58,5%	30,9%	3,3%
4	90,1%	81,7%	68,2%	44,7%	9,9%
5	92,1%	85,3%	74,3%	54,3%	17,9%

Tab. D.1 - Congestão prevista por Engset para 34 fontes.

A tabela seguinte apresenta os valores da congestão, também obtidos pela fórmula de Engset, quando o número de fontes é 69, para vários valores de capacidades e intensidades de tráfego.

<i>Capacidades</i> \ <i>Intensidades</i>	10	15	20	25	30
1	83,7%	73,5%	61,6%	47,7%	31,9%
2	91,8%	86,5%	80,3%	72,8%	63,6%
3	94,5%	91,0%	86,8%	81,7%	75,4%
4	95,9%	93,2%	90,1%	86,2%	81,5%
5	96,7%	94,6%	92,0%	89,0%	85,1%

Tab. D.2 - Congestão prevista por Engset para 69 fontes.

## D.3 Resultados da simulação

A tabela seguinte apresenta os valores dos bloqueios obtidos por simulação quando o número de fontes é 35, para vários valores de capacidades e intensidades de tráfego.

<i>Capacidades</i> <i>Intensidades</i>	10	15	20	25	30
1	63,2%	35,7%	9,5%	0,3%	0,0%
2	80,8%	64,4%	41,6%	12,8%	0,3%
3	86,9%	75,7%	58,3%	30,9%	3,2%
4	90,1%	81,6%	68,1%	44,9%	9,5%
5	92,1%	85,3%	74,4%	54,5%	18,3%

Tab. D-3 - Bloqueios obtidos por simulação para 35 fontes.

A tabela seguinte apresenta os valores dos bloqueios, também obtidos por simulação, quando o número de fontes é 70, para vários valores de capacidades e intensidades de tráfego.

<i>Capacidades</i> <i>Intensidades</i>	10	15	20	25	30
1	83,8%	73,8%	61,2%	47,7%	32,0%
2	91,7%	86,4%	80,4%	73,0%	63,8%
3	94,5%	91,0%	86,7%	81,7%	75,4%
4	95,8%	93,2%	89,9%	86,4%	81,3%
5	96,7%	94,5%	92,0%	89,1%	85,1%

Tab. D-4 - Bloqueios obtidos por simulação para 70 fontes.

## D.4 Conclusões

Na tabela D-5 encontram-se os valores das diferenças normalizadas entre os valores obtidos pela fórmula de Engset e por simulação no caso de 34 fontes (35 na simulação). Os valores das diferenças apresentadas encontram-se normalizados através da divisão pelo maior valor.

<i>Capacidades</i> <i>Intensidades</i>	10	15	20	25	30
1	0,7%	0,3%	3,3%	11,4%	*
2	0,2%	0,2%	1,2%	2,8%	3,6%
3	0,1%	0,1%	0,3%	0,3%	1,8%
4	0,0%	0,1%	0,2%	0,4%	3,7%
5	0,1%	0,0%	0,1%	0,4%	1,9%

Tab. D-5 - Diferenças normalizadas, no caso de 35 fontes, entre a fórmula de Engset e a simulação.

Na tabela D-6 encontram-se os valores das diferenças normalizadas entre os valores obtidos pela fórmula de Engset e por simulação no caso de 69 fontes (70 na simulação). Os valores das diferenças apresentadas encontram-se também normalizados através da divisão pelo maior valor.

<i>Capacidades</i> <i>Intensidades</i>	10	15	20	25	30
1	0,1%	0,4%	0,6%	0,0%	0,3%
2	0,1%	0,1%	0,0%	0,2%	0,3%
3	0,0%	0,0%	0,1%	0,0%	0,1%
4	0,0%	0,0%	0,1%	0,2%	0,3%
5	0,0%	0,1%	0,0%	0,1%	0,0%

Tab. D-6 - Diferenças normalizadas, no caso de 70 fontes, entre a fórmula de Engset e a simulação.

Note-se que quando as dimensões das percentagens são muito pequenas as diferenças normalizadas aparecem com valores elevados embora existam diferenças objectivamente pequenas entre essas percentagens. No geral pode concluir-se que existe grande similaridade entre os valores obtidos por simulação e os valores calculados, como era esperado.

# Apêndice E

## Estrutura do Programa de Simulação Implementado

### E.1 Componentes gerais para todos os encaminhamentos

Na figura E-1 apresenta-se um diagrama com as principais classes do modelo de simulação que foi construído. O diagrama apresentado foi construído já ao nível da implementação. Nele podem ver-se os membros de cada uma das classes. As classes que correspondem aos componentes controlador, encaminhador e estatísticas são respectivamente as classes com os nomes `controlador`, `router` e `stats`. Também fazem parte do diagrama as classes que correspondem ao processo gerador e ao processo central.

Em relação a este diagrama podem ser feitas várias observações que a seguir passamos a descrever. As classes possuem membros que são privados, outros que são protegidos e outros que são públicos. Isso permite definir diferentes níveis de acessibilidade a um membro. Tanto a classe gerador como a classe central são classes derivadas da classe `cSimpleModule`. Esta derivação já era esperada pois, como já foi dito, todas as classes correspondentes a módulos simples, que sejam definidos no modelo, têm que ser derivadas da classe `cSimpleModule` (classe definida na biblioteca de classes de simulação do OMNeT++).

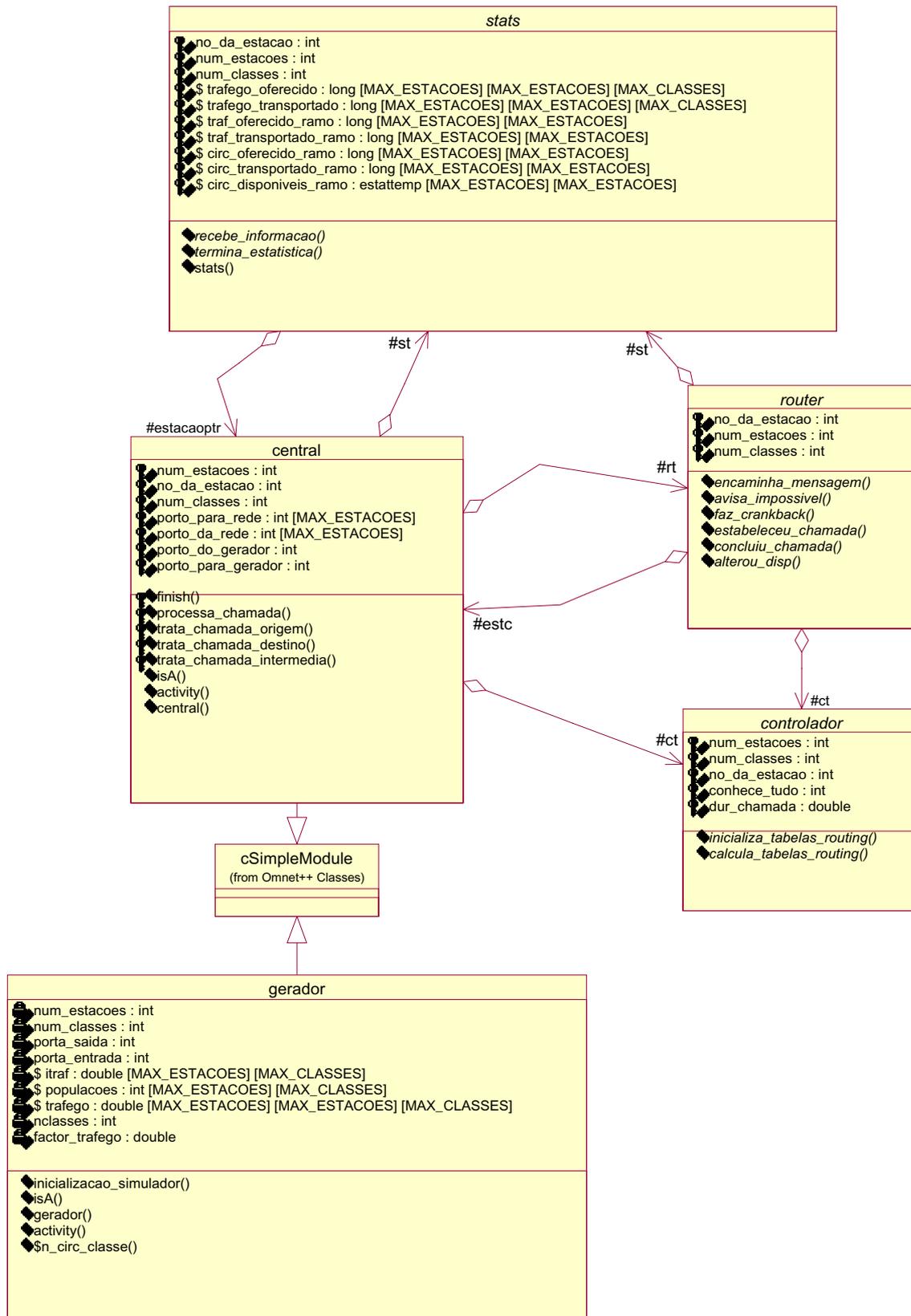


Fig. E-1 – Diagrama das classes principais do modelo.

A classe `central` tem membros que são ponteiros para objectos do tipo `router`, `controlador` e `stats` com os nomes `rt`, `ct` e `st` respectivamente, para aceder aos métodos dessas classes. A classe `router` tem membros que são ponteiros para objectos do tipo `controlador` e `stats` com os nomes, `ct` e `st` respectivamente, pela mesma razão. A classe `stats` tem um membro que é um ponteiro para um objecto do tipo `central` com o nome `estacaoptr` e a classe `router` um com o nome `estc` também para a `central`. Estes últimos ponteiros permitem a estas classes aceder à informação armazenada no objecto `central`.

Ainda em relação aos membros das várias classes vão-se de seguida referir quais os valores principais, armazenados por cada um dos objectos de cada um dos tipos. Assim todos os objectos do tipo `central`, `router`, `controlador` e `stats` armazenam o número de centrais existentes no modelo, o número da `central` no modelo ao qual o objecto está associado e o número de classes de tráfego que coexistem no modelo nos membros com os nomes `num_estacoes`, `no_da_estacao` e `num_classes` respectivamente. Os objectos do tipo gerador armazenam o número de centrais existentes no modelo e qual o número de classes de tráfego que coexistem no modelo, nos membros com os nomes `num_estacoes` e `num_classes` respectivamente.

A classe `central` tem várias funções membro, para já vamos falar apenas da função `activity`, da função `finish` e da função `isA`. A função `activity` é uma função virtual pura da classe `cSimpleModule`. Por essa razão essa função tem que ser obrigatoriamente definida na classe `central`. É o código das funções `activity` dos vários processos do modelo que são executados, em paralelo, desde o início da simulação até ao fim da simulação. É portanto na função `activity` da classe `central` que está definido o código que corresponde a todas as operações que cada `central` deve realizar durante a simulação. A função `finish` é uma função da classe `cModule`, é executada quando a simulação termina. Nesta função foi chamada a função membro `termina_estatisticas` da classe `stats` que definiremos na subsecção seguinte. A função `isA`, é uma função da classe `cObject`, e foi redefinida com o objectivo de atribuir o nome ao objecto `central` (usado pelas funções internas do simulador). As restantes funções membro da classe `central` são funções que são solicitadas pela função `activity` da classe `central`, na implementação do funcionamento desta. Como o funcionamento do processo `central` foi descrito no capítulo 5, secção 5.5.2.1, não são aqui definidas as outras funções desse processo.

Pela mesma razão que se redefiniu a função `activity` na classe `central` também é necessário fazê-lo na classe gerador. Na função `activity` da classe gerador está definido o código que corresponde a todas as operações que cada gerador deve realizar durante a simulação. As operações a serem realizadas pelo gerador foram descritas também no capítulo 5, na secção 5.5.2.2.

## E.1.1 Componentes base

### E.1.1.1 Controlador

O controlador fornece as funções necessárias para inicializar e calcular as tabelas de encaminhamento. Os métodos da classe controlador são, além dos construtores, duas funções virtuais puras designadas por `inicializa_tabelas_routing` e `calcula_tabelas_routing` como pode ser observado no diagrama da figura E-1.

A função `inicializa_tabelas_routing` é chamada no início da simulação, pelo processo central, deve por isso inicializar os parâmetros de controle para a central ou para as centrais. A função `calcula_tabelas_routing` é chamada durante toda a simulação, pelo processo central, com intervalos mais ou menos longos, dependendo da frequência com que é necessário actualizar a informação a utilizar no encaminhamento das chamadas.

### E.1.1.2 Encaminhador

O encaminhador (classe `router`) fornece as funções de encaminhar uma chamada, efectuar registos quando foi impossível encaminhar a chamada (quando esta se encontrava já numa central intermédia), e indicar se a chamada deve ou não retornar por *crankback*, através das funções membro com os nomes respectivamente `encaminha_mensagem`, `avisa_impossivel` e `faz_crankback` como pode ser observado no diagrama da figura E-1. Estas três funções são virtuais puras o que faz com que seja obrigatório redefini-las em todas as classes que sejam derivadas a partir desta classe.

Quando surge uma chamada, do tipo chamada a estabelecer, na central origem esta chama a função `encaminha_mensagem` e espera receber a identificação da central para onde deve enviar a chamada; se a identificação que recebeu não corresponder a uma central válida então é porque o encaminhador não conseguiu determinar uma central possível. A função `encaminha_mensagem` deve então determinar, com a informação que recebe (informação da chamada que se pretende encaminhar), a central para onde a chamada deve seguir de modo a chegar ao destino. A função pode também determinar o caminho completo que a chamada deve seguir até à central destino. A função deve devolver o caminho vazio como o caminho completo se não calcular o caminho completo e deve devolver o número `-1` como sendo o número da próxima central para onde a chamada deva ser enviada se não conseguiu determinar nenhuma central para onde seja possível enviar a chamada de modo a que ela chegue ao destino.

A função `encaminha_mensagem` pode ser chamada pela central onde a chamada teve origem, como já referido, mas pode também ser chamada por uma central intermédia que seja usada pela chamada no seu encaminhamento. Esta segunda possibilidade ocorre quando a chamada chega a uma central intermédia

sem conhecer o caminho que deve seguir, isto é, não sabe para que central deve seguir para chegar ao destino.

Quando uma chamada, do tipo chamada a estabelecer, se encontra numa central intermédia e se determine (usando a função `encaminha_mensagem` ou um caminho já presente na mensagem) não ser possível obter uma central para onde enviar a chamada é, nesta situação que a central intermédia chama a função `faz_crankback` para saber se a chamada pode ou não tentar um caminho alternativo. Esta função deve então determinar se é permitido à chamada que se pretende encaminhar tentar outro caminho.

A função `avisa_impossivel` é chamada quando se está a tentar encaminhar uma chamada e se determine que não é possível encaminhar a chamada pelo caminho que se estava a seguir. A invocação da função `avisa_impossivel` ocorre, na situação anterior, quando a chamada que se tentava encaminhar se encontra de novo na central origem. Esta função pode ser utilizada quando se pretender efectuar alguma operação no caso de uma chamada que foi bloqueada o ter sido quando já se encontrava numa central intermédia.

Além das três funções membro já referidas a classe `router` possui ainda outras três funções membro que também são virtuais puras. Estas funções têm os nomes `estabeleceu_chamada`, `concluiu_chamada` e `alterou_disp` como pode também ser observado no diagrama da figura E.1. A função `estabeleceu_chamada` é invocada, pela central destino, quando uma chamada atingiu o seu destino. Esta função pode ser definida por exemplo para contar o número de chamadas estabelecidas para cada uma das classes de tráfego entre cada par de centrais (origem-destino). A função `concluiu_chamada` é invocada, pela central origem, quando uma chamada a atingiu depois de ter sido estabelecida com sucesso. Esta função é chamada apenas quando a chamada regressou à central origem porque só quando isso aconteceu é que a capacidade ocupada nos feixes foi completamente libertada. A função pode ser definida por exemplo para contar o número de chamadas concluídas para cada uma das classes de tráfego entre cada par de centrais (origem-destino). Por último a função `alterou_disp` é invocada, pela central em que a chamada se encontrar, depois desta ter incrementado ou decrementado a capacidade disponível de um feixe. Operações de incrementação da capacidade de um feixe podem ocorrer tanto numa central destino como numa central intermédia. E operações de decrementação da capacidade de um feixe podem ocorrer tanto numa central origem como numa central intermédia. Por isso a função `alterou_disp` pode ser chamada pela central origem, pela central intermédia e pela central destino. Esta função pode ser definida com operações que devam ser realizadas quando ocorra a alteração da capacidade disponível de um feixe.

### E.1.1.3 Estatísticas

Os métodos da classe `stats` são, além dos construtores, duas funções virtuais puras designadas por `recebe_informacao` e `termina_estatistica` como pode ser observado no diagrama da figura E.1. A função `recebe_informacao` é chamada pelo processo central sempre que ocorre uma das acções seguintes:

- Início de uma chamada (sendo a função chamada pela central origem);
- Uma chamada foi concluída com sucesso (sendo a função chamada pela central destino);
- Uma chamada foi concluída sem sucesso (sendo a função chamada pela central origem);
- Uma chamada vai tentar um caminho alternativo após ter ocorrido *crankback* (sendo a função chamada pela central origem);
- Uma chamada foi concluída sem sucesso depois de ter ocorrido *crankback* (sendo a função chamada pela central origem);
- Ocupação de circuitos de um feixe (sendo a função chamada pela central origem e pela central intermédia);
- Libertação de circuitos de um feixe (sendo a função chamada pela central intermédia e pela central destino);
- Um feixe foi tentado sem sucesso (sendo a função chamada pelo `router` associado à central).

A função `recebe_informacao` pode também ser utilizada para actualizar outros contadores.

A função `termina_estatistica` é chamada pela função `finish` do processo central, como já foi referido. Esta função pode ser utilizada para registar os resultados da simulação em ficheiros.

## E.1.2 Outros componentes

Nos dois diagramas das figuras seguintes, figura E.2 e figura E.3, estão representadas as classes auxiliares que foram utilizadas no modelo. Neles podem ver-se os membros de cada uma dessas classes. As classes que correspondem aos componentes caminhos e armazém de dados são respectivamente as classes com os nomes `caminhos` e `arm_dados` e estão representadas na figura E.2. As classes que correspondem aos componentes reserva e tráfego são respectivamente as classes com os nomes `rr` e `classe_trafego` e estão representadas na figura E.3. Repare que nessa figura aparece também um tipo de dados designado por `MPTR` (usado pelo processo gerador na declaração da variável que armazena cópias das mensagens, do tipo “chamada a estabelecer”, para os tráfegos de população finita) que é definido como um ponteiro para `cMessage`.

### E.1.2.1 Caminhos

A classe `caminhos` tem um membro chamado `no_estacoes` onde armazena o número de centrais no modelo e outro membro chamado `buffer` onde pode armazenar um caminho de forma descodificada (uma sequência de números de centrais separados pelo carácter “-”, em que o primeiro número corresponde à central origem, o último número à central destino e os números intermédios às centrais intermédias do caminho, desde a origem até ao destino). Os métodos da classe `caminhos` são, além dos construtores, as funções designadas por `novo_caminho`, `acrescenta_caminho`, `decrece_caminho`, `ultima_estacao`, `cria_caminho`, `vazio` e `escreve_caminho` como pode ser observado no diagrama da figura E-2. Cada uma das funções faz uma determinada operação sobre um caminho. O que cada uma faz especificamente é: a função `novo_caminho` devolve um caminho vazio, a função `acrescenta_caminho` acrescenta uma nova central a um caminho e devolve o novo caminho, a função `decrece_caminho` retira uma central de um caminho e devolve o novo caminho e também a central que foi obtida do caminho, a função `ultima_estacao` obtém a última central de um caminho mas mantém o caminho inalterado, a função `cria_caminho` cria um caminho com uma central apenas, a função `vazio` devolve o valor verdadeiro se o caminho que recebe é um caminho vazio e o valor falso no caso contrário, a função `escreve_caminho` devolve um ponteiro para uma cadeia de caracteres que correspondem a um caminho na forma descodificada (uma sequência de números de centrais separados pelo carácter “-”, em que o primeiro número corresponde à central origem, o último número à central destino e os números intermédios às centrais intermédias do caminho, desde a origem até ao destino).

### E.1.2.2 Armazém de dados

Na classe `arm_dados` todas as variáveis internas são estáticas bem como as funções membro (métodos). Tal significa que todas as variáveis desta classe são únicas e existem sem a classe ser instanciada. Os métodos são estáticos porque apenas precisam de modificar membros estáticos. As afirmações anteriores significam que não houve necessidade de criar nenhuma instância desta classe.

As variáveis desta classe são quatro sendo cada uma delas uma tabela. A variável `disp` e a variável `cap` são tabelas bidimensionais que são usadas para armazenar o número de circuitos disponíveis e o número de circuitos existentes no feixe entre quaisquer dois pares de centrais, respectivamente. A variável `reserva` e a variável `tplanclasse` são tabelas tridimensionais que são usadas para armazenar o número de circuitos reservados e o número de circuitos planeados nos feixes entre quaisquer dois pares de centrais para cada uma das classes, respectivamente. O número de circuitos planeados para uma classe em cada feixe ou por outras palavras a largura de banda mínima garantida para uma classe quando existe bloqueio é uma fracção dos circuitos existentes no feixe que é atribuída a essa classe quando as condições de bloqueio, das chamadas dessa classe, o justificam.

As funções `inc_disp` e `dec_disp` incrementam e decrementam, de um dado valor, a capacidade disponível do feixe entre duas dadas centrais, respectivamente. A função `set_disp` armazena um dado valor na variável `disp` e a função `obt_disp` dá-nos o valor armazenado na variável `disp`, este valor corresponde à capacidade disponível do feixe entre duas dadas centrais. A função `set_cap` armazena um dado valor na variável `cap` e a função `obt_cap` dá-nos o valor armazenado na variável `cap`, corresponde este valor à capacidade do feixe entre duas dadas centrais. A função `set_reserva` armazena um dado valor na variável `reserva` e a função `obt_reserva` dá-nos o valor armazenado na variável `reserva`, o valor armazenado corresponde aos circuitos reservados no feixe entre duas dadas centrais. Por último a função `set_tplanclass` armazena um dado valor na variável `tplanclass` e a função `obt_tplanclass` dá-nos o valor armazenado na variável `tplanclass`, o valor armazenado nesta variável corresponde aos circuitos planeados no feixe entre duas dadas centrais.

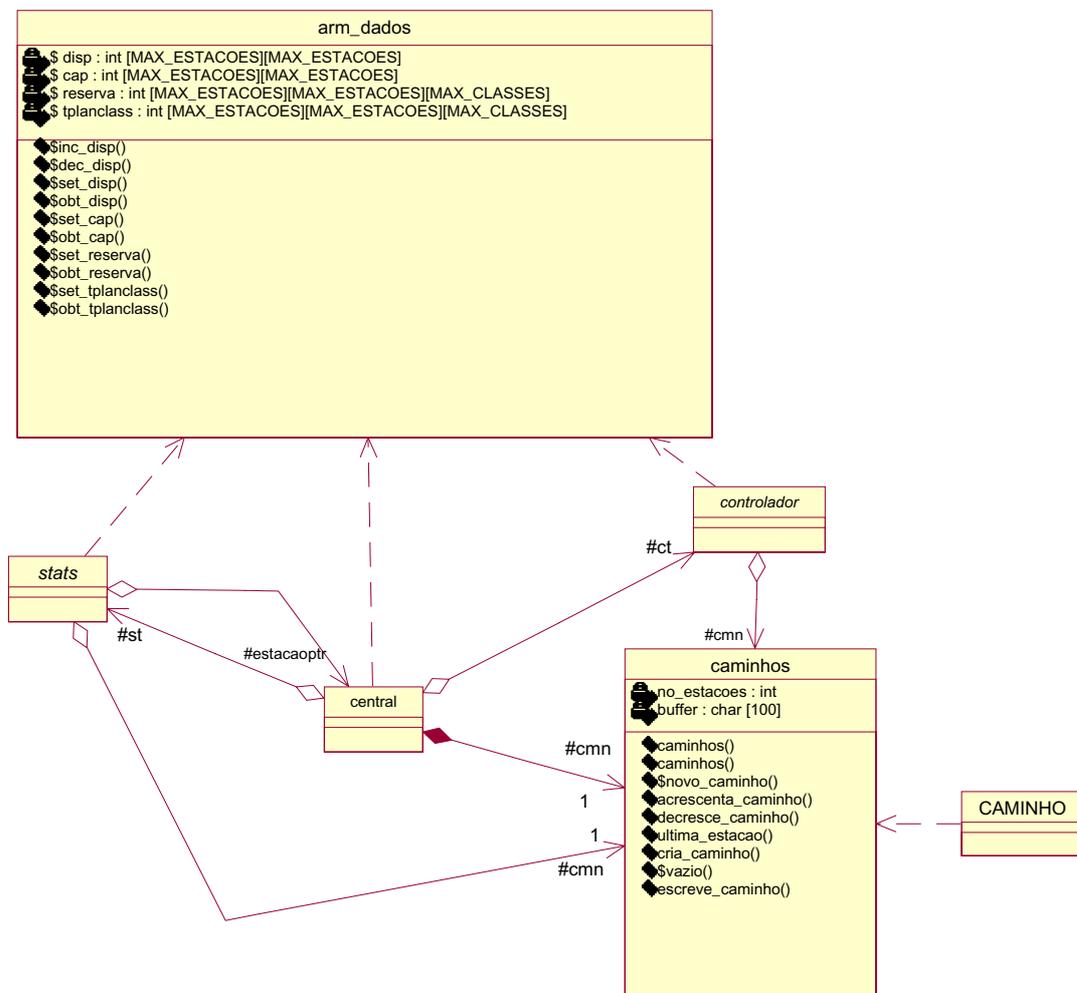


Fig. E-2 – Diagrama das classes auxiliares do modelo, usadas pelo processo central.

### E.1.2.3 Tráfego

Ao componente tráfego corresponde uma estrutura chamada `classe_trafego` que foi renomeada para CTRAF. Esta estrutura possui três campos o que lhe permite armazenar, para cada classe de tráfego, o número de circuitos necessários às chamadas, em cada feixe, o tipo de duração (tempo de serviço) das chamadas e a indicação de se o tráfego é ou não de população finita.

### E.1.2.4 Reserva de números aleatórios

A classe que corresponde a este componente, classe `rr`, apenas tem membros estáticos daí que pela mesma razão que foi indicada para a classe `arm_dados` também esta classe não foi instanciada. As variáveis internas desta classe `treserv`, `resrv`, `sresrv` e `npop` armazenam respectivamente o número total de geradores de números aleatórios reservados, o número de geradores de números aleatórios reservados para cada origem (por classe em cada origem), o número do primeiro gerador de números aleatórios reservado para cada origem (por classe em cada origem) e o número de fontes por classe para cada origem (no caso de tráfego de população finita). As duas funções desta classe são `reserva_rand` e `escolhe_rand`. A função `reserva_rand` (chamada pelo processo gerador na inicialização) é a responsável pela atribuição (inicialização) dos valores correctos às variáveis internas da classe. A função `escolhe_rand` (chamada pelo processo gerador na geração do tráfego) dá-nos o índice do gerador de números aleatórios que deve ser usado numa dada situação do processo de geração de uma chamada de uma dada classe entre um dado par origem/destino.

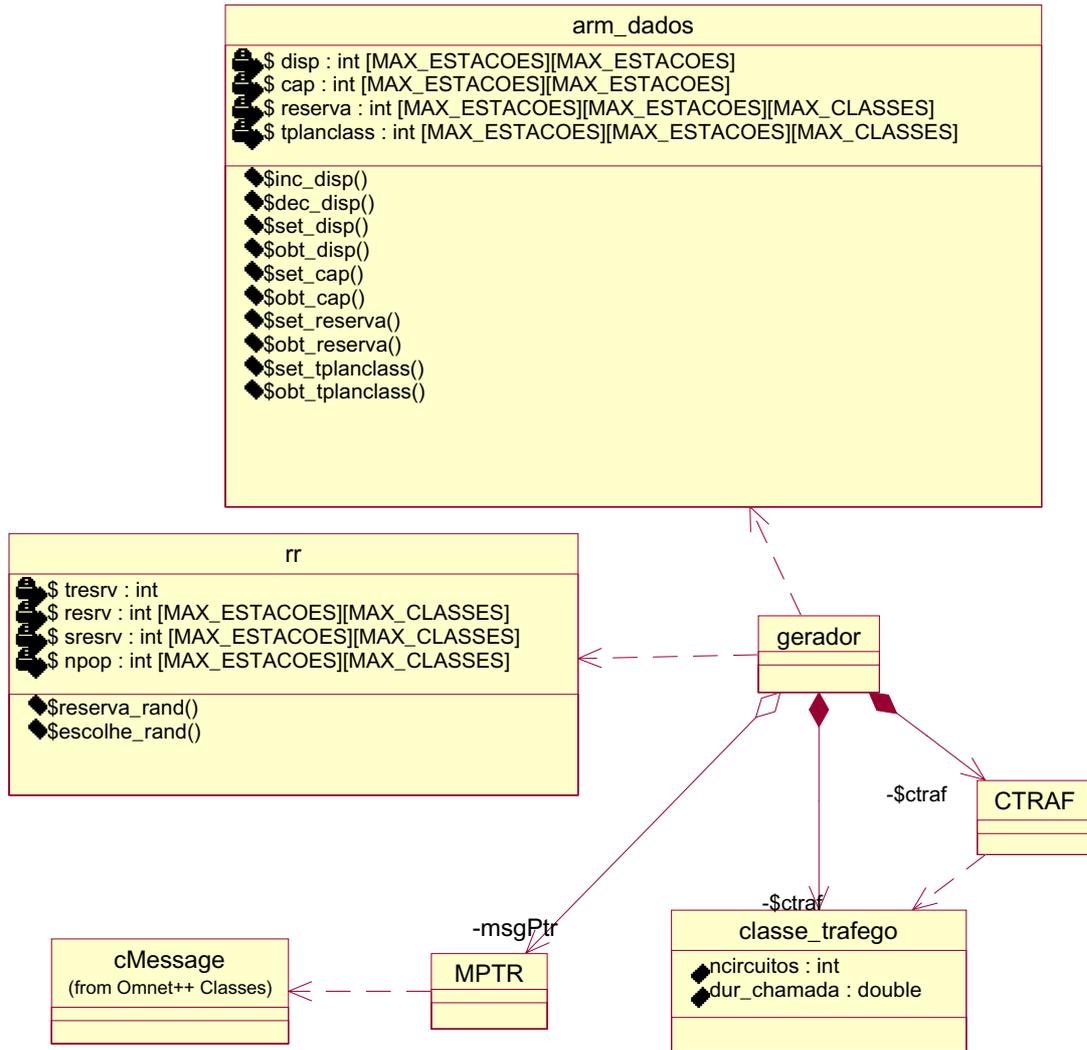


Fig. E-3 – Diagrama das classes auxiliares do modelo, usadas pelo processo gerador.

## E.2 Componentes específicos de cada encaminhamento

### E.2.1 Estatísticas globais

Antes de falar dos componentes específicos de cada um dos métodos de encaminhamento vamos referir um componente que é comum aos vários métodos implementados, mas que poderia ser específico do método de encaminhamento a implementar. Por esta razão é que não foi introduzido na subsecção anterior. Ao referido componente está associada a classe `stats_glob` que é derivada da classe `stats` e portanto herda desta algumas características (observar a figura E-4). Como descrito anteriormente a classe `stats` possui duas funções chamadas `recebe_informacao` e `termina_estatisticas` que são virtuais puras e que, por isto, tem que ser implementadas na classe `stats_glob`.

Observe, na figura E-4 que além das classes já referidas está representada uma classe com o nome `estattemp`, que é uma classe com funções estatísticas. Esta classe permite registar o número de amostras, o valor máximo e o valor mínimo das amostras e obter a média das amostras através dos métodos com nome óbvio.

A função `recebe_informacao`, implementada na classe `stats_glob`, faz a actualização dos contadores correspondentes à acção que ocorreu antes da sua invocação. As acções que podem desencadear a invocação desta função foram enumeradas quando se descreveu a classe `stats`. Os contadores existentes, implementados através de variáveis internas da classe `stats`, registam o número de chamadas oferecidas por uma determinada classe de tráfego entre cada par origem-destino e também o número de chamadas transportadas, respectivamente nas variáveis `trafego_oferecido` e `trafego_transportado`. Os contadores existentes registam também o número de chamadas oferecidas a um feixe e o número de chamadas transportadas no feixe, respectivamente nas variáveis `traf_oferecido_ramo` e `traf_transportado_ramo`. Registam ainda o tráfego de circuitos oferecido a um feixe e o tráfego de circuitos transportado num feixe, respectivamente nas variáveis `circ_oferecido_ramo` e `circ_transportado_ramo`. A classe `stats` possui, além das variáveis referidas, uma outra variável interna, com o nome `circ_disponiveis_ramo` que é uma tabela onde cada elemento é uma instância da classe `estattemp`, que regista para cada feixe o número médio, ponderado, de circuitos disponíveis no feixe além de outros valores estatísticos relativos aos circuitos disponíveis no feixe.

A única variável interna que a classe `stats_glob` possui, tem o nome `cont_adicional` e disponibiliza vários contadores adicionais para cada par origem-destino, para serem usados se necessário. Na implementação desenvolvida esta variável apenas é utilizada no método de encaminhamento RTNR. A função `encaminha_mensagem` da classe `router_rtnr`, referida numa secção seguinte, invoca a função `recebe_informacao` quando pretende actualizar o valor de um dos contadores adicionais. Esta última função, por sua vez, invoca a função `processa_adicional`. Na implementação feita estão a ser utilizados 7 contadores adicionais para cada par origem-destino, para registar entre outras informações o número de chamadas encaminhadas directamente entre cada par origem-destino, o número de chamadas encaminhadas alternativamente e o número de chamadas que não foi possível encaminhar após *crankback*.

Todos os contadores referidos apenas começam a ser actualizados a partir do momento em que a simulação atingiu o instante definido como sendo a altura em que a recolha deve ser iniciada, o que permite tentar evitar recolher valores correspondentes ao estado transitório de arranque da simulação.

A outra função implementada na classe `stats_glob` é a função `termina_estatisticas`. Esta função é invocada pelo processo central no instante imediatamente antes da simulação terminar. Esta

função constrói os ficheiros de resultados, exemplificados e descritos no apêndice F. Para além da informação apresentada nestes ficheiros, outra informação é recolhida durante a simulação. Para os efeitos deste estudo essa informação não foi considerada relevante, no entanto pode se desejado ser apresentada, sendo para tal necessário modificar esta função.

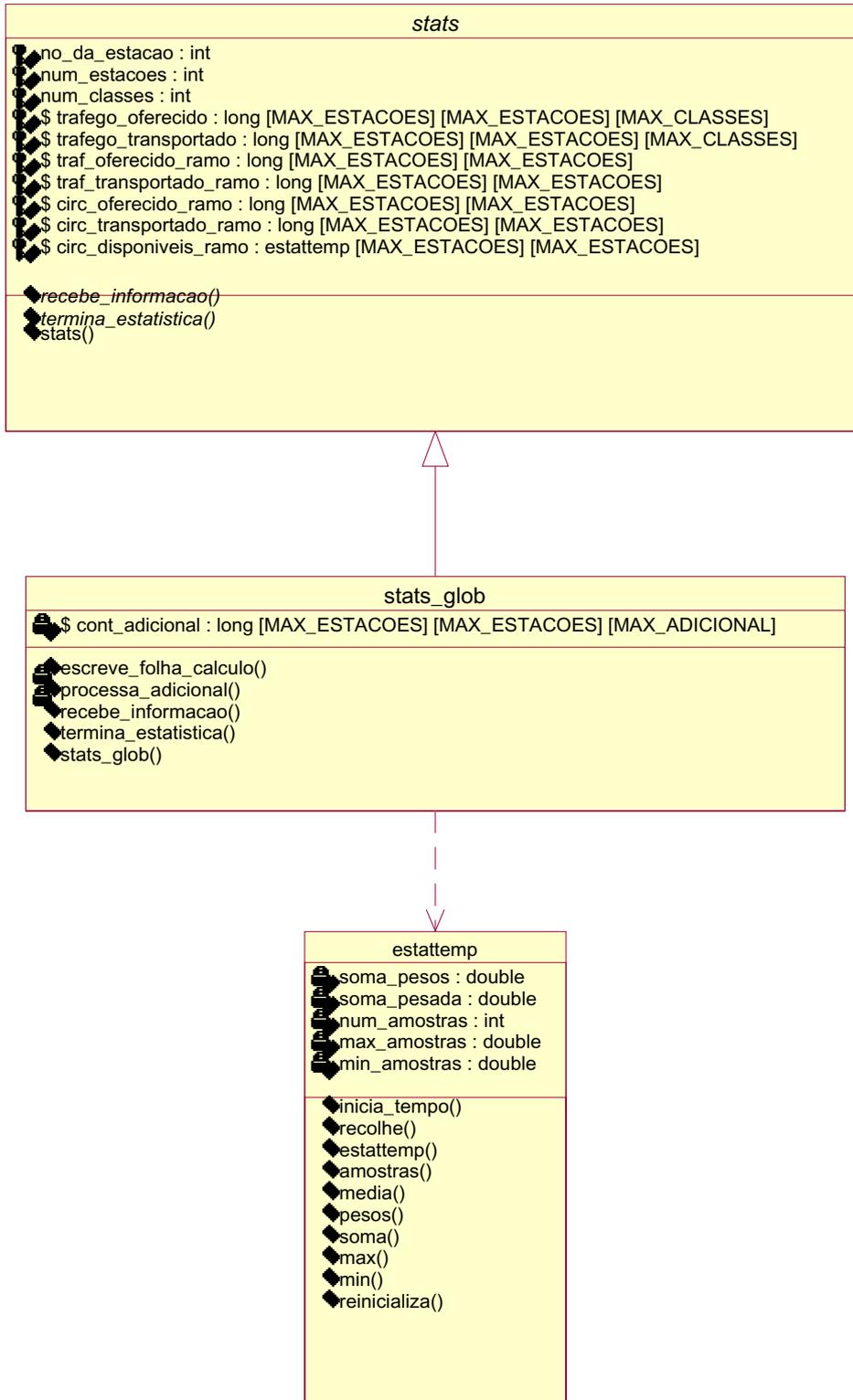


Fig. E-4 - Diagrama de classe stats, glob e estattemp.

Cada um dos componentes seguintes corresponde à implementação de um algoritmo de encaminhamento.

## E.2.2 DIRECTO

No método de encaminhamento DIRECTO foram definidas duas classes, com o nome `router_directo` e `controlador_directo`, que foram derivadas, respectivamente, da classe `router` e da classe `controlador`, como ilustrado na figura seguinte.

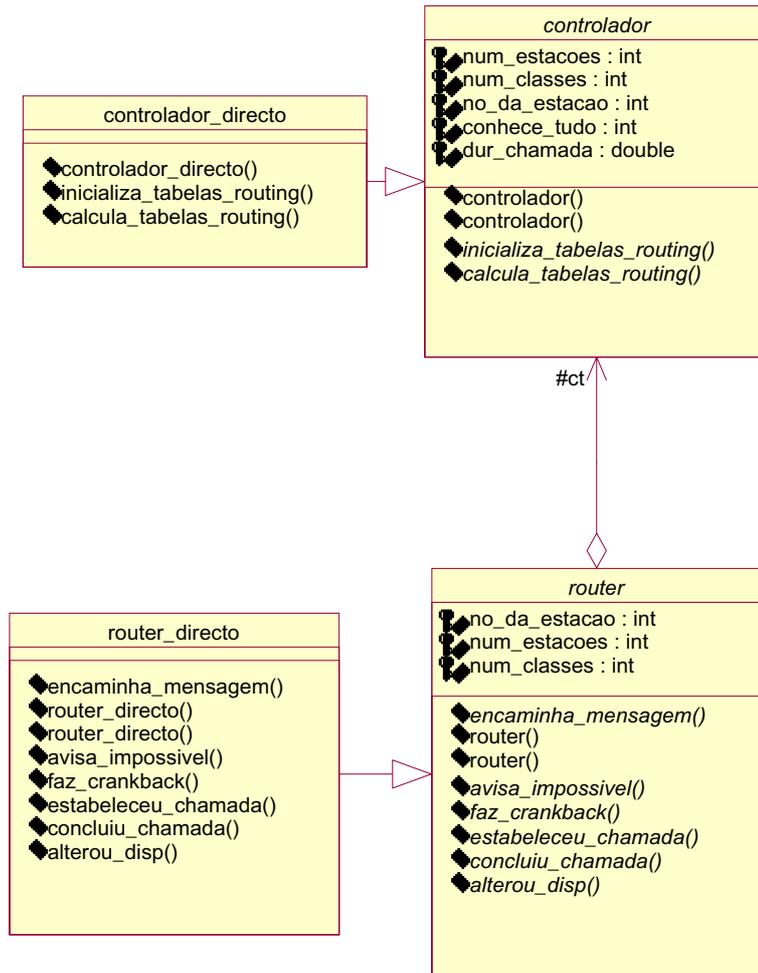


Fig. E-5 - Diagrama de classe específicas do método Directo.

As funções da classe `router` são todas virtuais puras razão pela qual tiveram de se redefinir na classe `router_directo`. No entanto, nesta classe, a função `encaminha_mensagem` foi a única que foi redefinida com alguma funcionalidade as restantes foram redefinidas como funções vazias. A função `encaminha_mensagem`, verifica se há capacidade disponível, no feixe entre a central origem e a central destino, suficiente para encaminhar a chamada. Se assim for indica esse caminho (caminho formado

apenas pelo feixe que liga a central origem à central destino) para ser usado no encaminhamento da chamada, caso contrário devolve a indicação da impossibilidade de a encaminhar.

As funções da classe controlador tiveram de ser redefinidas na classe `controlador_directo` também por se tratar de funções virtuais puras. A função `calcula_tabelas_routing` é chamada pelo processo central, periodicamente, para efectuar algum processamento da informação a usar no encaminhamento das chamadas. Como neste método de encaminhamento não é necessário nenhum tipo de processamento essa função foi definida de modo a tornar o período, entre invocações, grande.

### E.2.3 FAR

No método de encaminhamento FAR também foram definidas duas classes, uma com o nome `router_far` e outra com o nome `controlador_far`, que também foram derivadas, respectivamente, da classe `router` e da classe `controlador`, como ilustrado na figura seguinte.

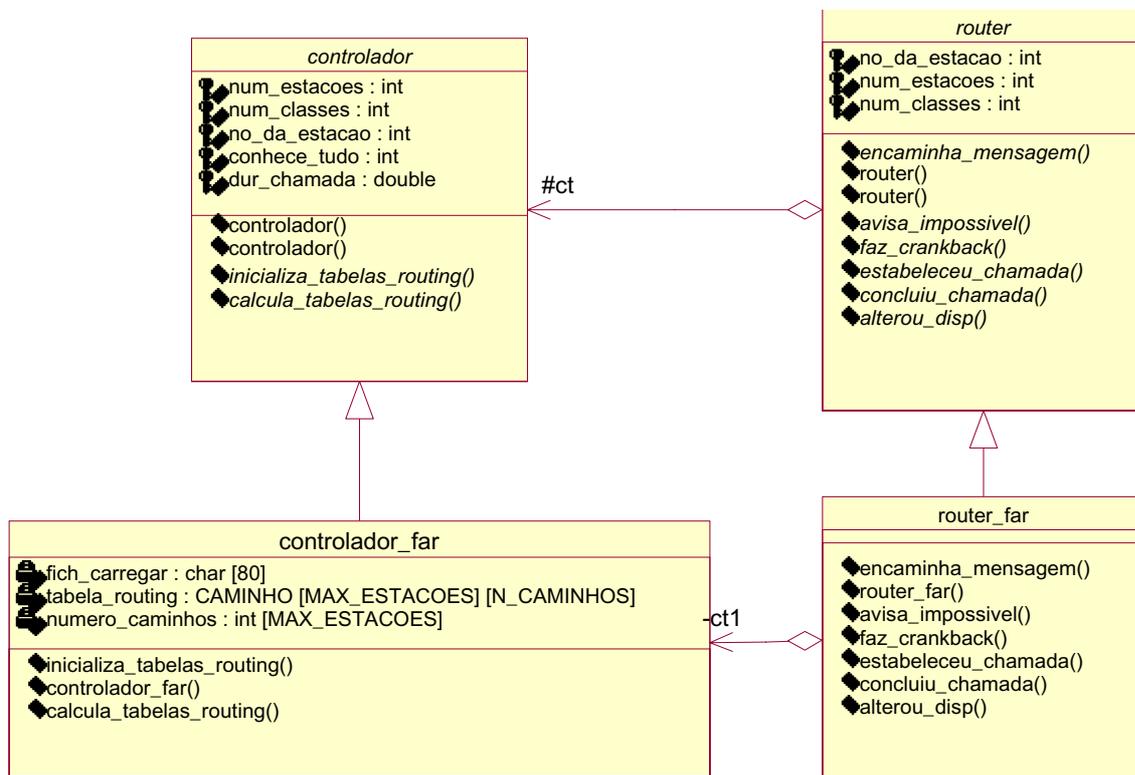


Fig.E-6 - Diagrama de classe específicas do método FAR.

Todas as funções da classe `controlador` tiveram de ser redefinidas na classe `controlador_far`, pelas razões já referidas no método anterior. A função `inicializa_tabelas_routing`, da classe `controlador_far`, tem como tarefa inicializar as variáveis internas, `tabela_routing` (tabelas de encaminhamento) e `numero_caminhos` (número de caminhos entre cada par origem destino). Para o conseguir recorre ao componente carrega tabelas de encaminhamento o qual se descreve a seguir.

### E.2.3.1 Carrega tabelas de encaminhamento

O componente carrega tabelas de encaminhamento é utilizado para obter de um ficheiro as tabelas de encaminhamento, armazenar essa informação e permitir o seu acesso. O nome do ficheiro de onde vão ser lidas as tabelas de encaminhamento é obtido a partir de um parâmetro do módulo central.

A classe que corresponde a este componente, classe `ctr`, possui 4 variáveis internas. Na variável `tab_routing` é onde vão ser armazenadas as tabelas de encaminhamento que são lidas de um ficheiro. Na variável `n_caminhos` vai ser armazenado o número de caminhos entre cada par de centrais. A variável `já_carregado` guarda a indicação de se já se leu a informação do ficheiro ou não, o que permite não ler de novo as tabelas de encaminhamento se essa operação já foi efectuada. Por último a variável `n_estacoes` armazena o número das centrais existentes no sistema a simular.

As funções desta classe são as funções `carrega_tabelas_routing`, `obtem_n_caminhos` e `obtem_caminho`. A função `carrega_tabelas_routing` é a função que armazena na variável `tab_routing` as tabelas de encaminhamento, na variável `n_caminhos` o número de caminhos entre cada par de centrais e coloca na variável `ja_carregado` a indicação de que as tabelas de encaminhamento já foram lidas. A função `obtem_n_caminhos` e a função `obtem_caminho` dão-nos o número de caminhos e um dado caminho, entre um dado par de centrais origem/destino, respectivamente.

Passando de novo às funções da classe `controlador_far`, a função `inicializa_tabelas_routing` começa por chamar a função `carrega_tabelas_routing`, da classe `ctr`, fornecendo-lhe o nome do ficheiro de onde deve ir ler a informação, tendo este nome sido obtido através de um parâmetro do módulo central. Após isto, acede à informação, armazenada por esta função, e inicializa com ela as variáveis internas `tabela_routing` e `numero_caminhos`.

A função `calcula_tabelas_routing` foi redefinida do mesmo modo como no método de encaminhamento directo, pois neste método também não é necessário fazer nenhuma alteração na informação a usar no encaminhamento das chamadas. A informação armazenada nas tabelas de encaminhamento (variável `tabela_routing`) mantém-se fixa após a sua inicialização.

Também aqui, pelas mesmas razões que no método anterior, foi necessário redefinir todas as funções da classe `router` na classe `router_far`. As funções `avisa_impossivel`, `estabeleceu_chamada`, `conclui_chamada` e `alterou_disp` foram redefinidas como funções vazias.

Para determinar o caminho a usar, por uma chamada, a função `encaminha_mensagem` precisa de ter acesso à informação armazenada no controlador (`controlador_far`). Para tal, a classe `router_far` possui um ponteiro para a classe `controlador_far`, como pode ser observado na figura E-6. Este ponteiro foi inicializado no construtor do `router_far`.

A função `encaminha_mensagem` começa por ver se a chamada que recebeu é nova (se é a primeira vez que vai tentar ser encaminhada) ou se é uma chamada de *crankback*. Se a chamada é nova então vai-se determinar a capacidade do caminho directo. Se for superior ou igual à capacidade necessária pela chamada então a função indica ao processo central esse caminho para ser usado no encaminhamento da chamada. No caso de, a função determinar que a chamada não pode ser encaminhada no caminho directo ou no caso de ela ser uma chamada de *crankback*, vão ser tentados, sequencialmente, os caminhos alternativos, armazenados na tabela de encaminhamento (variável `tabela_routing`). Os caminhos alternativos vão ser tentados até se encontrar um que tenha capacidade disponível, no primeiro feixe, ou se terem tentado todos sem sucesso. Um feixe é considerado com capacidade disponível se a capacidade livre, for superior ou igual à capacidade necessária pela chamada mais a capacidade reservada para encaminhamento directo. Se for encontrado um caminho com capacidade disponível, no primeiro feixe, a função indica-o para ser usado no encaminhamento da chamada. Senão a função devolve a indicação da impossibilidade de arranjar um caminho para encaminhar a chamada.

Quando uma chamada tenta um caminho alternativo pode ocorrer que não consiga ser encaminhada por falta de capacidade no segundo feixe do caminho. Neste caso o processo central invoca a função `faz_crankback` para determinar se pode ser tentado novo caminho alternativo. Na função `encaminha_mensagem` foi acrescentado um parâmetro (designado por `pode_tentar_alternativo`) à mensagem correspondente à chamada. Nesse parâmetro é guardada a indicação de se a chamada pode tentar ou não outro caminho alternativo. A função `faz_crankback`, acede a esse parâmetro, simplesmente, para verificar se a chamada pode ou não tentar outro caminho alternativo.

Este método pode ser usado para encaminhar chamadas com diferentes necessidades de recursos, isto é, o número de circuitos necessários em cada feixe do caminho usado para o estabelecimento das chamadas pode ser um qualquer especificado.

## E.2.4 DAR

No método de encaminhamento DAR foram definidas duas classes, com o nome `router_dar` e `controlador_dar`, que também foram derivadas, respectivamente, da classe `router` e da classe `controlador`, como ilustrado na figura seguinte.

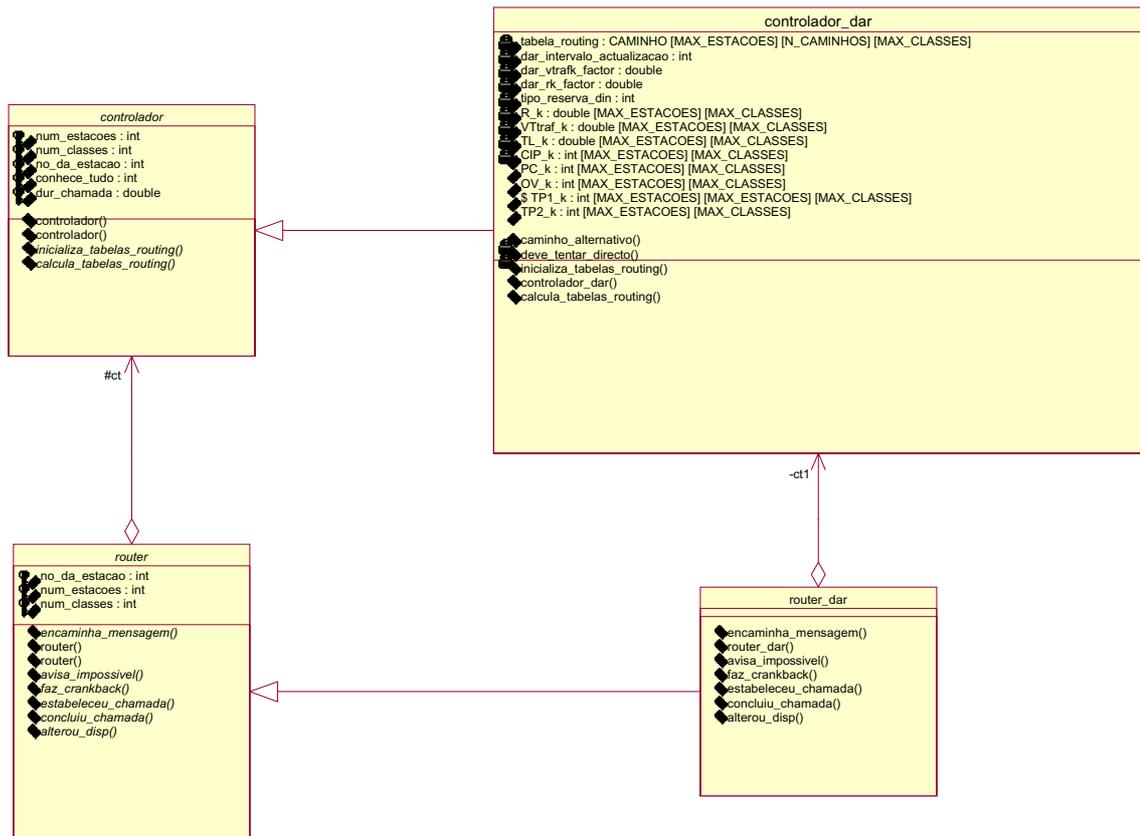


Fig. E.7 - Diagrama de classes específicas do método DAR.

Todas as funções da classe `controlador` tiveram de ser redefinidas na classe `controlador_dar` e todas as funções da classe `router` tiveram de ser redefinidas na classe `router_dar`, pela razões já referidas anteriormente.

Vamos começar por referir as funções da classe `controlador_dar` para em seguida passar às funções da classe `router_dar`. Apenas vamos descrever a implementação das funções pois a utilização de quase todas elas já foi referida quando se descreveram as classes pai.

Os parâmetros do módulo central, `tipo_reserva_din`, `dar_intervalo_atualização`, `dar_vtrafk_factor` e `dar_rk_factor`, são utilizados pelo `controlador_dar` para inicializar as variáveis internas, da classe `controlador_dar`, com o mesmo nome dos parâmetros. A variável interna `tipo_reserva_din` vai armazenar a indicação do tipo de mecanismo de protecção do tráfego de primeiro encaminhamento que é para ser usado (reserva fixa ou reserva dinâmica); a variável interna `dar_intervalo_atualizacao`, guarda o valor do intervalo de tempo que determina os instantes de actualização, do número de circuitos reservados em cada feixe; a variável interna `dar_vtrafk_factor` é usada para permitir a escolha do factor multiplicativo da expressão 3·11 (1.1 nessa expressão); e a variável interna `dar_rk_factor` é usada para permitir a escolha do factor multiplicativo a usar no

cálculo de  $R_k^i$ , na tabela 3-1 o valor utilizado foi 0.05. O construtor além disto também inicializa outras variáveis internas da classe, que representam contadores.

A função `deve_tentar_directo` indica para cada chamada, de uma dada classe, entre um par origem destino, se ela pode ou não tentar estabelecer-se no caminho directo. Se tiver sido escolhida reserva de circuitos dinâmica o critério usado por esta função é o descrito no método de encaminhamento RTNR. Se tiver sido escolhida reserva fixa a função indica sempre que a chamada pode tentar estabelecer-se no caminho directo. A função `caminho_alternativo` é a função que coloca um novo caminho alternativo na tabela de encaminhamento (variável `tabela_routing`, interna da classe `controlador_dar`) para ser usado pelas chamadas, de uma dada classe, entre um par origem destino. A função `inicializa_tabelas_routing`, utiliza a função `caminho_alternativo` para inicializar a tabela de encaminhamento de uma dada central (variável `tabela_routing`, local em cada `controlador_dar`). Por último a função `calcula_tabelas_routing` não é usada para actualizar a informação das tabelas de encaminhamento, pois neste método essa actualização não é feita periodicamente. O que é necessário fazer periodicamente é actualizar o número de circuitos reservados em cada feixe, isto no caso de se estar a usar reserva dinâmica. A função foi então definida do seguinte modo: se o tipo de reserva que se está a usar for reserva fixa a função simplesmente torna o período, entre invocações grande, e se o tipo de reserva que se está a usar for reserva dinâmica a função actualiza o número de circuitos reservados, para cada classe, em cada feixe de acordo com a expressão 5-1 do capítulo 5.

A função `encaminha_mensagem` precisa de ter acesso à informação armazenada no controlador (`controlador_dar`). Para tal, a classe `router_dar` possui um ponteiro para a classe `controlador_dar`, como pode ser observado na figura E-7. Este ponteiro foi inicializado no construtor do `router_dar`.

A função `encaminha_mensagem` começa por fazer a actualização de alguns contadores, necessários no caso de se usar reserva dinâmica para de seguida, passar à função principal que é, obter um caminho para encaminhar a chamada. Para isso determina a capacidade do caminho directo. Se for superior ou igual à capacidade necessária pela chamada e se o controlador indicar que essa chamada pode ser estabelecida no caminho directo então a função indica ao processo central esse caminho para ser usado no encaminhamento da chamada. No caso da função determinar que a chamada não pode ser encaminhada no caminho directo vai ser tentado o caminho alternativo, armazenado na tabela de encaminhamento (variável `tabela_routing`), correspondente à classe de tráfego da chamada que se pretende encaminhar. Se o caminho armazenado tiver capacidade disponível, em ambos os feixes, a função indica-o para ser usado no encaminhamento da chamada. Senão a função devolve a indicação da impossibilidade de arranjar um caminho para encaminhar a chamada, mas antes disso faz com que seja

armazenado um novo caminho alternativo na tabela de encaminhamento (variável `tabela_routing`), invocando para isso a função `caminho_alternativo` da classe `controlador_dar`. Um feixe é considerado com capacidade disponível se a capacidade livre, for superior ou igual à capacidade necessária pela chamada mais a capacidade reservada para encaminhamento directo.

As funções `estabeleceu_chamada` e `concluiu_chamada` são invocadas, pelo processo central, sempre que uma chamada foi estabelecida e sempre que uma chamada foi concluída, respectivamente. A sua função, neste método, é apenas fazer o incremento dos contadores do número de chamadas estabelecidas e do número de chamadas concluídas, respectivamente. Estes contadores são necessários na actualização dos valores dos números de circuitos reservados, feita apenas no caso em que se usa reserva dinâmica.

Por último, a função `faz_crankback` devolve sempre a indicação de que `crankback` não é permitido. As funções `avisa_impossivel` e `alterou_disp`, foram redefinidas como funções vazias.

Pela descrição feita deve ter ficado claro que esta implementação permite o encaminhamento simultâneo de vários tipos de tráfego.

## E.2.5 DCR

Na implementação do método de encaminhamento DCR foram definidas duas classes, uma com o nome `router_dcr` e outra com o nome `controlador_dcr`, como ilustrado na figura E-8.

As funções da classe `controlador_dcr` são as funções `calcula_prob_caminho`, `actualiza_Rs`, `actualiza_As`, `controlador_dcr`, `inicializa_tabelas_routing` e `calcula_tabelas_routing`. No construtor desta classe são inicializadas algumas variáveis internas da classe. São inicializadas as variáveis internas que representam contadores, com valores nulos. E, são também inicializadas as variáveis internas, `tau`, `delta`, `theta`, `tipo_reserva_din`, `dcr_intervalo_atualizacao`, `dcr_vtrafk_factor` e `dcr_rk_factor`, a partir de parâmetros do módulo central. Algumas destas variáveis são necessárias no cálculo das expressões definidas, no capítulo 3, no método de encaminhamento DCR (caso de `tau` – intervalo de extrapolação, `delta` – intervalo de actualização, `theta` – peso de média móvel), outras são necessárias se for usada reserva dinâmica como definido pela expressão 5-1 do capítulo 5. A sua utilização será melhor compreendida depois de uma análise das restantes funções da classe `controlador_dcr`.

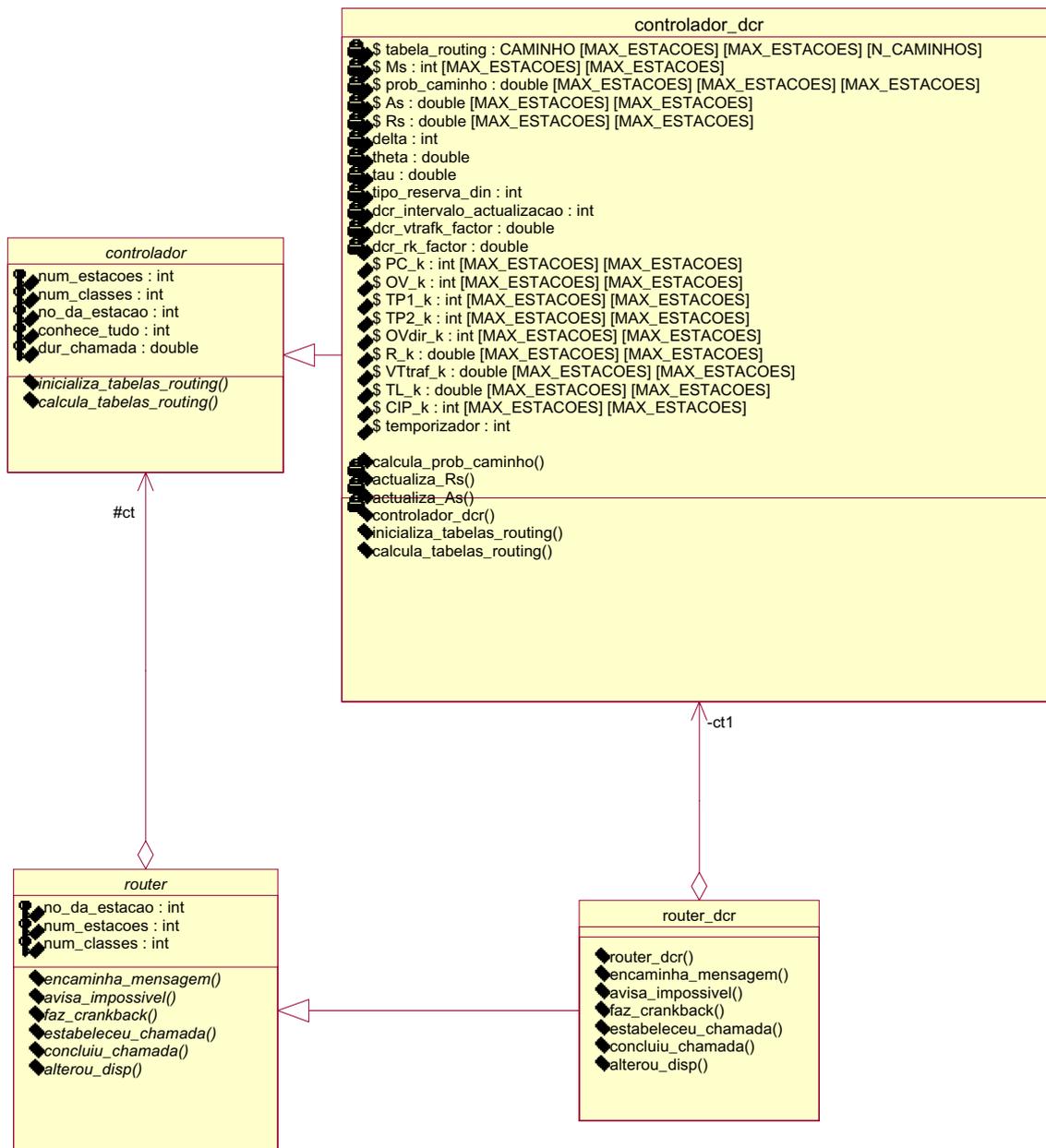


Fig. E-8 - Diagrama de classes específicas do método DCR.

A função `inicializa_tabelas_routing` inicializa as matrizes `As`, `Ms`, `Rs`, `prob_caminho` e `tabela_routing`. As primeiras são inicializadas com zeros e a última é inicializada pela invocação da função `calcula_tabelas_routing`. A função `actualiza_As` calcula, para todos os feixes, os valores obtidos a partir da expressão 3-13 que representam as intensidades, estimadas, de chegada de chamadas, a cada um dos feixes (armazena esses valores na variável `As`). Para esse cálculo são utilizados os valores escolhidos para o intervalo de actualização (armazenado na variável `delta`) e também para o coeficiente (armazenado na variável `theta`). Por sua vez a função `actualiza_Rs` calcula, para todos os feixes, os valores obtidos a partir da expressão 3-12 que representam estimativas da capacidade residual de cada um dos feixes (armazena esses valores na variável `Rs`). Para esse cálculo

é utilizado o valor escolhido para o intervalo de extrapolação (armazenado na variável `tau`). A função `calcula_prob_caminho` vai obter, para todas as centrais intermédias entre todos os pares de centrais, os valores a partir da expressão 3-14, que representam a probabilidade de selecção de uma qualquer central intermédia por uma chamada entre quaisquer duas centrais (vai guardá-los na variável `prob_caminhos`). Como para o cálculo desses valores é necessário ter os valores de `Rs` actualizados e como para o cálculo destes é necessário ter os valores de `As` actualizados, a função começa por invocar `actualiza_As` e de seguida `actualiza_Rs`. O que de facto é guardado na variável `prob_caminhos` não é o valor das probabilidades obtido pela expressão 3-14, mas sim as probabilidades cumulativas.

A função `calcula_tabelas_routing` é invocada periodicamente, sendo o período de invocação igual ao valor armazenado na variável `delta`. Tem por função armazenar na tabela de encaminhamento (variável `tabela_routing`) o caminho alternativo entre quaisquer dois pares de centrais origem-destino. Para isso, recorre aos valores das probabilidades de selecção das centrais intermédias (armazenados na variável `prob_caminho`, depois de ter invocado `calcula_prob_caminho`). Além disso, vai fazer a actualização do número de circuitos reservados entre todos os pares de centrais (apenas no caso de a variável `tipo_reserva_din` indicar que se está a usar reserva dinâmica). Nesse caso, a actualização é feita segundo a expressão 5-1 ou segundo a expressão 5-2 (ambas do capítulo 5) dependendo do valor armazenado na variável `tipo_reserva_din`. O valor armazenado na variável `dcr_intervalo_atualizacao` indica a frequência com que esta actualização é feita. Como o método de encaminhamento DCR é um método centralizado, e para tornar a implementação mais semelhante ao sistema a ser simulado, a função `calcula_tabelas_routing` só tem as funcionalidades descritas numa única central, a tarefa das funções `calcula_tabela_routing` das outras centrais é nula. A central na qual os cálculos são obtidos tem assim também a função desempenhada pelo processador central no sistema.

Vamos agora analisar a classe `router_dcr`. O `router` (`router_dcr`) acede à informação armazenada no `controlador` (`controlador_dcr`) através da utilização de um ponteiro, como já acontecia nos métodos anteriores. As funções `estabeleceu_chamada`, `concluiu_chamada`, `faz_crankback`, `avisa_impossivel` e `alterou_disp` foram definidas neste método com as mesmas funcionalidades que as funções com o mesmo nome no método DAR. A função `encaminha_mensagem` possui, também, praticamente as mesmas funcionalidades que a função `encaminha_mensagem` do método anterior. Possui porém duas excepções, sendo a primeira que quando a capacidade do caminho directo é superior ou igual à necessária pela chamada, então esse é sempre o caminho indicado ao processo central como o caminho que deve ser usado para encaminhar a chamada. Isto é assim porque aqui não podem existir várias classes a competir pelos mesmos circuitos (a implementação deste método é monoclasse). A segunda excepção é, quando a função tenta o caminho alternativo sem sucesso, esse caminho

permanece o caminho alternativo armazenado na tabela de encaminhamento (variável `tabela_routing`). Pois neste método o caminho alternativo é apenas determinado periodicamente.

### E.2.6 RTNR

Todas as referências feitas neste ponto a expressões, tabelas, etc, referem-se a definições feitas no capítulo 3, a menos que seja dito o contrário.

Na implementação foram definidas duas classes, uma com o nome `router_rtnr` e outra com o nome `controlador_rtnr`, como ilustrado na figura seguinte.

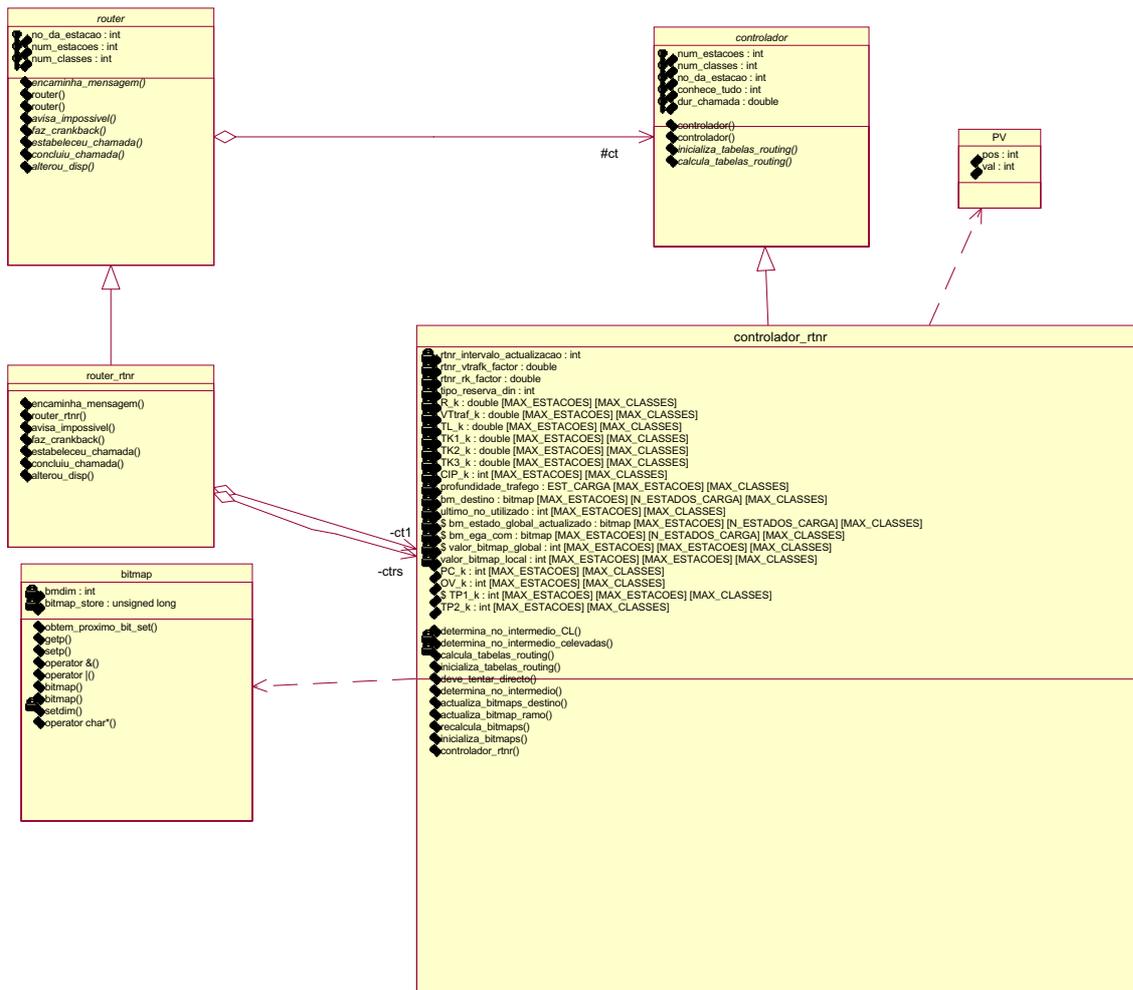


Fig. E-9 - Diagrama de classes específicas do método RTNR.

As funções da classe `router_rtnr` são as designadas por `router_rtnr`, `encaminha_mensagem`, `avisa_impossivel`, `faz_crankback`, `alterou_disp`, `estabeleceu_chamada` e `concluiu_chamada`. A classe `router_rtnr` além dos membros já referidos possui também um vector de ponteiros para armazenar os ponteiros para os controladores (`controlador_rtnr`) associados às várias centrais da rede. A justificação para a necessidade destes ponteiros é a seguinte: Quando uma

chamada é estabelecida ou concluída é solicitado ao *router* (`router_rtnr`), associado à central destino ou à central origem, respectivamente, para actualizar os *bit maps*. A actualização dos *bit maps* tem de ser feita nos controladores (`controlador_rtnr`), associados à central origem, à central destino e possivelmente a uma central intermédia, no caso de a chamada ter usado um caminho de dois feixes. Por isto os *routers* têm que ter um meio de aceder a determinados controladores, quando uma chamada é estabelecida e também quando uma chamada é concluída. Isto é conseguido com o vector de ponteiros referido. Foi definido como um membro estático o que faz com que possa ser partilhado (variável global) por todos os *routers*. Recorde que os *bit maps* são o modo utilizado no RTNR para registar, em cada central, o estado de carga dos feixes que ligam essa central a todas as outras na rede. Recorde também que essa informação, registada nas centrais, é actualizada todas as vezes que um circuito virtual é libertado ou ocupado num feixe.

Vamos agora analisar cada uma das funções da classe `router_rtnr`. Começamos por referir o construtor que tem como única função inicializar um dos elementos, do vector de ponteiros referido anteriormente, com o endereço do `controlador_rtnr` a si associado.

Relembre, da descrição do método de encaminhamento RTNR feita no capítulo 3, que para tentar reduzir o atraso no estabelecimento de uma chamada, quando é necessário um caminho de dois feixes, a central origem utiliza a última informação que foi usada numa ligação entre essa origem/destino. Só se a chamada não puder ser estabelecida com essa informação é que vai tentar estabelecer-se com a informação actualizada. Isto tem como consequência a necessidade das centrais terem a possibilidade de *crankback*.

A função `encaminha_mensagem` vai associar à mensagem, correspondente a uma chamada, um parâmetro para permitir identificar se a chamada deve ainda retornar por *crankback*, nas situações em que seja necessária essa decisão. A esse parâmetro deu-se o nome de `pode_tentar_alternativo`. A função começa por verificar se a mensagem que recebeu já possui o parâmetro `pode_tentar_alternativo`. Caso a mensagem ainda não o possua isso indica que é a primeira vez que se está a tentar encaminhar essa chamada. Nesta situação a função começa por adicionar o parâmetro à mensagem de modo a indicar que a chamada ainda pode tentar um caminho alternativo. E em seguida vai testar a capacidade disponível do caminho directo. Se o caminho directo tiver capacidade disponível e se for permitido a essa chamada usá-lo (indicação dada pela função `deve_tenter_directo`, da classe `controlador_rtnr`) então a função `encaminha_mensagem` indica esse caminho para ser usado no encaminhamento da chamada. Se a função determinar que a chamada não pode ser encaminhada no caminho directo então vai pesquisar qual o último caminho alternativo usado para esse destino. Para tal vai utilizar a última informação que recebeu da central destino, *bit maps* locais correspondentes ao destino em causa (invocando a função

determina\_no\_intermedio, da classe controlador\_rtnr). E de seguida actualiza esses *bit maps* (invocando a função *actualiza\_bitmaps\_destino*, da classe controlador\_rtnr). Se a central intermédia, indicada pela informação não actualizada, não existir ou se a capacidade disponível do feixe entre a central origem e essa central intermédia não for suficiente para a chamada então não é possível encaminhar a chamada com a informação não actualizada. Nesta situação é colocado no parâmetro *pode\_tentar\_alternativo* a indicação de que não pode ser tentado outro caminho alternativo, pois o caminho alternativo que resta vai ser tentado por esta função de seguida. Agora a função vai determinar uma nova central intermédia mas com os *bit maps*, correspondentes ao destino, actualizados (invocando de novo a função *determina\_no\_intermedio*). Se também não for possível encontrar uma central intermédia, para formar um caminho alternativo, com a informação actualizada então a função *encaminha\_mensagem* indica ao processo central que não foi possível encontrar um caminho para estabelecer a chamada. Se por outro lado for possível encontrar uma central intermédia (com a informação actualizada ou não) então a função *encaminha\_mensagem* indica o caminho com essa central de trânsito como caminho a ser seguido pela chamada.

Se quando a função foi invocada a mensagem já possuía o parâmetro *pode\_tentar\_alternativo* isto indica que a central tentou estabelecer a chamada num caminho alternativo, determinado com informação não actualizada, da central destino, que foi descoberto como indisponível a partir da central de trânsito. Isto levou a que a chamada retornasse por *crankback* para a central origem e a central origem então chamou, de novo, a função *encaminha\_mensagem*. Nesta situação, quando a função é invocada coloca no parâmetro *pode\_tentar\_alternativo* a indicação de que não pode ser tentado mais nenhum caminho alternativo pois o caminho que ainda é possível tentar vai ser tentado de seguida. Para isso a função determina uma nova central intermédia, mas agora com a informação actualizada (invocando a função *determina\_no\_intermedio*). Se não for possível encontrar essa central intermédia a função *encaminha\_mensagem* devolve essa indicação ao processo central. Se, caso contrário, for possível encontrar uma central intermédia (com a informação actualizada) então a função indica o caminho, com essa central intermédia, como caminho alternativo.

Quando uma chamada se encontra numa central intermédia e esta determina não ser possível atingir a central destino a partir desse ponto, a central intermédia invoca a função *faz\_crankback*. Esta função indica a essa central se ela deve ou não enviar a chamada por *crankback* para tentar novo caminho alternativo. A função consegue dar essa indicação simplesmente através da análise ao parâmetro da mensagem *pode\_tentar\_alternativo*.

A função *avisa\_impossivel*, invocada pela central origem quando recebe uma chamada, vinda de uma central intermédia, que não foi possível estabelecer, apenas tem por função incrementar o contador de chamadas, de uma dada classe, não entregues para um determinado destino.

A função `concluiu_chamada`, invocada pela central origem, quando uma chamada esgotou o tempo de ocupação, foi definida de modo a fazer a actualização dos *bit maps* correspondente ao feixe ou feixes libertados. E a função `estabeleceu_chamada`, invocada pela central destino, quando se acabou de estabelecer uma chamada, foi definida de modo a fazer a actualização dos *bit maps* correspondentes ao feixe ou feixes ocupados. Para fazer estas actualizações as funções utilizam o vector de ponteiros para os controladores, já anteriormente referido, necessário porque, como indicado, os bit maps estão armazenados nos controladores (`controlador_rtnr`), e são actualizados por funções deste, neste caso, pela função `actualiza_bit_map_amo`.

A função `alterou_disp` não tem nenhuma funcionalidade neste encaminhamento.

As funções da classe `controlador_rtnr` são as designadas por `calcula_tabelas_routing`, `inicializa_tabelas_routing`, `deve_tentar_directo`, `inicializa_bitmaps`, `determina_no_intermedio`, `actualiza_bitmaps_destino`, `recalcula_bitmaps`, `actualiza_bitmaps_amo` e `controlador_rtnr` (o construtor da classe). Todas estas funções são funções públicas. Além destas funções públicas a classe possui duas funções privadas que são `determina_no_intemedio_celevadas` e `determina_no_intermedio_CL`.

O construtor desempenha as mesmas funções que o construtor da classe `controlador_dar`. Utiliza alguns parâmetros do módulo central (`tipo_reserva_din`, `rtnr_rk_factor`, `rtnr_vtrafk_factor` e `rtnr_intervalo_atualizacao`) para inicializar as variáveis internas, da sua classe, com o mesmo nome dos parâmetros. O parâmetro `tipo_reserva_din` indica qual o tipo de reserva dinâmica deve ser usada. O parâmetro `rtnr_intervalo_atualizacao` indica com que periodicidade devem ser feitas as actualizações da informação. O parâmetro `rtnr_vtrafk_factor` dá-nos o factor multiplicativo a usar no cálculo da expressão 3.11 do capítulo 3. E o parâmetro `rtnr_rk_factor` dá-nos o factor multiplicativo a usar no cálculo dos valores de  $R_k$  dados pela tabela 3.1 do capítulo 3. No construtor são inicializadas também outras variáveis internas da classe, algumas das quais representam contadores, com valores nulos. Essas variáveis são as matrizes  $P_{c,k}$ ,  $OV_k$ ,  $VTtraf_k$ ,  $R_k$ ,  $TK1_k$ ,  $TK2_k$ ,  $TK3_k$  e  $TL_k$  que correspondem, com grande similaridade, à notação utilizada na descrição, do método de encaminhamento, feita no capítulo 3. E ainda as variáveis  $TP1_k$  e  $TP2_k$  que armazenam o contador de chamadas transportadas de uma origem para um destino, antes de terminarem o tempo de ocupação e depois de terminarem o tempo de ocupação, respectivamente (a diferença entre estes dois contadores indica o número de chamadas nesse momento em curso).

Vamos dividir as funções da classe `controlador_rtnr` em três grupos, de acordo com as suas funcionalidades. O primeiro grupo contendo apenas a função `calcula_tabelas_routing`, o segundo grupo contendo as funções `inicializa_tabelas_routing`, `inicializa_bitmaps`, `recalcula_bitmaps`, `actualiza_bitmap_amo` e `actualiza_bitmaps_destino` e por último o

terceiro grupo contendo as funções `deve_tentar_directo`, `determina_no_intermedio`, `determina_no_intermedio_CL` e `determina_no_intermedio_celevadas`. Ficando assim no primeiro grupo a função que actualiza periodicamente alguma da informação necessária na actualização dos *bit maps*, que são usados nas decisões de encaminhamento, no segundo grupo as funções de actualização dos *bit maps* e no terceiro grupo as funções solicitadas pelo *router* (`router_rtnr`) quando necessita de encaminhar uma chamada. As funções deste último grupo que são invocadas directamente pelo `router_rtnr` são as funções `deve_tentar_directo` e `determina_no_intermedio`. As funções `determina_no_intermedio_celevadas` e `determina_no_intermedio_CL` são invocadas pela função `determina_no_intermedio` conforme a explicação que se segue. O objectivo da função `determina_no_intermedio` é a determinação da central intermédia preferida para ser usada no encaminhamento alternativo da chamada (utilizando os *bit maps* do destino actualizados, ou não). A função foi definida de modo a começar a pesquisa pelos estados de carga mais leves. Por isso começa por chamar a função `determina_no_intermedio_CL`. Esta função pesquisa os *bit maps* começando pelo estado de carga LL1 até ao estado LL3. Logo que encontre uma central intermédia deixa de pesquisar os próximos *bit maps* e devolve essa central, ou, no caso contrário, devolve a indicação da impossibilidade de encontrar uma central intermédia com feixes nos estados LL1, LL2 ou LL3. Neste caso, a função `determina_no_intermedio`, vai solicitar à função `determina_no_intermedio_celevadas` para esta pesquisar nos *bit maps* do estado de carga HL. Se mesmo assim ainda não foi possível encontrar uma central intermédia, então a função `determina_no_intermedio` solicita de novo à função `determina_no_intermedio_celevadas` para esta pesquisar, agora nos *bit maps* do estado de carga reservado. No entanto, só ocorrem essas invocações, à função `determina_no_intermedio_celevadas`, no caso da profundidade de selecção do caminho alternativo o permitir.

Vamos analisar a implementação, seguida tanto pela função `determina_no_intermedio_CL` como pela função `determina_no_intermedio_celevadas`, relativa à determinação de uma central intermédia a partir da informação dos *bit maps*. Em ambas as funções são usados o *bit map* correspondente à central destino e o *bit map* correspondente à central origem. Dependendo da informação que a função tenha recebido (utilização dos *bit map* actualizados ou não) o *bit map* correspondente à central destino pode ser obtido da informação armazenada localmente no `controlador_rtnr` (informação que foi obtida pela central, daquele destino, no estabelecimento da última chamada) ou da informação dos *bit maps* actualizados armazenada na variável partilhada por todos os controladores. Seguidamente esses dois *bit maps* são sujeitos a uma conjunção lógica (*AND*). Agora vão procurar um *bit* a um neste *bit map*, recorrendo à função `obtem_proximo_bit_set` do componente `bits`. Se não foi encontrado nenhum então as funções devolvem a indicação de que não foi possível encontrar nenhuma central intermédia, utilizando os *bit maps* respectivos. Caso contrário, existe

pelo menos uma central, vai então ser escolhida a melhor de acordo com o estado de carga dos feixes que a ela ligam. Para tal calculam as somas dos estados de ambos os feixes, de acordo com a tabela 3-6 do capítulo 3, para cada uma das centrais identificadas como possíveis centrais intermédias. Depois do cálculo das somas, para as várias centrais intermédias aceitáveis, as funções vão devolver a central intermédia com o maior valor da soma. No caso do maior valor da soma corresponder à última central que foi utilizada numa ligação entre essa origem/destino e existir outra com o mesmo valor para a soma então as funções devolvem sempre a alternativa. Para a implementação das funções relativa ao cálculo das somas e determinação da maior delas foi definida uma estrutura de dados (um vector de estruturas) que armazena em cada posição o valor da soma e a posição do *bit map* correspondente a essa soma. As funções armazenam em cada posição do vector as somas e respectivas posições, correspondentes apenas às posições a “1” do *bit map* resultante da conjunção lógica, como já referido. Depois recorrem a uma função definida com o nome `bs` (definida utilizando o algoritmo de ordenação `bubble sort`, de modo a colocar o vector com os valores das somas ordenados por ordem decrescente) para obterem o vector ordenado pelas somas, ficando com o maior valor da soma na primeira posição do vector.

A última função do terceiro grupo que falta descrever é a funções `deve_tentar_directo` que devolve a indicação de se uma chamada entre uma determinada origem/destino, de uma dada classe, deve ou não tentar o caminho directo. A função foi implementada do seguinte modo. Começa por analisar se o número de chamadas em progresso da origem para o destino, da classe em causa é inferior ao número de chamadas planeadas, se for verdadeiro então a função indica que a chamada pode tentar o caminho directo. Se for falso, a função verifica se a largura de banda livre no feixe é superior à largura de banda reservada pelas outras redes virtuais que não têm o seu objectivo de bloqueio satisfeito (inequação 3-8 do capítulo 3). No caso de ser verdadeiro, a função indica, também nesta situação, que a chamada pode tentar o caminho directo.

A função do primeiro grupo, `calcula_tabelas_routing`, actualiza periodicamente alguma da informação necessária na actualização dos *bit maps*. A função vai determinar, para cada uma das redes virtuais (associada a cada classe de serviço), vários valores. O código no início da implementação da função determina o nível de bloqueio total (variável `TO_j`, local à função), da central à qual este controlador está associado para todas as outras, de acordo com a expressão 3-4 do capítulo 3. O próximo código serve para determinar os valores do nível de bloqueio ponto a ponto (variável `NN_k`, local à função), do número de chamadas em progresso (variável `CIP_k`, interna da classe `controlador_rtnr`), da carga de tráfego estimada (variável `TL_k`, interna da classe `controlador_rtnr`), do número de circuitos virtuais necessário para obter um determinado grau de serviço (variável `VTtraf_k`, interna da classe `controlador_rtnr`), dos limiares associados com a profundidade de selecção do caminho (variáveis `Dh1_k` e `Dr_k`, locais à função), da profundidade de selecção do caminho alternativo (variável `profundidade_trafego`, variável interna da classe

controlador\_rtnr), do nível de reserva ponto a ponto (variável  $R_k$ , variável interna da classe controlador\_rtnr), dos limiares do estado de carga dos feixes (variáveis  $TK1_k$ ,  $TK2_k$ ,  $TK3_k$ , internas da classe controlador\_rtnr). Todos estes valores são calculados para cada um dos destinos, possíveis a partir da central associada ao controlador (controlador\_rtnr) na qual a função vai ser executada. A  $NN_k$  foi atribuído o valor da razão entre o contador do número de chamadas bloqueadas (variável  $OV_k$ , interna da classe controlador\_rtnr) e o contador do número de chamadas oferecidas (variável  $PC_k$ , interna da classe controlador\_rtnr), para um destino, no intervalo. O valor de  $CIP_k$  foi obtido da diferença entre os contadores  $TP1_k$  e  $TP2_k$ . Os valores atribuídos às variáveis  $TL_k$  e  $VTtraf_k$  são o resultado da implementação das expressões 3.10 e 3.11, respectivamente. O valor atribuído à variável  $R_k$  foi obtido de acordo com a tabela 3.1. Os valores atribuídos às variáveis  $TK1_k$ ,  $TK2_k$ ,  $TK3_k$  foram obtidos de acordo com a tabela 3.8. E os valores atribuídos às variáveis  $Dh1_k$ ,  $Dr_k$  foram obtidos de acordo com a tabela 3.10. Na variável *profundidade\_trafego*, é guardada a indicação da profundidade de selecção do caminho, que pode ser caminho reservado, caminho com carga pesada ou caminho no estado de carga LL3, dependendo dos valores das variáveis  $Dh1_k$ ,  $Dr_k$  e  $VTtraf_k$ , como descrito no capítulo 3. Depois de terem sido actualizados todos os valores referidos é invocada a função *recalcula\_bitmaps*, descrita no próximo parágrafo, de modo a reflectir esta actualização nos *bit maps*. Por último a função reinicializa as variáveis  $TP2_k$ ,  $PC_k$ ,  $OV_k$  e  $TP1_k$ , que representam contadores, as três últimas com valores nulos.

No segundo grupo colocamos as funções de actualização dos *bit maps* que são as funções *inicializa\_tabelas\_routing*, *inicializa\_bitmaps*, *recalcula\_bitmaps*, *actualiza\_bitmap\_ramo* e *actualiza\_bitmaps\_destino*.

Recorde que, no sistema cada central possui um *bit map* (lista de bits, com um número de bits igual ao número de centrais na rede), por cada um dos estados de carga para cada uma das classes de tráfego. E que em cada *bit map* uma entrada a “1” significa que o feixe que liga a central correspondente a essa entrada à central possuidora desse bit map tem carga igual ou menor que o estado de carga representado por esse bit map. Na implementação foram usados seis estados de carga discretos para os feixes. Uma importante variável interna da classe controlador\_rtnr é a variável *bm\_ega\_com*. Essa variável é uma matriz que armazena a informação dos *bit maps* de todas as centrais. Uma outra variável, interna à classe controlador\_rtnr, do mesmo tipo que a anterior é a variável *bm\_estado\_global\_atualizado*. Esta variável foi usada principalmente para *debug*, a diferença entre ela e a anterior está no significado das entradas a “1” que aqui indicam que feixe correspondente a essa entrada tem carga igual ao estado de carga representado por esse *bit map*. Estas variáveis foram declaradas como estáticas o que as torna variáveis partilhadas por todos os controladores (controlador\_rtnr). Uma outra variável, também interna à classe controlador\_rtnr, que guarda

informação com o mesmo significado que a variável `bm_ega_com`, é a variável `bm_destino`. Nesta variável é guardada a informação que esteve armazenada na variável `bm_ega_com`, e que possivelmente não reflecte as últimas actualizações feitas em `bm_ega_com`. A necessidade desta variável prende-se com o facto de ser preciso armazenar em cada `controlador_rtnr` as últimas actualizações dos *bit maps*, que foram recebidas, de todos os destinos, pois vai ser baseado nesta informação a determinação do caminho alternativo, e só se não for possível com ela é que se utiliza a informação dos bit maps actualizados. Relacionada ainda com a informação dos *bit maps* temos ainda as variáveis `valor_bitmap_global` e `valor_bitmap_local` que armazenam o valor, correspondente ao estado do feixe, entre quaisquer duas centrais, de acordo com a tabela 3-6 do capítulo 3. A primeira, é uma variável estática logo partilhada por todos os controladores, onde vão estar armazenados valores sempre actualizados. Ao passo que na segunda, uma variável local a cada `controlador_rtnr`, vão estar armazenados valores, que correspondem à informação do estado armazenada na variável `bm_destino`, e que por isso, podem ser valores que não correspondem ao estado actual dos feixes. A última variável que nos falta referir da classe `controlador_rtnr` é a variável `ultimo_no_utilizado` que regista, em cada controlador, as últimas centrais intermédias utilizadas em ligações para cada uma das outras centrais, por cada uma das classes.

A função `inicializa_bitmaps` faz a inicialização de parte das variáveis `bm_ega_com`, `bm_estado_global_atualizado`. A parte das variáveis que é inicializada é a parte relativa aos *bit maps* correspondentes à central que tem associado o `controlador_rtnr` que corre esta função. Começa por inicializar a zeros todas as posições, que são inicializadas pela função (*bit maps* da central correspondente), da variável `bm_estado_global_atualizado`. E de seguida coloca, nessa variável, para cada uma das classes, uns em todas as posições, correspondentes ao estado LL1, que verifiquem a condição de a capacidade do feixe ser superior à necessidade da classe em questão. Se esta condição não se verifica então os uns são colocados nas posições da variável correspondentes ao estado Ocupado. A inicialização da variável `bm_ega_com` é feita também de acordo com a verificação ou não da condição anterior. As inicializações anteriores foram conseguidas recorrendo às funções do componente bits. A função `inicializa_tabelas_routing` simplesmente invoca a função `inicializa_bitmaps`. A função `actualiza_bitmap_amo` é invocada pela função `recalcula_bitmaps` e também pelas funções `estabeleceu_chamada` e `concluiu_chamada` da classe `router_rtnr`. A função `recalcula_bitmaps` é utilizada sempre que se pretende actualizar a totalidade dos *bit maps*, o que é necessário sempre que haja actualização da informação, na qual eles são baseados. Sempre que se estabeleceu ou se concluiu uma chamada também é necessário fazer a actualização dos *bit maps*, não da totalidade mas apenas da parte correspondente aos feixes envolvidos. Surge daí a necessidade da invocação da função `actualiza_bitmap_amo` tanto pela função `estabeleceu_chamada` como pela função `concluiu_chamada`.

A função `actualiza_bitmap_ramo` começa por calcular os valores de `RTtraf_k` (variável local da função, onde vai ficar armazenado o número de circuitos reservados, para as várias classes, relativos ao feixe em questão), para todas as classes. Esses valores são calculados de acordo com a expressão 3-5 do capítulo 3 ou de acordo com a simplificação apresentada para essa expressão na implementação do método de encaminhamento DAR, expressão 5-1, do capítulo 5. É usado o parâmetro `tipo_reserva_din` para decidir qual das expressões anteriores deve ser usada. Após o cálculo destes valores a função vai actualizar a informação dos *bit maps* relativa ao feixe em questão, para todas as classes (actualização feita nas variáveis `bm_ega_com` e `bm_estado_global_atualizado`), de acordo com a tabela 3-9. Para fazer essas actualizações a função necessitou de calcular o valor da expressão 3-6, o qual armazena na variável `RBWtraf_k`, para cada uma das classes. De acordo com a informação armazenada nos *bit maps* a função também actualiza parte da variável `valor_bitmap_global` com os valores correspondentes, dados na tabela 3-6. A função `actualiza_bitmaps_destino` actualiza a parte das variáveis `bm_destino` e `valor_bitmap_local`, correspondente a um determinado destino e uma determinada classe, com informação armazenada nas variáveis `bm_ega_com` e `valor_bitmap_global`, respectivamente. Genericamente o que a função faz é actualizar a informação armazenada localmente no controlador `rtnr`, para um determinado destino e determinada classe. A função `recalcula_bitmaps` recalcula a totalidade dos *bit maps*, ou seja actualiza completamente as variáveis `bm_estado_global_atualizado`, `bm_ega_com` e `valor_bitmap_global`. Consegue-o invocando várias vezes a função `actualiza_bitmap_ramo`.

A informação do estado dos feixes (*bit maps*) é armazenada no controlador. Para implementar esses *bit maps* foi definido o componente bits que se passa a descrever.

### E.2.6.1 Bits

O componente bits foi definido para ser utilizado quando se pretender armazenar e efectuar operações sobre um conjunto de *bits*.

A classe que implementa este componente é a classe `bitmap` que tem duas variáveis internas, a variável `bmdim` e a variável `bitmap_store`. Na variável `bitmap_store` é armazenado o conjunto de bits e na variável `bmdim` guarda-se o valor correspondente ao número de bits armazenados na variável `bitmap_store`. São várias as operações que podem ser realizadas sobre o conjunto de bits. A função `obtem_proximo_bit_set` dá-nos a posição, a partir de uma dada posição, onde se encontra o próximo bit a um, na variável `bitmap_store`. A função `getp` dá-nos o valor do bit, na variável `bitmap_store`, de uma dada posição. A função `setp` coloca um determinado bit, na variável `bitmap_store`, a zero ou a um, como desejado. As funções `operator&` e `operator|` sobrepõem o

operador `&` e o operador `|` para serem usados com operandos do tipo `bitmap`, com o significado “natural”. A função `setdim` atribui à variável `bmdim` (número de bits armazenados na variável `bitmap_store`) um determinado valor. A função `operator char *` sobrepõe o operando `char *`, conseguindo-se deste modo obter uma cadeia de caracteres que representa o conjunto de bits armazenados na variável `bitmap_store`. Além das funções referidas existem dois construtores que diferem no número e tipo dos argumentos.



# Apêndice F

## Saídas da Simulação

A função `termina_estatisticas` da classe `stats_glob` é invocada pelo processo central no instante imediatamente antes da simulação terminar. A execução desta função faz com que sejam produzidos dois ficheiros de texto. Os resultados guardados num ficheiro são praticamente os guardados no outro, como será referido a seguir, diferindo principalmente no seu aspecto. Os valores armazenados num dos ficheiros foram-no através da função `recordscalar` (função já referida e que faz parte da biblioteca de classes de simulação do OMNeT++) e por essa razão esse ficheiro é formado por várias linhas tendo um formato adequado à importação para uma base de dados ou folha de cálculo e posterior tratamento nela. No outro ficheiro a informação é apresentada em forma de matriz o que torna a sua visualização simples. Este último ficheiro é útil quando se pretende ter uma visão global rápida dos resultados de uma simulação efectuada.

### F.1 Ficheiro de Escalares

O ficheiro de escalares, como já referido, é um ficheiro de texto e o seu nome deve ser especificado no ficheiro de configuração. No exemplo de um ficheiro de configuração apresentado no apêndice C foi especificado que, por exemplo, para a corrida 32 o nome do ficheiro de escalares será `mult-4--2.0-DAR0.sca` nessa corrida.

#### F.1.1 Estrutura do ficheiro de saída de escalares

O conteúdo deste ficheiro é um conjunto de linhas. Este ficheiro tem um formato próximo do formato do ficheiro de saída do COMNET. Em cada uma das linhas começa por aparecer a palavra *scalar* seguida de uma *string* (pelo facto de os valores terem sido armazenados nesse ficheiro pela função

`recordscalar()`), indicando essa *string* qual foi o módulo que produziu o ficheiro. Ainda na mesma linha segue-se uma segunda *string* e por fim um valor. Nesta segunda *string* aparecem grupos de valores que indicam o que representa o último valor da linha. O último valor de cada linha pode dizer respeito a um feixe, a informação ponto a ponto ou a informação global da rede. No caso de o último valor representar informação relativa a um feixe ou ponto a ponto, a *string* que o antecede possui cinco grupos de valores. O primeiro grupo é formado por um conjunto de caracteres que representam o significado específico do último valor da linha.

Os conjuntos de caracteres que aparecem são os indicados na tabela seguinte, com os significados correspondentes:

Rnof	tráfego oferecido ao feixe (número de chamadas oferecidas ao feixe)
Rnbl	bloqueio no feixe (número de chamadas bloqueadas no feixe)
Rntr	tráfego transportado no feixe (número de chamadas transportadas no feixe)
Rpbl	percentagem de bloqueio no feixe (percentagem de chamadas bloqueadas no feixe) – calculada como $\left(1 - \frac{Rntr}{Rnof}\right)$
Ptof{?}	tráfego oferecido ponto a ponto (número de chamadas oferecidas ponto a ponto)
Ppbl{?}	percentagem de bloqueio ponto a ponto (percentagem de chamadas bloqueadas ponto a ponto) – calculada como $\left(1 - \frac{Pptr\{?\}}{Ptof\{?\}}\right)$ , onde $Pptr\{?\}$ é o tráfego transportado ponto-a-ponto (valor recolhido mas não apresentado directamente)

Tab. F.1 – Valores recolhidos ponto a ponto ou por feixe – significado das abreviaturas.

Nos dois últimos conjuntos em vez do ? aparece o número da classe a que dizem respeito os valores. Desde o segundo grupo até ao último grupo os valores representam respectivamente: o número da central origem, o número da central destino, o nome do método de encaminhamento usado e o valor do factor de carga utilizado.

Por exemplo, se a segunda *string* for “Rntr 1 2 DAR0 1”, indica que o último valor da linha é o número de chamadas transportadas no feixe que liga a central 1 à central 2 quando foi usado o método de encaminhamento DAR com o tipo de reserva de circuitos 0 e com o factor de carga 1.

No caso de o último valor representar informação global da rede a *string* que o antecede possui apenas três grupos de valores. Os dois últimos grupos têm o mesmo significado que os dois últimos grupos do caso anterior. O primeiro grupo, é também aqui, formado por um conjunto de caracteres com os seguintes significados:

TpblClass{?}	<p>percentagem de bloqueio global para a classe ? – calculada como</p> $\left( \frac{\sum_{i,j} Ptof\{?\} - Ptr\{?\}}{\sum_{i,j} Ptof\{?\}} \right)$ <p>onde o somatório sobre <math>i</math> e <math>j</math> significa de todas as origens para todos os destinos</p>
Tnof	tráfego global oferecido por todas as classes (número de chamadas)
Tpbl	<p>percentagem de bloqueio global – calculada como</p> $\left( \frac{\sum_{cl} \sum_{i,j} (Ptof\{cl\} - Ptr\{cl\})}{\sum_{cl} \sum_{i,j} Ptof\{cl\}} \right)$ <p>onde <math>cl</math> itera sobre todas as classes</p>
TpblCirc	<p>percentagem de bloqueio global de circuitos (soma pesada), de acordo com o número de recursos, <math>R\{?\}</math>, necessários para cada classe – calculada como</p> $\left( \frac{\sum_{cl} \sum_{i,j} R\{cl\} (Ptof\{cl\} - Ptr\{cl\})}{\sum_{cl} \sum_{i,j} R\{cl\} Ptof\{cl\}} \right)$

Tab. F.2 – Valores globais recolhidos– significado das abreviaturas.

## F.1.2 Exemplo de um ficheiro de saída de escalares

```

run 33 "rede"
scalar "rede.central[0]"      "Rnof"    1 2   DAR1  2"    181615
scalar "rede.central[0]"      "Rnbl"    1 2   DAR1  2"    98833
scalar "rede.central[0]"      "Rntr"    1 2   DAR1  2"    82782
scalar "rede.central[0]"      "Rpbl"    1 2   DAR1  2"    0.54419
scalar "rede.central[0]"      "Rnof"    1 3   DAR1  2"    66108
scalar "rede.central[0]"      "Rnbl"    1 3   DAR1  2"    33756
scalar "rede.central[0]"      "Rntr"    1 3   DAR1  2"    32352
scalar "rede.central[0]"      "Rpbl"    1 3   DAR1  2"    0.510619
scalar "rede.central[0]"      "Rnof"    1 4   DAR1  2"    1.04478e+06
scalar "rede.central[0]"      "Rnbl"    1 4   DAR1  2"    250555
scalar "rede.central[0]"      "Rntr"    1 4   DAR1  2"    794229
scalar "rede.central[0]"      "Rpbl"    1 4   DAR1  2"    0.239815
scalar "rede.central[0]"      "Rnof"    1 5   DAR1  2"    136900
scalar "rede.central[0]"      "Rnbl"    1 5   DAR1  2"    63283
scalar "rede.central[0]"      "Rntr"    1 5   DAR1  2"    73617
scalar "rede.central[0]"      "Rpbl"    1 5   DAR1  2"    0.462257
scalar "rede.central[0]"      "Rnof"    1 6   DAR1  2"    166943
scalar "rede.central[0]"      "Rnbl"    1 6   DAR1  2"    72796
scalar "rede.central[0]"      "Rntr"    1 6   DAR1  2"    94147
scalar "rede.central[0]"      "Rpbl"    1 6   DAR1  2"    0.436053
scalar "rede.central[0]"      "Rnof"    2 3   DAR1  2"    134993
scalar "rede.central[0]"      "Rnbl"    2 3   DAR1  2"    26693
scalar "rede.central[0]"      "Rntr"    2 3   DAR1  2"    108300
scalar "rede.central[0]"      "Rpbl"    2 3   DAR1  2"    0.197736
scalar "rede.central[0]"      "Rnof"    2 4   DAR1  2"    417024
scalar "rede.central[0]"      "Rnbl"    2 4   DAR1  2"    190677
scalar "rede.central[0]"      "Rntr"    2 4   DAR1  2"    226347
scalar "rede.central[0]"      "Rpbl"    2 4   DAR1  2"    0.457233
scalar "rede.central[0]"      "Rnof"    2 5   DAR1  2"    312234
scalar "rede.central[0]"      "Rnbl"    2 5   DAR1  2"    120365
scalar "rede.central[0]"      "Rntr"    2 5   DAR1  2"    191869
scalar "rede.central[0]"      "Rpbl"    2 5   DAR1  2"    0.385496
scalar "rede.central[0]"      "Rnof"    2 6   DAR1  2"    335400
scalar "rede.central[0]"      "Rnbl"    2 6   DAR1  2"    131754
scalar "rede.central[0]"      "Rntr"    2 6   DAR1  2"    203646
scalar "rede.central[0]"      "Rpbl"    2 6   DAR1  2"    0.392826
scalar "rede.central[0]"      "Rnof"    3 4   DAR1  2"    130081
scalar "rede.central[0]"      "Rnbl"    3 4   DAR1  2"    112928
scalar "rede.central[0]"      "Rntr"    3 4   DAR1  2"    17153
scalar "rede.central[0]"      "Rpbl"    3 4   DAR1  2"    0.868136
scalar "rede.central[0]"      "Rnof"    3 5   DAR1  2"    68300
scalar "rede.central[0]"      "Rnbl"    3 5   DAR1  2"    15017
scalar "rede.central[0]"      "Rntr"    3 5   DAR1  2"    53283
scalar "rede.central[0]"      "Rpbl"    3 5   DAR1  2"    0.219868
scalar "rede.central[0]"      "Rnof"    3 6   DAR1  2"    85033
scalar "rede.central[0]"      "Rnbl"    3 6   DAR1  2"    53489
scalar "rede.central[0]"      "Rntr"    3 6   DAR1  2"    31544
scalar "rede.central[0]"      "Rpbl"    3 6   DAR1  2"    0.629038
scalar "rede.central[0]"      "Rnof"    4 5   DAR1  2"    372845
scalar "rede.central[0]"      "Rnbl"    4 5   DAR1  2"    123857
scalar "rede.central[0]"      "Rntr"    4 5   DAR1  2"    248988
scalar "rede.central[0]"      "Rpbl"    4 5   DAR1  2"    0.332194
scalar "rede.central[0]"      "Rnof"    4 6   DAR1  2"    370145
scalar "rede.central[0]"      "Rnbl"    4 6   DAR1  2"    193263
scalar "rede.central[0]"      "Rntr"    4 6   DAR1  2"    176882
scalar "rede.central[0]"      "Rpbl"    4 6   DAR1  2"    0.522128
scalar "rede.central[0]"      "Rnof"    5 6   DAR1  2"    472810
scalar "rede.central[0]"      "Rnbl"    5 6   DAR1  2"    132
scalar "rede.central[0]"      "Rntr"    5 6   DAR1  2"    472678
scalar "rede.central[0]"      "Rpbl"    5 6   DAR1  2"    0.000279182
scalar "rede.central[0]"      "Ptof{0}" 1 1   DAR1  2"    0
scalar "rede.central[0]"      "Ppbl{0}" 1 1   DAR1  2"    0
scalar "rede.central[0]"      "Ptof{0}" 1 2   DAR1  2"    39389
scalar "rede.central[0]"      "Ppbl{0}" 1 2   DAR1  2"    0.312828
scalar "rede.central[0]"      "Ptof{0}" 1 3   DAR1  2"    9990
scalar "rede.central[0]"      "Ppbl{0}" 1 3   DAR1  2"    0.182883
scalar "rede.central[0]"      "Ptof{0}" 1 4   DAR1  2"    370950
scalar "rede.central[0]"      "Ppbl{0}" 1 4   DAR1  2"    0.214493
scalar "rede.central[0]"      "Ptof{0}" 1 5   DAR1  2"    29235

```

scalar "rede.central[0]"	"Ppbl{0}	1	5	DAR1	2"	0.161279
scalar "rede.central[0]"	"Ptof{0}	1	6	DAR1	2"	41612
scalar "rede.central[0]"	"Ppbl{0}	1	6	DAR1	2"	0.338316
scalar "rede.central[0]"	"Ptof{0}	2	1	DAR1	2"	39682
scalar "rede.central[0]"	"Ppbl{0}	2	1	DAR1	2"	0.125775
scalar "rede.central[0]"	"Ptof{0}	2	2	DAR1	2"	0
scalar "rede.central[0]"	"Ppbl{0}	2	2	DAR1	2"	0
scalar "rede.central[0]"	"Ptof{0}	2	3	DAR1	2"	36145
scalar "rede.central[0]"	"Ppbl{0}	2	3	DAR1	2"	0.060285
scalar "rede.central[0]"	"Ptof{0}	2	4	DAR1	2"	146034
scalar "rede.central[0]"	"Ppbl{0}	2	4	DAR1	2"	0.271129
scalar "rede.central[0]"	"Ptof{0}	2	5	DAR1	2"	111252
scalar "rede.central[0]"	"Ppbl{0}	2	5	DAR1	2"	0.129445
scalar "rede.central[0]"	"Ptof{0}	2	6	DAR1	2"	118724
scalar "rede.central[0]"	"Ppbl{0}	2	6	DAR1	2"	0.190703
scalar "rede.central[0]"	"Ptof{0}	3	1	DAR1	2"	9920
scalar "rede.central[0]"	"Ppbl{0}	3	1	DAR1	2"	0.161492
scalar "rede.central[0]"	"Ptof{0}	3	2	DAR1	2"	36245
scalar "rede.central[0]"	"Ppbl{0}	3	2	DAR1	2"	0.0646434
scalar "rede.central[0]"	"Ptof{0}	3	3	DAR1	2"	0
scalar "rede.central[0]"	"Ppbl{0}	3	3	DAR1	2"	0
scalar "rede.central[0]"	"Ptof{0}	3	4	DAR1	2"	17365
scalar "rede.central[0]"	"Ppbl{0}	3	4	DAR1	2"	0.575641
scalar "rede.central[0]"	"Ptof{0}	3	5	DAR1	2"	9971
scalar "rede.central[0]"	"Ppbl{0}	3	5	DAR1	2"	0.0914652
scalar "rede.central[0]"	"Ptof{0}	3	6	DAR1	2"	19110
scalar "rede.central[0]"	"Ppbl{0}	3	6	DAR1	2"	0.202041
scalar "rede.central[0]"	"Ptof{0}	4	1	DAR1	2"	371677
scalar "rede.central[0]"	"Ppbl{0}	4	1	DAR1	2"	0.209712
scalar "rede.central[0]"	"Ptof{0}	4	2	DAR1	2"	146132
scalar "rede.central[0]"	"Ppbl{0}	4	2	DAR1	2"	0.387157
scalar "rede.central[0]"	"Ptof{0}	4	3	DAR1	2"	17068
scalar "rede.central[0]"	"Ppbl{0}	4	3	DAR1	2"	0.53697
scalar "rede.central[0]"	"Ptof{0}	4	4	DAR1	2"	0
scalar "rede.central[0]"	"Ppbl{0}	4	4	DAR1	2"	0
scalar "rede.central[0]"	"Ptof{0}	4	5	DAR1	2"	114225
scalar "rede.central[0]"	"Ppbl{0}	4	5	DAR1	2"	0.127371
scalar "rede.central[0]"	"Ptof{0}	4	6	DAR1	2"	119208
scalar "rede.central[0]"	"Ppbl{0}	4	6	DAR1	2"	0.37656
scalar "rede.central[0]"	"Ptof{0}	5	1	DAR1	2"	29323
scalar "rede.central[0]"	"Ppbl{0}	5	1	DAR1	2"	0.150599
scalar "rede.central[0]"	"Ptof{0}	5	2	DAR1	2"	110691
scalar "rede.central[0]"	"Ppbl{0}	5	2	DAR1	2"	0.285994
scalar "rede.central[0]"	"Ptof{0}	5	3	DAR1	2"	9559
scalar "rede.central[0]"	"Ppbl{0}	5	3	DAR1	2"	0.093629
scalar "rede.central[0]"	"Ptof{0}	5	4	DAR1	2"	114414
scalar "rede.central[0]"	"Ppbl{0}	5	4	DAR1	2"	0.128944
scalar "rede.central[0]"	"Ptof{0}	5	5	DAR1	2"	0
scalar "rede.central[0]"	"Ppbl{0}	5	5	DAR1	2"	0
scalar "rede.central[0]"	"Ptof{0}	5	6	DAR1	2"	182611
scalar "rede.central[0]"	"Ppbl{0}	5	6	DAR1	2"	0.000191664
scalar "rede.central[0]"	"Ptof{0}	6	1	DAR1	2"	42054
scalar "rede.central[0]"	"Ppbl{0}	6	1	DAR1	2"	0.2001
scalar "rede.central[0]"	"Ptof{0}	6	2	DAR1	2"	118645
scalar "rede.central[0]"	"Ppbl{0}	6	2	DAR1	2"	0.309233
scalar "rede.central[0]"	"Ptof{0}	6	3	DAR1	2"	19216
scalar "rede.central[0]"	"Ppbl{0}	6	3	DAR1	2"	0.191091
scalar "rede.central[0]"	"Ptof{0}	6	4	DAR1	2"	120208
scalar "rede.central[0]"	"Ppbl{0}	6	4	DAR1	2"	0.372396
scalar "rede.central[0]"	"Ptof{0}	6	5	DAR1	2"	183042
scalar "rede.central[0]"	"Ppbl{0}	6	5	DAR1	2"	0.000202139
scalar "rede.central[0]"	"Ptof{0}	6	6	DAR1	2"	0
scalar "rede.central[0]"	"Ppbl{0}	6	6	DAR1	2"	0
scalar "rede.central[0]"	"TpblClas{0}			DAR1	2"	0.206126
scalar "rede.central[0]"	"Ptof{1}	1	1	DAR1	2"	0
scalar "rede.central[0]"	"Ppbl{1}	1	1	DAR1	2"	0
scalar "rede.central[0]"	"Ptof{1}	1	2	DAR1	2"	7953
scalar "rede.central[0]"	"Ppbl{1}	1	2	DAR1	2"	0.293851
scalar "rede.central[0]"	"Ptof{1}	1	3	DAR1	2"	1989
scalar "rede.central[0]"	"Ppbl{1}	1	3	DAR1	2"	0.255405
scalar "rede.central[0]"	"Ptof{1}	1	4	DAR1	2"	74534
scalar "rede.central[0]"	"Ppbl{1}	1	4	DAR1	2"	0.24362
scalar "rede.central[0]"	"Ptof{1}	1	5	DAR1	2"	5905

scalar "rede.central[0]"	"Ppbl{1}	1	5	DAR1	2"	0.104488
scalar "rede.central[0]"	"Ptof{1}	1	6	DAR1	2"	8408
scalar "rede.central[0]"	"Ppbl{1}	1	6	DAR1	2"	0.277236
scalar "rede.central[0]"	"Ptof{1}	2	1	DAR1	2"	7919
scalar "rede.central[0]"	"Ppbl{1}	2	1	DAR1	2"	0.22023
scalar "rede.central[0]"	"Ptof{1}	2	2	DAR1	2"	0
scalar "rede.central[0]"	"Ppbl{1}	2	2	DAR1	2"	0
scalar "rede.central[0]"	"Ptof{1}	2	3	DAR1	2"	7318
scalar "rede.central[0]"	"Ppbl{1}	2	3	DAR1	2"	0.125991
scalar "rede.central[0]"	"Ptof{1}	2	4	DAR1	2"	29290
scalar "rede.central[0]"	"Ppbl{1}	2	4	DAR1	2"	0.383851
scalar "rede.central[0]"	"Ptof{1}	2	5	DAR1	2"	22124
scalar "rede.central[0]"	"Ppbl{1}	2	5	DAR1	2"	0.217275
scalar "rede.central[0]"	"Ptof{1}	2	6	DAR1	2"	23600
scalar "rede.central[0]"	"Ppbl{1}	2	6	DAR1	2"	0.270339
scalar "rede.central[0]"	"Ptof{1}	3	1	DAR1	2"	2003
scalar "rede.central[0]"	"Ppbl{1}	3	1	DAR1	2"	0.223165
scalar "rede.central[0]"	"Ptof{1}	3	2	DAR1	2"	7278
scalar "rede.central[0]"	"Ppbl{1}	3	2	DAR1	2"	0.12476
scalar "rede.central[0]"	"Ptof{1}	3	3	DAR1	2"	0
scalar "rede.central[0]"	"Ppbl{1}	3	3	DAR1	2"	0
scalar "rede.central[0]"	"Ptof{1}	3	4	DAR1	2"	3376
scalar "rede.central[0]"	"Ppbl{1}	3	4	DAR1	2"	0.509182
scalar "rede.central[0]"	"Ptof{1}	3	5	DAR1	2"	1971
scalar "rede.central[0]"	"Ppbl{1}	3	5	DAR1	2"	0.17859
scalar "rede.central[0]"	"Ptof{1}	3	6	DAR1	2"	3817
scalar "rede.central[0]"	"Ppbl{1}	3	6	DAR1	2"	0.226356
scalar "rede.central[0]"	"Ptof{1}	4	1	DAR1	2"	74453
scalar "rede.central[0]"	"Ppbl{1}	4	1	DAR1	2"	0.03245
scalar "rede.central[0]"	"Ptof{1}	4	2	DAR1	2"	29103
scalar "rede.central[0]"	"Ppbl{1}	4	2	DAR1	2"	0.384359
scalar "rede.central[0]"	"Ptof{1}	4	3	DAR1	2"	3393
scalar "rede.central[0]"	"Ppbl{1}	4	3	DAR1	2"	0.504863
scalar "rede.central[0]"	"Ptof{1}	4	4	DAR1	2"	0
scalar "rede.central[0]"	"Ppbl{1}	4	4	DAR1	2"	0
scalar "rede.central[0]"	"Ptof{1}	4	5	DAR1	2"	22707
scalar "rede.central[0]"	"Ppbl{1}	4	5	DAR1	2"	0.122077
scalar "rede.central[0]"	"Ptof{1}	4	6	DAR1	2"	23806
scalar "rede.central[0]"	"Ppbl{1}	4	6	DAR1	2"	0.182937
scalar "rede.central[0]"	"Ptof{1}	5	1	DAR1	2"	5847
scalar "rede.central[0]"	"Ppbl{1}	5	1	DAR1	2"	0.0867111
scalar "rede.central[0]"	"Ptof{1}	5	2	DAR1	2"	22056
scalar "rede.central[0]"	"Ppbl{1}	5	2	DAR1	2"	0.224383
scalar "rede.central[0]"	"Ptof{1}	5	3	DAR1	2"	1987
scalar "rede.central[0]"	"Ppbl{1}	5	3	DAR1	2"	0.1384
scalar "rede.central[0]"	"Ptof{1}	5	4	DAR1	2"	22922
scalar "rede.central[0]"	"Ppbl{1}	5	4	DAR1	2"	0.123593
scalar "rede.central[0]"	"Ptof{1}	5	5	DAR1	2"	0
scalar "rede.central[0]"	"Ppbl{1}	5	5	DAR1	2"	0
scalar "rede.central[0]"	"Ptof{1}	5	6	DAR1	2"	36446
scalar "rede.central[0]"	"Ppbl{1}	5	6	DAR1	2"	0.000164627
scalar "rede.central[0]"	"Ptof{1}	6	1	DAR1	2"	8322
scalar "rede.central[0]"	"Ppbl{1}	6	1	DAR1	2"	0.0668109
scalar "rede.central[0]"	"Ptof{1}	6	2	DAR1	2"	23855
scalar "rede.central[0]"	"Ppbl{1}	6	2	DAR1	2"	0.27206
scalar "rede.central[0]"	"Ptof{1}	6	3	DAR1	2"	3848
scalar "rede.central[0]"	"Ppbl{1}	6	3	DAR1	2"	0.201143
scalar "rede.central[0]"	"Ptof{1}	6	4	DAR1	2"	24032
scalar "rede.central[0]"	"Ppbl{1}	6	4	DAR1	2"	0.175308
scalar "rede.central[0]"	"Ptof{1}	6	5	DAR1	2"	36483
scalar "rede.central[0]"	"Ppbl{1}	6	5	DAR1	2"	0.00035633
scalar "rede.central[0]"	"Ptof{1}	6	6	DAR1	2"	0
scalar "rede.central[0]"	"Ppbl{1}	6	6	DAR1	2"	0
scalar "rede.central[0]"	"TpblClas{1}			DAR1	2"	0.176307
scalar "rede.central[0]"	"Ptof{2}	1	1	DAR1	2"	0
scalar "rede.central[0]"	"Ppbl{2}	1	1	DAR1	2"	0
scalar "rede.central[0]"	"Ptof{2}	1	2	DAR1	2"	8007
scalar "rede.central[0]"	"Ppbl{2}	1	2	DAR1	2"	0.665543
scalar "rede.central[0]"	"Ptof{2}	1	3	DAR1	2"	2037
scalar "rede.central[0]"	"Ppbl{2}	1	3	DAR1	2"	0.767796
scalar "rede.central[0]"	"Ptof{2}	1	4	DAR1	2"	74976
scalar "rede.central[0]"	"Ppbl{2}	1	4	DAR1	2"	0.157477
scalar "rede.central[0]"	"Ptof{2}	1	5	DAR1	2"	5913



scalar "rede.central[0]"	"Ppbl{3}	1	5	DAR1	2"	0.980822
scalar "rede.central[0]"	"Ptof{3}	1	6	DAR1	2"	1076
scalar "rede.central[0]"	"Ppbl{3}	1	6	DAR1	2"	0.939591
scalar "rede.central[0]"	"Ptof{3}	2	1	DAR1	2"	0
scalar "rede.central[0]"	"Ppbl{3}	2	1	DAR1	2"	0
scalar "rede.central[0]"	"Ptof{3}	2	2	DAR1	2"	0
scalar "rede.central[0]"	"Ppbl{3}	2	2	DAR1	2"	0
scalar "rede.central[0]"	"Ptof{3}	2	3	DAR1	2"	0
scalar "rede.central[0]"	"Ppbl{3}	2	3	DAR1	2"	0
scalar "rede.central[0]"	"Ptof{3}	2	4	DAR1	2"	0
scalar "rede.central[0]"	"Ppbl{3}	2	4	DAR1	2"	0
scalar "rede.central[0]"	"Ptof{3}	2	5	DAR1	2"	0
scalar "rede.central[0]"	"Ppbl{3}	2	5	DAR1	2"	0
scalar "rede.central[0]"	"Ptof{3}	2	6	DAR1	2"	0
scalar "rede.central[0]"	"Ppbl{3}	2	6	DAR1	2"	0
scalar "rede.central[0]"	"Ptof{3}	3	1	DAR1	2"	0
scalar "rede.central[0]"	"Ppbl{3}	3	1	DAR1	2"	0
scalar "rede.central[0]"	"Ptof{3}	3	2	DAR1	2"	0
scalar "rede.central[0]"	"Ppbl{3}	3	2	DAR1	2"	0
scalar "rede.central[0]"	"Ptof{3}	3	3	DAR1	2"	0
scalar "rede.central[0]"	"Ppbl{3}	3	3	DAR1	2"	0
scalar "rede.central[0]"	"Ptof{3}	3	4	DAR1	2"	0
scalar "rede.central[0]"	"Ppbl{3}	3	4	DAR1	2"	0
scalar "rede.central[0]"	"Ptof{3}	3	5	DAR1	2"	0
scalar "rede.central[0]"	"Ppbl{3}	3	5	DAR1	2"	0
scalar "rede.central[0]"	"Ptof{3}	3	6	DAR1	2"	0
scalar "rede.central[0]"	"Ppbl{3}	3	6	DAR1	2"	0
scalar "rede.central[0]"	"Ptof{3}	4	1	DAR1	2"	0
scalar "rede.central[0]"	"Ppbl{3}	4	1	DAR1	2"	0
scalar "rede.central[0]"	"Ptof{3}	4	2	DAR1	2"	0
scalar "rede.central[0]"	"Ppbl{3}	4	2	DAR1	2"	0
scalar "rede.central[0]"	"Ptof{3}	4	3	DAR1	2"	0
scalar "rede.central[0]"	"Ppbl{3}	4	3	DAR1	2"	0
scalar "rede.central[0]"	"Ptof{3}	4	4	DAR1	2"	0
scalar "rede.central[0]"	"Ppbl{3}	4	4	DAR1	2"	0
scalar "rede.central[0]"	"Ptof{3}	4	5	DAR1	2"	0
scalar "rede.central[0]"	"Ppbl{3}	4	5	DAR1	2"	0
scalar "rede.central[0]"	"Ptof{3}	4	6	DAR1	2"	0
scalar "rede.central[0]"	"Ppbl{3}	4	6	DAR1	2"	0
scalar "rede.central[0]"	"Ptof{3}	5	1	DAR1	2"	0
scalar "rede.central[0]"	"Ppbl{3}	5	1	DAR1	2"	0
scalar "rede.central[0]"	"Ptof{3}	5	2	DAR1	2"	0
scalar "rede.central[0]"	"Ppbl{3}	5	2	DAR1	2"	0
scalar "rede.central[0]"	"Ptof{3}	5	3	DAR1	2"	0
scalar "rede.central[0]"	"Ppbl{3}	5	3	DAR1	2"	0
scalar "rede.central[0]"	"Ptof{3}	5	4	DAR1	2"	0
scalar "rede.central[0]"	"Ppbl{3}	5	4	DAR1	2"	0
scalar "rede.central[0]"	"Ptof{3}	5	5	DAR1	2"	0
scalar "rede.central[0]"	"Ppbl{3}	5	5	DAR1	2"	0
scalar "rede.central[0]"	"Ptof{3}	5	6	DAR1	2"	0
scalar "rede.central[0]"	"Ppbl{3}	5	6	DAR1	2"	0
scalar "rede.central[0]"	"Ptof{3}	6	1	DAR1	2"	0
scalar "rede.central[0]"	"Ppbl{3}	6	1	DAR1	2"	0
scalar "rede.central[0]"	"Ptof{3}	6	2	DAR1	2"	0
scalar "rede.central[0]"	"Ppbl{3}	6	2	DAR1	2"	0
scalar "rede.central[0]"	"Ptof{3}	6	3	DAR1	2"	0
scalar "rede.central[0]"	"Ppbl{3}	6	3	DAR1	2"	0
scalar "rede.central[0]"	"Ptof{3}	6	4	DAR1	2"	0
scalar "rede.central[0]"	"Ppbl{3}	6	4	DAR1	2"	0
scalar "rede.central[0]"	"Ptof{3}	6	5	DAR1	2"	0
scalar "rede.central[0]"	"Ppbl{3}	6	5	DAR1	2"	0
scalar "rede.central[0]"	"Ptof{3}	6	6	DAR1	2"	0
scalar "rede.central[0]"	"Ppbl{3}	6	6	DAR1	2"	0
scalar "rede.central[0]"	"TpblClas{3}			DAR1	2"	0.556932
scalar "rede.central[0]"	"Tnof			DAR1	2"	3.39229e+06
scalar "rede.central[0]"	"Tpbl			DAR1	2"	0.203586
scalar "rede.central[0]"	"TpblCirc			DAR1	2"	0.225608

Fig. F-1 – Conteúdo de um ficheiro de resultados escalares correspondente ao sistema 3.

Este exemplo corresponde à corrida 33 do ficheiro *ini* apresentado no apêndice C.

## F.2 Ficheiro de tabelas (ficheiro txt)

O ficheiro de tabelas, como já referido, é também um ficheiro de texto e o seu nome deve ser especificado no ficheiro de configuração. No exemplo de um ficheiro de configuração apresentado no apêndice C foi especificado que, por exemplo, para a corrida 32 o nome do ficheiro de tabelas será `mult-4--2.0-DAR0.txt` nessa corrida.

### F.2.1 Estrutura do ficheiro de saída contendo tabelas

Este ficheiro possui informação de todos os feixes da rede, informação relativa a todos os pares origem destino. O conteúdo do ficheiro são duas matrizes que representam o tráfego oferecido e transportado nos feixes respectivamente, duas matrizes que representam o tráfego de circuitos oferecido e transportado nos feixes respectivamente e ainda mais tantos conjuntos de duas matrizes tantos os números de classes de tráfego. Cada par de matrizes representa o tráfego oferecido e transportado ponto a ponto para a classe correspondente.

Além da informação referida o ficheiro contém ainda informação relativa a alguns parâmetros da simulação efectuada. Esses parâmetros incluem: o nome do ficheiro de topologia (este ficheiro está descrito no apêndice C) que foi usado como entrada de valores para a simulação, o método de encaminhamento que foi simulado, o factor de carga utilizado, os valores dos instantes utilizados para iniciar a recolha de valores e para terminar a simulação, e ainda os quatro valores para cada um dos quatro tipos possíveis de duração de chamada.

## F.2.2 Exemplo de um ficheiro de saída contendo tabelas

Parametros da Simulacao:

```

duracao da chamada [1]: 60
duracao da chamada [2]: 60
duracao da chamada [3]: 60
duracao da chamada [4]: 60
inicio da recolha: 86400
fim da recolha: 172800
factor de carga: 2
tipo de routing: DAR
ficheiro de topologia: red20000808-4.txt
    
```

trafego oferecido aos ramos

origem	destino	->				
V	1	2	3	4	5	6
1	0	181615	66108	1044784	136900	166943
2	181615	0	134993	417024	312234	335400
3	66108	134993	0	130081	68300	85033
4	1044784	417024	130081	0	372845	370145
5	136900	312234	68300	372845	0	472810
6	166943	335400	85033	370145	472810	0

trafego transportado nos ramos

origem	destino	->				
V	1	2	3	4	5	6
1	0	82782	32352	794229	73617	94147
2	82782	0	108300	226347	191869	203646
3	32352	108300	0	17153	53283	31544
4	794229	226347	17153	0	248988	176882
5	73617	191869	53283	248988	0	472678
6	94147	203646	31544	176882	472678	0

trafego (circuitos) oferecido aos ramos

origem	destino	->				
V	1	2	3	4	5	6
1	0	334460	156650	1813925	270660	320006
2	334460	0	161199	499443	370645	398019
3	156650	161199	0	154551	81411	101566
4	1813925	499443	154551	0	445758	449656
5	270660	370645	81411	445758	0	561110
6	320006	398019	101566	449656	561110	0

trafego (circuitos) transportado nos ramos

origem	destino	->				
V	1	2	3	4	5	6
1	0	111655	41419	1371308	105522	153489
2	111655	0	128196	258664	224751	235460
3	41419	128196	0	19774	62110	38570
4	1371308	258664	19774	0	298130	212555
5	105522	224751	62110	298130	0	560942
6	153489	235460	38570	212555	560942	0

trafego oferecido ponto a ponto{0}

origem	destino	->				
V	1	2	3	4	5	6
1	0	39389	9990	370950	29235	41612
2	39682	0	36145	146034	111252	118724
3	9920	36245	0	17365	9971	19110
4	371677	146132	17068	0	114225	119208
5	29323	110691	9559	114414	0	182611
6	42054	118645	19216	120208	183042	0

trafego transportado ponto a ponto{0}

origem	destino	->				
V	1	2	3	4	5	6
1	0	27067	8163	291384	24520	27534
2	34691	0	33966	106440	96851	96083
3	8318	33902	0	7369	9059	15249

4	293732	89556	7903	0	99676	74319
5	24907	79034	8664	99661	0	182576
6	33639	81956	15544	75443	183005	0

trafego oferecido ponto a ponto{1}

origem	destino	->				
V	1	2	3	4	5	6
1	0	7953	1989	74534	5905	8408
2	7919	0	7318	29290	22124	23600
3	2003	7278	0	3376	1971	3817
4	74453	29103	3393	0	22707	23806
5	5847	22056	1987	22922	0	36446
6	8322	23855	3848	24032	36483	0

trafego transportado ponto a ponto{1}

origem	destino	->				
V	1	2	3	4	5	6
1	0	5616	1481	56376	5288	6077
2	6175	0	6396	18047	17317	17220
3	1556	6370	0	1657	1619	2953
4	72037	17917	1680	0	19935	19451
5	5340	17107	1712	20089	0	36440
6	7766	17365	3074	19819	36470	0

trafego oferecido ponto a ponto{2}

origem	destino	->				
V	1	2	3	4	5	6
1	0	8007	2037	74976	5913	8489
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

trafego transportado ponto a ponto{2}

origem	destino	->				
V	1	2	3	4	5	6
1	0	2678	473	63169	3527	5753
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

trafego oferecido ponto a ponto{3}

origem	destino	->				
V	1	2	3	4	5	6
1	0	1057	245	9319	730	1076
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

trafego transportado ponto a ponto{3}

origem	destino	->				
V	1	2	3	4	5	6
1	0	0	0	5427	14	65
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

Fig. F-2 – Conteúdo de um ficheiro de resultados matriciais correspondente ao sistema 3.

Este exemplo apresenta os resultados correspondentes à mesma corrida que o exemplo do ficheiro apresentado na secção anterior.



# Apêndice G

## Estudo Estatístico da Confiança associada aos Resultados Simulacionais

Neste apêndice são apresentadas as tabelas correspondentes aos intervalos de confiança para cada um dos valores usados no capítulo 6. Os intervalos de confiança foram determinados de acordo com a fórmula 5.14 apresentada no capítulo 5. O grau de confiança utilizado foi de 95% ao qual corresponde um alfa de 5%. Em cada tabela o valor de alfa usado é apresentado no seu cabeçalho.

Para cada tabela, é apresentado para cada valor o mínimo, máximo e ponto médio correspondente ao intervalo de confiança. O limite inferior de cada intervalo foi, quando necessário, truncado a 0, uma vez que percentagens de bloqueio negativas não são admissíveis. Uma medida da dispersão dos valores, ou seja, o valor relativo da dimensão do intervalo é também apresentado (calculado como a dimensão do semi intervalo dividido pelo valor).

De acordo com os resultados apresentados nas diversas tabelas, pode-se verificar que a dimensão do intervalo de confiança é em geral aceitável. Os valores apresentam taxas de dispersão muito elevadas apenas quando representam o resultado de acontecimentos pouco frequentes, e cuja aproximação com um intervalo de confiança menor implicaria um tempo simulado muito mais elevado. Tal ocorre quase exclusivamente em casos em que a congestão é muito reduzida, e mesmo nesses casos geralmente essa dispersão não torna impossível determinar quais os melhores métodos de encaminhamento a utilizar para esses casos (embora uma análise com tempos de simulação bastante mais elevados seja provavelmente prudente para confirmar tais afirmações).

## G.1 Intervalos de confiança da experiência com o Sistema 1

reserva	(All)		alfa	5%	
item	Tpbl				
Average of valor	metodo	Limite Inferior	Média	Limite Superior	Conf (Rel)
carga					
1	DAR	0,131%	0,141%	0,151%	7,2%
	DCR	0,060%	0,064%	0,068%	6,8%
	DIRECTO	2,292%	2,332%	2,372%	1,7%
	FAR	1,561%	1,597%	1,633%	2,2%
	RTNR	0,001%	0,002%	0,003%	49,7%
1,1	DAR	0,685%	0,722%	0,758%	5,0%
	DCR	0,500%	0,528%	0,557%	5,5%
	DIRECTO	3,962%	4,018%	4,074%	1,4%
	FAR	2,765%	2,822%	2,879%	2,0%
	RTNR	0,055%	0,062%	0,069%	11,8%
1,2	DAR	1,918%	1,966%	2,014%	2,4%
	DCR	1,716%	1,762%	1,808%	2,6%
	DIRECTO	6,226%	6,295%	6,364%	1,1%
	FAR	4,709%	4,760%	4,811%	1,1%
	RTNR	0,397%	0,433%	0,469%	8,2%
1,3	DAR	4,131%	4,207%	4,284%	1,8%
	DCR	3,953%	4,036%	4,119%	2,1%
	DIRECTO	9,334%	9,439%	9,544%	1,1%
	FAR	7,497%	7,579%	7,662%	1,1%
	RTNR	2,544%	2,652%	2,759%	4,0%
1,4	DAR	7,267%	7,338%	7,409%	1,0%
	DCR	7,069%	7,144%	7,220%	1,1%
	DIRECTO	12,775%	12,852%	12,929%	0,6%
	FAR	10,560%	10,626%	10,691%	0,6%
	RTNR	6,143%	6,225%	6,307%	1,3%
1,5	DAR	10,741%	10,785%	10,830%	0,4%
	DCR	10,497%	10,549%	10,602%	0,5%
	DIRECTO	16,061%	16,117%	16,173%	0,3%
	FAR	13,479%	13,516%	13,554%	0,3%
	RTNR	9,987%	10,047%	10,106%	0,6%
1,6	DAR	14,245%	14,304%	14,364%	0,4%
	DCR	14,003%	14,061%	14,119%	0,4%
	DIRECTO	19,205%	19,245%	19,285%	0,2%
	FAR	16,239%	16,288%	16,337%	0,3%
	RTNR	13,711%	13,758%	13,804%	0,3%
1,7	DAR	17,571%	17,638%	17,705%	0,4%
	DCR	17,357%	17,416%	17,474%	0,3%
	DIRECTO	21,997%	22,066%	22,136%	0,3%
	FAR	18,818%	18,883%	18,947%	0,3%
	RTNR	17,150%	17,203%	17,256%	0,3%
1,8	DAR	20,716%	20,747%	20,778%	0,1%
	DCR	20,513%	20,547%	20,581%	0,2%
	DIRECTO	24,596%	24,636%	24,677%	0,2%
	FAR	21,413%	21,442%	21,471%	0,1%
	RTNR	20,356%	20,395%	20,434%	0,2%
1,9	DAR	23,634%	23,680%	23,725%	0,2%
	DCR	23,453%	23,502%	23,551%	0,2%
	DIRECTO	26,982%	27,022%	27,062%	0,1%
	FAR	24,018%	24,056%	24,093%	0,2%
	RTNR	23,341%	23,397%	23,454%	0,2%
2	DAR	26,347%	26,412%	26,478%	0,2%
	DCR	26,186%	26,253%	26,321%	0,3%
	DIRECTO	29,165%	29,221%	29,278%	0,2%
	FAR	26,570%	26,633%	26,697%	0,2%
	RTNR	26,125%	26,183%	26,242%	0,2%

Tab. G-1 – Intervalos de confiança para a probabilidade de bloqueio global para todos os métodos de encaminhamentos implementados (sistema 1).

item	Tpbl			alfa	5%	
Average of valor			Limite Inferior	Média	Limite Superior	Conf (Rel)
carga	reserva	metodo				
1	0	DAR	0,131%	0,141%	0,151%	7,2%
		DCR	0,060%	0,064%	0,068%	6,8%
		RTNR	0,001%	0,002%	0,003%	49,7%
	1	DAR	0,048%	0,050%	0,053%	4,9%
		DCR	0,026%	0,029%	0,031%	8,1%
		RTNR	0,001%	0,003%	0,006%	79,2%
1,1	0	DAR	0,685%	0,722%	0,758%	5,0%
		DCR	0,500%	0,528%	0,557%	5,5%
		RTNR	0,055%	0,062%	0,069%	11,8%
	1	DAR	0,261%	0,280%	0,300%	6,9%
		DCR	0,185%	0,202%	0,219%	8,3%
		RTNR	0,054%	0,062%	0,071%	13,7%
1,2	0	DAR	1,918%	1,966%	2,014%	2,4%
		DCR	1,716%	1,762%	1,808%	2,6%
		RTNR	0,397%	0,433%	0,469%	8,2%
	1	DAR	0,949%	0,983%	1,018%	3,5%
		DCR	0,844%	0,887%	0,931%	4,9%
		RTNR	0,404%	0,441%	0,478%	8,4%
1,3	0	DAR	4,131%	4,207%	4,284%	1,8%
		DCR	3,953%	4,036%	4,119%	2,1%
		RTNR	2,544%	2,652%	2,759%	4,0%
	1	DAR	3,037%	3,150%	3,264%	3,6%
		DCR	2,859%	2,956%	3,054%	3,3%
		RTNR	2,552%	2,671%	2,789%	4,4%
1,4	0	DAR	7,267%	7,338%	7,409%	1,0%
		DCR	7,069%	7,144%	7,220%	1,1%
		RTNR	6,143%	6,225%	6,307%	1,3%
	1	DAR	6,569%	6,679%	6,790%	1,7%
		DCR	6,309%	6,414%	6,520%	1,6%
		RTNR	6,158%	6,237%	6,315%	1,3%
1,5	0	DAR	10,741%	10,785%	10,830%	0,4%
		DCR	10,497%	10,549%	10,602%	0,5%
		RTNR	9,987%	10,047%	10,106%	0,6%
	1	DAR	10,378%	10,420%	10,461%	0,4%
		DCR	10,112%	10,181%	10,249%	0,7%
		RTNR	9,993%	10,051%	10,109%	0,6%
1,6	0	DAR	14,245%	14,304%	14,364%	0,4%
		DCR	14,003%	14,061%	14,119%	0,4%
		RTNR	13,711%	13,758%	13,804%	0,3%
	1	DAR	14,014%	14,079%	14,145%	0,5%
		DCR	13,799%	13,863%	13,927%	0,5%
		RTNR	13,705%	13,759%	13,813%	0,4%

Tab. G-2 – Intervalos de confiança para a probabilidade de bloqueio global nos métodos de encaminhamento dinâmicos para dois tipos de reserva (sistema 1).

item	Ppbl(0)
carga	1,4
reserva	0

alfa 5%

Average of valor			Limite Inferior	Média	Limite Superior	Conf (Rel)			
metodo	origem	destino							
DAR	1	1	1	0,000%	0,000%	0,000%	0,0%		
			2	6,346%	6,461%	6,576%	1,8%		
			3	0,077%	0,110%	0,144%	30,0%		
			4	8,549%	8,766%	8,984%	2,5%		
			5	0,554%	0,631%	0,708%	12,2%		
			6	2,712%	2,820%	2,928%	3,8%		
	2	2	1	1	6,285%	6,533%	6,780%	3,8%	
				2	0,000%	0,000%	0,000%	0,0%	
				3	0,000%	0,000%	0,000%	0,0%	
				4	27,389%	27,776%	28,162%	1,4%	
				5	0,905%	0,939%	0,973%	3,6%	
				6	14,677%	14,833%	14,989%	1,1%	
	3	3	1	1	0,054%	0,091%	0,129%	41,0%	
				2	0,000%	0,000%	0,000%	0,0%	
				3	0,000%	0,000%	0,000%	0,0%	
				4	5,410%	5,801%	6,191%	6,7%	
				5	0,000%	0,000%	0,000%	0,0%	
				6	0,481%	0,528%	0,575%	8,9%	
	4	4	1	1	8,546%	8,785%	9,024%	2,7%	
				2	27,232%	27,678%	28,125%	1,6%	
				3	4,927%	5,694%	6,461%	13,5%	
				4	0,000%	0,000%	0,000%	0,0%	
				5	0,130%	0,157%	0,183%	17,0%	
				6	2,566%	2,810%	3,055%	8,7%	
	5	5	1	1	0,568%	0,614%	0,660%	7,5%	
				2	0,815%	0,897%	0,978%	9,1%	
				3	0,000%	0,000%	0,000%	0,0%	
				4	0,117%	0,162%	0,206%	27,7%	
				5	0,000%	0,000%	0,000%	0,0%	
				6	0,000%	0,000%	0,000%	0,0%	
	6	6	1	1	2,740%	2,880%	3,020%	4,9%	
				2	14,558%	14,747%	14,935%	1,3%	
				3	0,278%	0,527%	0,776%	47,2%	
				4	2,623%	2,790%	2,956%	6,0%	
				5	0,000%	0,000%	0,000%	0,0%	
				6	0,000%	0,000%	0,000%	0,0%	
	DCR	1	1	1	0,000%	0,000%	0,000%	0,0%	
				2	4,960%	5,199%	5,439%	4,6%	
				3	0,044%	0,102%	0,160%	56,6%	
				4	8,879%	9,091%	9,302%	2,3%	
				5	0,370%	0,399%	0,428%	7,2%	
				6	2,765%	2,935%	3,105%	5,8%	
		2	2	1	1	5,035%	5,165%	5,295%	2,5%
					2	0,000%	0,000%	0,000%	0,0%
					3	0,000%	0,000%	0,000%	0,0%
					4	26,346%	26,683%	27,019%	1,3%
					5	0,740%	0,757%	0,774%	2,2%
					6	13,958%	14,177%	14,396%	1,5%
3		3	1	1	0,096%	0,157%	0,217%	38,6%	
				2	0,000%	0,000%	0,000%	0,0%	
				3	0,000%	0,000%	0,000%	0,0%	
				4	3,662%	3,707%	3,752%	1,2%	
				5	0,000%	0,000%	0,000%	0,0%	
				6	0,262%	0,327%	0,391%	19,6%	
4		4	1	1	8,852%	9,098%	9,344%	2,7%	
				2	26,256%	26,645%	27,033%	1,5%	
				3	3,161%	3,442%	3,722%	8,2%	
				4	0,000%	0,000%	0,000%	0,0%	
				5	0,152%	0,186%	0,220%	18,2%	
				6	2,274%	2,403%	2,532%	5,4%	
5		5	1	1	0,314%	0,371%	0,428%	15,4%	
				2	0,718%	0,774%	0,830%	7,2%	
				3	0,000%	0,003%	0,011%	277,6%	
				4	0,143%	0,183%	0,224%	22,2%	
				5	0,000%	0,000%	0,000%	0,0%	
				6	0,000%	0,000%	0,000%	0,0%	
6		6	1	1	2,611%	2,907%	3,204%	10,2%	
				2	13,989%	14,232%	14,475%	1,7%	
				3	0,221%	0,264%	0,307%	16,4%	
				4	2,299%	2,423%	2,546%	5,1%	
				5	0,000%	0,000%	0,000%	0,0%	
				6	0,000%	0,000%	0,000%	0,0%	

Tab. G-3 – Intervalos de confiança para a probabilidade de bloqueio ponto a ponto nos métodos de encaminhamento dinâmicos e no DIRECTO (sistema 1).

DIRECTO	1	1	0,000%	0,000%	0,000%	0,0%
		2	15,292%	15,721%	16,150%	2,7%
		3	0,000%	0,000%	0,000%	0,0%
		4	11,770%	12,008%	12,246%	2,0%
		5	0,009%	0,024%	0,039%	61,2%
		6	2,977%	3,319%	3,661%	10,3%
	2	1	15,537%	15,760%	15,983%	1,4%
		2	0,000%	0,000%	0,000%	0,0%
		3	0,000%	0,000%	0,000%	0,0%
		4	33,430%	33,866%	34,301%	1,3%
		5	6,957%	7,039%	7,121%	1,2%
		6	19,892%	20,086%	20,281%	1,0%
	3	1	0,000%	0,003%	0,011%	277,6%
		2	0,000%	0,000%	0,000%	0,0%
		3	0,000%	0,000%	0,000%	0,0%
		4	35,993%	36,937%	37,882%	2,6%
		5	0,000%	0,000%	0,000%	0,0%
		6	4,049%	4,286%	4,523%	5,5%
	4	1	11,754%	12,020%	12,287%	2,2%
		2	33,415%	33,854%	34,292%	1,3%
		3	35,712%	36,710%	37,709%	2,7%
		4	0,000%	0,000%	0,000%	0,0%
		5	0,000%	0,000%	0,000%	0,0%
		6	29,147%	29,530%	29,914%	1,3%
5	1	0,006%	0,015%	0,023%	58,8%	
	2	6,843%	7,040%	7,238%	2,8%	
	3	0,000%	0,000%	0,000%	0,0%	
	4	0,000%	0,000%	0,000%	0,0%	
	5	0,000%	0,000%	0,000%	0,0%	
	6	0,000%	0,000%	0,000%	0,0%	
6	1	3,110%	3,325%	3,540%	6,5%	
	2	19,787%	20,119%	20,451%	1,6%	
	3	3,822%	4,032%	4,242%	5,2%	
	4	29,020%	29,398%	29,777%	1,3%	
	5	0,000%	0,000%	0,000%	0,0%	
	6	0,000%	0,000%	0,000%	0,0%	
RTNR	1	1	0,000%	0,000%	0,000%	0,0%
		2	5,618%	5,912%	6,205%	5,0%
		3	4,253%	4,663%	5,073%	8,8%
		4	6,707%	6,923%	7,139%	3,1%
		5	2,976%	3,143%	3,311%	5,3%
		6	3,260%	3,322%	3,384%	1,9%
	2	1	6,114%	6,217%	6,321%	1,7%
		2	0,000%	0,000%	0,000%	0,0%
		3	0,058%	0,078%	0,098%	25,8%
		4	25,508%	25,938%	26,368%	1,7%
		5	2,202%	2,305%	2,409%	4,5%
		6	8,451%	8,637%	8,824%	2,2%
	3	1	2,522%	2,844%	3,166%	11,3%
		2	0,022%	0,055%	0,089%	60,8%
		3	0,000%	0,000%	0,000%	0,0%
		4	4,616%	4,807%	4,998%	4,0%
		5	0,873%	0,977%	1,080%	10,6%
		6	1,393%	1,508%	1,622%	7,6%
	4	1	6,803%	7,015%	7,228%	3,0%
		2	24,119%	24,590%	25,061%	1,9%
		3	11,932%	12,337%	12,741%	3,3%
		4	0,000%	0,000%	0,000%	0,0%
		5	1,154%	1,250%	1,345%	7,6%
		6	4,450%	4,709%	4,967%	5,5%
5	1	1,743%	1,951%	2,158%	10,7%	
	2	1,825%	1,982%	2,140%	7,9%	
	3	0,882%	1,089%	1,295%	19,0%	
	4	1,051%	1,149%	1,246%	8,5%	
	5	0,000%	0,000%	0,000%	0,0%	
	6	0,000%	0,000%	0,000%	0,0%	
6	1	2,537%	2,641%	2,745%	3,9%	
	2	4,351%	4,573%	4,796%	4,9%	
	3	1,316%	1,549%	1,781%	15,0%	
	4	1,861%	2,011%	2,161%	7,5%	
	5	0,000%	0,000%	0,000%	0,0%	
	6	0,000%	0,000%	0,000%	0,0%	

Tab. G-4 – Intervalos de confiança para a probabilidade de bloqueio ponto a ponto nos métodos de encaminhamento dinâmicos e no DIRECTO (sistema 1) – conclusão.

Note-se que na tabela seguinte apenas são apresentados os intervalos de confiança para o bloqueio ponto-a-ponto da origem 2 ao destino 4. Uma análise aos resultados da simulação mostrou que em geral o máximo do bloqueio ponto-a-ponto ocorria entre as centrais 2 e 4 (com uma diferença bastante significativa em relação aos outros pares de centrais, com excepção do método directo para cargas reduzidas). Por esta razão não pareceu necessário apresentar os intervalos de confiança para os restantes casos.

item	Ppbl{0}
reserva	0

alfa 5%

Average of valor				Limite Inferior	Média	Limite Superior	Conf (Rel)
carga	metodo	origem	destino				
1	DAR	2	4	1,044%	1,128%	1,213%	7,5%
	DCR	2	4	0,485%	0,521%	0,557%	6,9%
	DIRECTO	2	4	10,996%	11,352%	11,707%	3,1%
	RTNR	2	4	0,012%	0,024%	0,036%	49,5%
1,1	DAR	2	4	5,367%	5,525%	5,683%	2,9%
	DCR	2	4	3,848%	4,073%	4,299%	5,5%
	DIRECTO	2	4	17,536%	17,902%	18,267%	2,0%
	RTNR	2	4	0,469%	0,529%	0,588%	11,2%
1,2	DAR	2	4	12,390%	12,711%	13,032%	2,5%
	DCR	2	4	11,052%	11,288%	11,524%	2,1%
	DIRECTO	2	4	23,379%	23,688%	23,997%	1,3%
	RTNR	2	4	2,301%	2,531%	2,760%	9,1%
1,3	DAR	2	4	20,499%	20,795%	21,090%	1,4%
	DCR	2	4	19,089%	19,456%	19,823%	1,9%
	DIRECTO	2	4	29,023%	29,246%	29,470%	0,8%
	RTNR	2	4	13,984%	14,481%	14,978%	3,4%
1,4	DAR	2	4	27,389%	27,776%	28,162%	1,4%
	DCR	2	4	26,346%	26,683%	27,019%	1,3%
	DIRECTO	2	4	33,430%	33,866%	34,301%	1,3%
	RTNR	2	4	25,508%	25,938%	26,368%	1,7%
1,5	DAR	2	4	33,504%	33,711%	33,918%	0,6%
	DCR	2	4	32,784%	32,865%	32,945%	0,2%
	DIRECTO	2	4	37,753%	37,984%	38,215%	0,6%
	RTNR	2	4	33,317%	33,493%	33,669%	0,5%
1,6	DAR	2	4	38,644%	38,846%	39,049%	0,5%
	DCR	2	4	38,177%	38,363%	38,549%	0,5%
	DIRECTO	2	4	41,668%	41,860%	42,051%	0,5%
	RTNR	2	4	39,456%	39,667%	39,879%	0,5%
1,7	DAR	2	4	42,920%	43,045%	43,170%	0,3%
	DCR	2	4	42,513%	42,697%	42,882%	0,4%
	DIRECTO	2	4	44,868%	45,092%	45,316%	0,5%
	RTNR	2	4	43,976%	44,147%	44,317%	0,4%
1,8	DAR	2	4	46,397%	46,681%	46,965%	0,6%
	DCR	2	4	46,152%	46,443%	46,733%	0,6%
	DIRECTO	2	4	47,882%	48,117%	48,351%	0,5%
	RTNR	2	4	47,480%	47,745%	48,011%	0,6%
1,9	DAR	2	4	49,666%	49,790%	49,913%	0,2%
	DCR	2	4	49,523%	49,651%	49,778%	0,3%
	DIRECTO	2	4	50,667%	50,805%	50,943%	0,3%
	RTNR	2	4	50,533%	50,665%	50,797%	0,3%
2	DAR	2	4	52,328%	52,501%	52,675%	0,3%
	DCR	2	4	52,215%	52,425%	52,635%	0,4%
	DIRECTO	2	4	53,064%	53,234%	53,404%	0,3%
	RTNR	2	4	53,010%	53,184%	53,357%	0,3%

Tab. G-5 – Intervalos de confiança para a probabilidade de bloqueio ponto a ponto máxima nos métodos de encaminhamento dinâmicos e no DIRECTO, para cada um dos vários factores de sobrecarga (sistema 1).

item	Rpbl
reserva	0
carga	1,4

alfa 5%

Average of valor			Limite Inferior	Média	Limite Superior	Conf (Rel)		
metodo	origem	destino						
DAR	1	2	38,242%	38,648%	39,054%	1,1%		
			7,305%	7,557%	7,808%	3,3%		
		2	3	0,000%	0,006%	0,013%	120,1%	
				37,136%	37,550%	37,964%	1,1%	
			4	21,273%	21,400%	21,527%	0,6%	
				23,739%	23,990%	24,242%	1,1%	
	5			69,148%	69,446%	69,743%	0,4%	
				1,185%	1,216%	1,248%	2,6%	
	3	4	23,649%	23,818%	23,986%	0,7%		
			1,152%	1,291%	1,429%	10,7%		
		5	37,381%	37,733%	38,084%	0,9%		
			0,000%	0,000%	0,000%	0,0%		
			4	1	49,039%	49,413%	49,786%	0,8%
					5,287%	5,424%	5,561%	2,5%
	2	18,449%		18,744%	19,038%	1,6%		
		23,684%		24,071%	24,457%	1,6%		
		3		12,769%	13,263%	13,757%	3,7%	
				0,000%	0,007%	0,017%	130,8%	
	5	4	34,240%	34,595%	34,951%	1,0%		
			21,323%	21,414%	21,505%	0,4%		
		6	21,557%	21,794%	22,030%	1,1%		
			41,090%	41,585%	42,080%	1,2%		
			5	1,179%	1,300%	1,422%	9,3%	
				19,246%	19,741%	20,236%	2,5%	
6	4	1,989%	2,166%	2,344%	8,2%			
		31,612%	31,856%	32,100%	0,8%			
	7	1	0,000%	0,000%	0,000%	0,0%		
			21,057%	21,247%	21,437%	0,9%		
		2	7,184%	7,343%	7,502%	2,2%		
			11,830%	12,069%	12,308%	2,0%		
3			10,650%	10,768%	10,885%	1,1%		
			9,277%	9,394%	9,511%	1,3%		
3	4	0,028%	0,044%	0,060%	36,5%			
		33,476%	33,887%	34,298%	1,2%			
	5	14,337%	14,493%	14,649%	1,1%			
		20,211%	20,407%	20,603%	1,0%			
		6	45,644%	46,210%	46,777%	1,2%		
			1,633%	1,762%	1,890%	7,3%		
4	5	22,479%	22,658%	22,837%	0,8%			
		0,767%	0,834%	0,902%	8,1%			
	6	29,661%	29,998%	30,335%	1,1%			
		0,000%	0,000%	0,000%	0,0%			

Tab. G-6 – Intervalos de confiança para a probabilidade de bloqueio nos feixes nos métodos de encaminhamento dinâmicos (sistema 1).

item	Tpbl
carga	1,2

alfa

5%

Average of valor		Limite Inferior	Média	Limite Superior	Conf (Rel)
metodo	idcorrida				
DAR	duração=26h	1,894%	1,966%	2,038%	3,7%
	duração=30h	1,951%	1,995%	2,039%	2,2%
	duração=34h	1,970%	1,987%	2,005%	0,9%
	duração=38h	1,959%	1,996%	2,033%	1,9%
	duração=42h	1,935%	1,985%	2,034%	2,5%
	duração=46h	1,917%	1,976%	2,035%	3,0%
	duração=48h	1,918%	1,966%	2,014%	2,4%
DCR	duração=26h	1,696%	1,754%	1,812%	3,3%
	duração=30h	1,764%	1,789%	1,813%	1,4%
	duração=34h	1,755%	1,782%	1,809%	1,5%
	duração=38h	1,745%	1,790%	1,836%	2,5%
	duração=42h	1,727%	1,780%	1,834%	3,0%
	duração=46h	1,714%	1,772%	1,829%	3,2%
	duração=48h	1,716%	1,762%	1,808%	2,6%
DIRECTO	duração=26h	6,038%	6,242%	6,445%	3,3%
	duração=30h	6,223%	6,305%	6,386%	1,3%
	duração=34h	6,253%	6,329%	6,405%	1,2%
	duração=38h	6,283%	6,328%	6,373%	0,7%
	duração=42h	6,250%	6,320%	6,390%	1,1%
	duração=46h	6,226%	6,301%	6,375%	1,2%
	duração=48h	6,226%	6,295%	6,364%	1,1%
RTNR	duração=26h	0,358%	0,415%	0,473%	13,9%
	duração=30h	0,393%	0,432%	0,471%	9,0%
	duração=34h	0,407%	0,433%	0,458%	5,9%
	duração=38h	0,410%	0,441%	0,471%	6,9%
	duração=42h	0,403%	0,438%	0,472%	7,9%
	duração=46h	0,395%	0,436%	0,476%	9,2%
	duração=48h	0,398%	0,432%	0,467%	8,0%

Tab. G-7 – Intervalos de confiança para a probabilidade de bloqueio global nos métodos de encaminhamento dinâmicos versus tempo de operação para um factor de sobrecarga de 1,2 (sistema 1).

item
carga

alfa

5%

Average of valor	Limite Inferior	Média	Limite Superior	Conf (Rel)
metodo				
DAR	1,970%	1,982%	1,993%	0,6%
DCR	1,763%	1,776%	1,788%	0,7%
DIRECTO	6,275%	6,303%	6,331%	0,4%
RTNR	0,425%	0,432%	0,440%	1,7%

Tab. G-8 – Intervalos de confiança para a probabilidade de bloqueio global nos métodos de encaminhamento dinâmicos usando observações em diferentes tempos de operação para um factor de sobrecarga de 1,2 (sistema 1).

## G.2 Intervalos de confiança da experiência com o Sistema 2

item	Tpbl
reserva	0
idcorrida	mult

alfa

5%

Average of valor		Limite Inferior	Média	Limite Superior	Conf (Rel)
carga	metodo				
1	DAR	0,000%	0,000%	0,001%	54,8%
	DIRECTO	0,413%	0,420%	0,427%	1,6%
	RTNR	0,000%	0,000%	0,000%	0,0%
1,1	DAR	0,002%	0,003%	0,004%	24,0%
	DIRECTO	0,710%	0,728%	0,746%	2,4%
	RTNR	0,000%	0,000%	0,000%	170,0%
1,2	DAR	0,035%	0,038%	0,042%	9,0%
	DIRECTO	1,364%	1,407%	1,451%	3,1%
	RTNR	0,001%	0,003%	0,004%	55,7%
1,3	DAR	0,312%	0,339%	0,366%	8,0%
	DIRECTO	2,522%	2,575%	2,629%	2,1%
	RTNR	0,094%	0,122%	0,151%	23,4%
1,4	DAR	1,636%	1,672%	1,709%	2,2%
	DIRECTO	4,106%	4,156%	4,205%	1,2%
	RTNR	1,626%	1,683%	1,739%	3,3%
1,5	DAR	4,881%	4,925%	4,968%	0,9%
	DIRECTO	6,165%	6,193%	6,222%	0,5%
	RTNR	5,183%	5,252%	5,320%	1,3%
1,6	DAR	9,017%	9,058%	9,098%	0,4%
	DIRECTO	8,789%	8,825%	8,861%	0,4%
	RTNR	8,947%	8,971%	8,996%	0,3%
1,7	DAR	13,032%	13,081%	13,130%	0,4%
	DIRECTO	11,696%	11,744%	11,792%	0,4%
	RTNR	12,160%	12,222%	12,284%	0,5%
1,8	DAR	16,682%	16,734%	16,787%	0,3%
	DIRECTO	14,640%	14,695%	14,750%	0,4%
	RTNR	15,269%	15,304%	15,340%	0,2%
1,9	DAR	20,027%	20,101%	20,175%	0,4%
	DIRECTO	17,466%	17,532%	17,597%	0,4%
	RTNR	18,255%	18,309%	18,364%	0,3%
2	DAR	23,219%	23,294%	23,369%	0,3%
	DIRECTO	20,178%	20,276%	20,375%	0,5%
	RTNR	21,093%	21,186%	21,279%	0,4%

Tab. G-9 – Intervalos de confiança para a congestão de chamadas na rede (sistema 2).

item	Tpbl
idcorrida	mult

alfa

5%

Average of valor			Limite Inferior	Média	Limite Superior	Conf (Rel)
carga	metodo	reserva				
1	DAR	0	0,000%	0,000%	0,001%	54,8%
		1	0,000%	0,000%	0,001%	54,8%
	DIRECTO	0	0,413%	0,420%	0,427%	1,6%
1,1	DAR	0	0,002%	0,003%	0,004%	24,0%
		1	0,002%	0,003%	0,004%	23,5%
	DIRECTO	0	0,710%	0,728%	0,746%	2,4%
1,2	DAR	0	0,035%	0,038%	0,042%	9,0%
		1	0,034%	0,039%	0,044%	13,4%
	DIRECTO	0	1,364%	1,407%	1,451%	3,1%
1,3	DAR	0	0,312%	0,339%	0,366%	8,0%
		1	0,382%	0,416%	0,449%	8,1%
	DIRECTO	0	2,522%	2,575%	2,629%	2,1%
1,4	DAR	0	1,636%	1,672%	1,709%	2,2%
		1	2,329%	2,373%	2,417%	1,8%
	DIRECTO	0	4,106%	4,156%	4,205%	1,2%
1,5	DAR	0	4,881%	4,925%	4,968%	0,9%
		1	5,496%	5,535%	5,574%	0,7%
	DIRECTO	0	6,165%	6,193%	6,222%	0,5%
1,6	DAR	0	9,017%	9,058%	9,098%	0,4%
		1	8,961%	9,009%	9,058%	0,5%
	DIRECTO	0	8,789%	8,825%	8,861%	0,4%
1,7	DAR	0	13,032%	13,081%	13,130%	0,4%
		1	12,353%	12,419%	12,484%	0,5%
	DIRECTO	0	11,696%	11,744%	11,792%	0,4%
1,8	DAR	0	16,682%	16,734%	16,787%	0,3%
		1	15,638%	15,678%	15,718%	0,3%
	DIRECTO	0	14,640%	14,695%	14,750%	0,4%
1,9	DAR	0	20,027%	20,101%	20,175%	0,4%
		1	18,677%	18,726%	18,774%	0,3%
	DIRECTO	0	17,466%	17,532%	17,597%	0,4%
2	DAR	0	23,219%	23,294%	23,369%	0,3%
		1	21,537%	21,620%	21,702%	0,4%
	DIRECTO	0	20,178%	20,276%	20,375%	0,5%

Tab. G-10 – Intervalos de confiança para a congestão de chamadas na rede, para dois tipos de reserva no DAR (sistema 2).

item	Tpbl
reserva	0

alfa

5%

Average of valor			Limite Inferior	Média	Limite Superior	Conf (Rel)
carga	metodo	idcorrida				
1	DAR	mult	0,000%	0,000%	0,001%	54,8%
		mult_dar_res	0,001%	0,001%	0,001%	41,1%
	DIRECTO	mult	0,413%	0,420%	0,427%	1,6%
1,1	DAR	mult	0,002%	0,003%	0,004%	24,0%
		mult_dar_res	0,008%	0,009%	0,010%	10,2%
	DIRECTO	mult	0,710%	0,728%	0,746%	2,4%
1,2	DAR	mult	0,035%	0,038%	0,042%	9,0%
		mult_dar_res	0,089%	0,096%	0,104%	7,8%
	DIRECTO	mult	1,364%	1,407%	1,451%	3,1%
1,3	DAR	mult	0,312%	0,339%	0,366%	8,0%
		mult_dar_res	0,543%	0,577%	0,611%	5,9%
	DIRECTO	mult	2,522%	2,575%	2,629%	2,1%
1,4	DAR	mult	1,636%	1,672%	1,709%	2,2%
		mult_dar_res	1,760%	1,801%	1,842%	2,3%
	DIRECTO	mult	4,106%	4,156%	4,205%	1,2%
1,5	DAR	mult	4,881%	4,925%	4,968%	0,9%
		mult_dar_res	3,984%	4,019%	4,055%	0,9%
	DIRECTO	mult	6,165%	6,193%	6,222%	0,5%
1,6	DAR	mult	9,017%	9,058%	9,098%	0,4%
		mult_dar_res	7,010%	7,060%	7,110%	0,7%
	DIRECTO	mult	8,789%	8,825%	8,861%	0,4%
1,7	DAR	mult	13,032%	13,081%	13,130%	0,4%
		mult_dar_res	10,305%	10,350%	10,395%	0,4%
	DIRECTO	mult	11,696%	11,744%	11,792%	0,4%
1,8	DAR	mult	16,682%	16,734%	16,787%	0,3%
		mult_dar_res	13,555%	13,602%	13,650%	0,4%
	DIRECTO	mult	14,640%	14,695%	14,750%	0,4%
1,9	DAR	mult	20,027%	20,101%	20,175%	0,4%
		mult_dar_res	16,593%	16,639%	16,684%	0,3%
	DIRECTO	mult	17,466%	17,532%	17,597%	0,4%
2	DAR	mult	23,219%	23,294%	23,369%	0,3%
		mult_dar_res	19,418%	19,512%	19,605%	0,5%
	DIRECTO	mult	20,178%	20,276%	20,375%	0,5%

Tab. G-11 – Intervalos de confiança para a congestão de chamadas na rede, com reserva de circuitos fixa diferente de 0 (sistema 2).

item	Tpbl			alfa	5%		
Average of valor				Limite Inferior	Média	Limite Superior	Conf (Rel)
carga	idcorrida	metodo	reserva				
1	mult	DAR	0	0,000%	0,000%	0,001%	54,8%
			1	0,000%	0,000%	0,001%	54,8%
		RTNR	0	0,000%	0,000%	0,000%	0,0%
	mult_dar_res	DAR	0	0,001%	0,001%	0,001%	41,1%
			1	0,007%	0,007%	0,025%	277,6%
		RTNR	0	0,000%	0,000%	0,000%	170,0%
1,1	mult	DAR	0	0,002%	0,003%	0,004%	24,0%
			1	0,002%	0,003%	0,004%	23,5%
		RTNR	0	0,000%	0,000%	0,000%	170,0%
	mult_dar_res	DAR	0	0,000%	0,000%	0,000%	170,0%
			1	0,008%	0,009%	0,010%	10,2%
		RTNR	0	0,008%	0,009%	0,010%	10,2%
1,2	mult	DAR	0	0,035%	0,038%	0,042%	9,0%
			1	0,034%	0,039%	0,044%	13,4%
		RTNR	0	0,001%	0,003%	0,004%	55,7%
	mult_dar_res	DAR	0	0,002%	0,003%	0,004%	45,4%
			1	0,089%	0,096%	0,104%	7,8%
		RTNR	0	0,089%	0,096%	0,104%	7,8%
1,3	mult	DAR	0	0,312%	0,339%	0,366%	8,0%
			1	0,382%	0,416%	0,449%	8,1%
		RTNR	0	0,094%	0,122%	0,151%	23,4%
	mult_dar_res	DAR	0	0,098%	0,120%	0,142%	18,5%
			1	0,543%	0,577%	0,611%	5,9%
		RTNR	0	0,543%	0,577%	0,611%	5,9%
1,4	mult	DAR	0	1,636%	1,672%	1,709%	2,2%
			1	2,329%	2,373%	2,417%	1,8%
		RTNR	0	1,626%	1,683%	1,739%	3,3%
	mult_dar_res	DAR	0	1,599%	1,654%	1,710%	3,4%
			1	1,760%	1,801%	1,842%	2,3%
		RTNR	0	1,760%	1,801%	1,842%	2,3%
1,5	mult	DAR	0	4,881%	4,925%	4,968%	0,9%
			1	5,496%	5,535%	5,574%	0,7%
		RTNR	0	5,183%	5,252%	5,320%	1,3%
	mult_dar_res	DAR	0	5,152%	5,202%	5,252%	1,0%
			1	3,984%	4,019%	4,055%	0,9%
		RTNR	0	3,984%	4,019%	4,055%	0,9%
1,6	mult	DAR	0	9,017%	9,058%	9,098%	0,4%
			1	8,961%	9,009%	9,058%	0,5%
		RTNR	0	8,947%	8,971%	8,996%	0,3%
	mult_dar_res	DAR	0	8,821%	8,868%	8,915%	0,5%
			1	7,010%	7,060%	7,110%	0,7%
		RTNR	0	7,010%	7,060%	7,110%	0,7%
1,7	mult	DAR	0	13,032%	13,081%	13,130%	0,4%
			1	12,353%	12,419%	12,484%	0,5%
		RTNR	0	12,160%	12,222%	12,284%	0,5%
	mult_dar_res	DAR	0	12,082%	12,150%	12,219%	0,6%
			1	10,305%	10,350%	10,395%	0,4%
		RTNR	0	10,305%	10,350%	10,395%	0,4%
1,8	mult	DAR	0	16,682%	16,734%	16,787%	0,3%
			1	15,638%	15,678%	15,718%	0,3%
		RTNR	0	15,269%	15,304%	15,340%	0,2%
	mult_dar_res	DAR	0	15,245%	15,298%	15,351%	0,3%
			1	13,555%	13,602%	13,650%	0,4%
		RTNR	0	13,555%	13,602%	13,650%	0,4%
1,9	mult	DAR	0	20,027%	20,101%	20,175%	0,4%
			1	18,677%	18,726%	18,774%	0,3%
		RTNR	0	18,255%	18,309%	18,364%	0,3%
	mult_dar_res	DAR	0	18,228%	18,295%	18,363%	0,4%
			1	16,593%	16,639%	16,684%	0,3%
		RTNR	0	16,593%	16,639%	16,684%	0,3%
2	mult	DAR	0	23,219%	23,294%	23,369%	0,3%
			1	21,537%	21,620%	21,702%	0,4%
		RTNR	0	21,093%	21,186%	21,279%	0,4%
	mult_dar_res	DAR	0	21,098%	21,183%	21,268%	0,4%
			1	19,418%	19,512%	19,605%	0,5%
		RTNR	0	19,418%	19,512%	19,605%	0,5%

Tab. G-12 – Intervalos de confiança para a probabilidade de bloqueio global de chamadas na rede, para os métodos DAR e RTNR (sistema 2).

item	TpblCirc
reserva	0
idcorrida	mult

alfa

5%

Average of valor		Limite Inferior	Média	Limite Superior	Conf (Rel)
carga	metodo				
1	DAR	0,000%	0,001%	0,001%	55,4%
	DIRECTO	0,458%	0,466%	0,474%	1,7%
	RTNR	0,000%	0,000%	0,000%	0,0%
1,1	DAR	0,003%	0,004%	0,005%	20,7%
	DIRECTO	0,785%	0,805%	0,825%	2,5%
	RTNR	0,000%	0,000%	0,000%	185,1%
1,2	DAR	0,043%	0,047%	0,051%	9,1%
	DIRECTO	1,502%	1,551%	1,600%	3,1%
	RTNR	0,001%	0,003%	0,004%	53,1%
1,3	DAR	0,380%	0,414%	0,447%	8,1%
	DIRECTO	2,772%	2,829%	2,887%	2,0%
	RTNR	0,103%	0,130%	0,158%	21,1%
1,4	DAR	1,950%	1,994%	2,038%	2,2%
	DIRECTO	4,498%	4,558%	4,617%	1,3%
	RTNR	1,631%	1,684%	1,738%	3,2%
1,5	DAR	5,707%	5,760%	5,813%	0,9%
	DIRECTO	6,737%	6,772%	6,806%	0,5%
	RTNR	4,840%	4,893%	4,946%	1,1%
1,6	DAR	10,402%	10,447%	10,493%	0,4%
	DIRECTO	9,590%	9,630%	9,670%	0,4%
	RTNR	8,412%	8,447%	8,482%	0,4%
1,7	DAR	14,867%	14,917%	14,966%	0,3%
	DIRECTO	12,740%	12,786%	12,832%	0,4%
	RTNR	11,831%	11,886%	11,940%	0,5%
1,8	DAR	18,857%	18,916%	18,974%	0,3%
	DIRECTO	15,905%	15,960%	16,014%	0,3%
	RTNR	15,198%	15,234%	15,270%	0,2%
1,9	DAR	22,485%	22,553%	22,620%	0,3%
	DIRECTO	18,936%	19,001%	19,065%	0,3%
	RTNR	18,392%	18,450%	18,508%	0,3%
2	DAR	25,886%	25,967%	26,047%	0,3%
	DIRECTO	21,825%	21,925%	22,024%	0,5%
	RTNR	21,393%	21,495%	21,598%	0,5%

Tab. G-13 – Intervalos de confiança para a probabilidade de bloqueio global de circuitos na rede (sistema 2).

item	TpblCirc
fluxos	2
reserva	0

alfa

5%

Average of valor			Limite Inferior	Média	Limite Superior	Conf (Rel)
carga	metodo	idcorrida				
1	DAR	mult_dar_res	0,001%	0,001%	0,001%	48,5%
	DIRECTO	mult	0,458%	0,466%	0,474%	1,7%
	RTNR	mult	0,000%	0,000%	0,000%	0,0%
1,1	DAR	mult_dar_res	0,010%	0,011%	0,012%	11,5%
	DIRECTO	mult	0,785%	0,805%	0,825%	2,5%
	RTNR	mult	0,000%	0,000%	0,000%	185,1%
1,2	DAR	mult_dar_res	0,110%	0,120%	0,129%	7,9%
	DIRECTO	mult	1,502%	1,551%	1,600%	3,1%
	RTNR	mult	0,001%	0,003%	0,004%	53,1%
1,3	DAR	mult_dar_res	0,663%	0,703%	0,743%	5,7%
	DIRECTO	mult	2,772%	2,829%	2,887%	2,0%
	RTNR	mult	0,103%	0,130%	0,158%	21,1%
1,4	DAR	mult_dar_res	2,093%	2,141%	2,188%	2,2%
	DIRECTO	mult	4,498%	4,558%	4,617%	1,3%
	RTNR	mult	1,631%	1,684%	1,738%	3,2%
1,5	DAR	mult_dar_res	4,600%	4,640%	4,679%	0,9%
	DIRECTO	mult	6,737%	6,772%	6,806%	0,5%
	RTNR	mult	4,840%	4,893%	4,946%	1,1%
1,6	DAR	mult_dar_res	7,944%	8,000%	8,055%	0,7%
	DIRECTO	mult	9,590%	9,630%	9,670%	0,4%
	RTNR	mult	8,412%	8,447%	8,482%	0,4%
1,7	DAR	mult_dar_res	11,545%	11,585%	11,625%	0,3%
	DIRECTO	mult	12,740%	12,786%	12,832%	0,4%
	RTNR	mult	11,831%	11,886%	11,940%	0,5%
1,8	DAR	mult_dar_res	15,039%	15,085%	15,130%	0,3%
	DIRECTO	mult	15,905%	15,960%	16,014%	0,3%
	RTNR	mult	15,198%	15,234%	15,270%	0,2%
1,9	DAR	mult_dar_res	18,281%	18,328%	18,375%	0,3%
	DIRECTO	mult	18,936%	19,001%	19,065%	0,3%
	RTNR	mult	18,392%	18,450%	18,508%	0,3%
2	DAR	mult_dar_res	21,277%	21,371%	21,465%	0,4%
	DIRECTO	mult	21,825%	21,925%	22,024%	0,5%
	RTNR	mult	21,393%	21,495%	21,598%	0,5%

Tab. G-14 – Intervalos de confiança para a probabilidade de bloqueio global de circuitos na rede, com reserva de circuitos fixa diferente de 0 para o DAR (sistema 2).

metodo	RTNR
item	Tpbl

alfa 5%

Average of valor		Limite Inferior	Média	Limite Superior	Conf (Rel)
carga	idcorrida				
1	mult_rtnr_rk=0.05	0,000%	0,000%	0,000%	0,0%
	mult_rtnr_rk=0.55	0,000%	0,000%	0,001%	277,6%
	mult_rtnr_rk=1.05	0,000%	0,000%	0,001%	277,6%
	mult_rtnr_rk=1.55	0,000%	0,000%	0,000%	277,6%
	mult_rtnr_rk=2.00	0,000%	0,000%	0,000%	277,6%
1,1	mult_rtnr_rk=0.05	0,000%	0,000%	0,000%	277,6%
	mult_rtnr_rk=0.55	0,000%	0,000%	0,000%	185,1%
	mult_rtnr_rk=1.05	0,000%	0,000%	0,000%	277,6%
	mult_rtnr_rk=1.55	0,000%	0,006%	0,022%	277,0%
	mult_rtnr_rk=2.00	0,000%	0,000%	0,000%	277,6%
1,2	mult_rtnr_rk=0.05	0,000%	0,014%	0,044%	203,9%
	mult_rtnr_rk=0.55	0,000%	0,004%	0,009%	109,8%
	mult_rtnr_rk=1.05	0,002%	0,003%	0,004%	34,8%
	mult_rtnr_rk=1.55	0,002%	0,003%	0,004%	35,2%
	mult_rtnr_rk=2.00	0,002%	0,003%	0,004%	35,2%
1,3	mult_rtnr_rk=0.05	0,093%	0,116%	0,139%	19,6%
	mult_rtnr_rk=0.55	0,106%	0,129%	0,153%	18,3%
	mult_rtnr_rk=1.05	0,101%	0,125%	0,150%	19,8%
	mult_rtnr_rk=1.55	0,101%	0,125%	0,150%	19,8%
	mult_rtnr_rk=2.00	0,101%	0,125%	0,150%	19,8%
1,4	mult_rtnr_rk=0.05	1,633%	1,685%	1,737%	3,1%
	mult_rtnr_rk=0.55	1,718%	1,770%	1,822%	2,9%
	mult_rtnr_rk=1.05	1,719%	1,781%	1,844%	3,5%
	mult_rtnr_rk=1.55	1,719%	1,781%	1,843%	3,5%
	mult_rtnr_rk=2.00	1,719%	1,781%	1,843%	3,5%
1,5	mult_rtnr_rk=0.05	5,217%	5,265%	5,312%	0,9%
	mult_rtnr_rk=0.55	5,110%	5,176%	5,242%	1,3%
	mult_rtnr_rk=1.05	5,129%	5,188%	5,246%	1,1%
	mult_rtnr_rk=1.55	5,134%	5,182%	5,229%	0,9%
	mult_rtnr_rk=2.00	5,129%	5,189%	5,250%	1,2%
1,6	mult_rtnr_rk=0.05	8,947%	8,971%	8,996%	0,3%
	mult_rtnr_rk=0.55	8,477%	8,517%	8,557%	0,5%
	mult_rtnr_rk=1.05	8,453%	8,511%	8,568%	0,7%
	mult_rtnr_rk=1.55	8,453%	8,511%	8,568%	0,7%
	mult_rtnr_rk=2.00	8,453%	8,511%	8,568%	0,7%
1,7	mult_rtnr_rk=0.05	12,160%	12,222%	12,284%	0,5%
	mult_rtnr_rk=0.55	11,639%	11,672%	11,705%	0,3%
	mult_rtnr_rk=1.05	11,636%	11,669%	11,702%	0,3%
	mult_rtnr_rk=1.55	11,636%	11,669%	11,702%	0,3%
	mult_rtnr_rk=2.00	11,615%	11,663%	11,710%	0,4%
1,8	mult_rtnr_rk=0.05	15,269%	15,304%	15,340%	0,2%
	mult_rtnr_rk=0.55	14,846%	14,888%	14,931%	0,3%
	mult_rtnr_rk=1.05	14,847%	14,891%	14,934%	0,3%
	mult_rtnr_rk=1.55	14,847%	14,891%	14,934%	0,3%
	mult_rtnr_rk=2.00	14,847%	14,891%	14,934%	0,3%
1,9	mult_rtnr_rk=0.05	18,246%	18,306%	18,367%	0,3%
	mult_rtnr_rk=0.55	17,926%	17,981%	18,036%	0,3%
	mult_rtnr_rk=1.05	17,912%	17,973%	18,035%	0,3%
	mult_rtnr_rk=1.55	17,912%	17,973%	18,035%	0,3%
	mult_rtnr_rk=2.00	17,911%	17,974%	18,038%	0,4%
2	mult_rtnr_rk=0.05	21,092%	21,189%	21,287%	0,5%
	mult_rtnr_rk=0.55	20,810%	20,900%	20,989%	0,4%
	mult_rtnr_rk=1.05	20,816%	20,902%	20,988%	0,4%
	mult_rtnr_rk=1.55	20,816%	20,902%	20,988%	0,4%
	mult_rtnr_rk=2.00	20,816%	20,902%	20,988%	0,4%

Tab. G-15 – Intervalos de confiança para a probabilidade de bloqueio global no RTNR versus o factor do  $R_k$  (sistema 2).

metodo		RTNR	alfa	5%	
item		Tpbl			
fluxos		2			
Average of valor		Limite Inferior	Média	Limite Superior	Conf (Rel)
carga	idcorrida				
1	mult_rtnr_vtrafk=0.30	0,000%	0,000%	0,000%	0,0%
	mult_rtnr_vtrafk=0.70	0,000%	0,003%	0,011%	277,6%
	mult_rtnr_vtrafk=1.10	0,000%	0,006%	0,024%	277,6%
	mult_rtnr_vtrafk=1.50	0,000%	0,000%	0,000%	170,0%
	mult_rtnr_vtrafk=1.90	0,000%	0,003%	0,013%	277,6%
	mult_rtnr_vtrafk=2.30	0,000%	0,000%	0,000%	0,0%
	mult_rtnr_vtrafk=2.70	0,000%	0,000%	0,000%	0,0%
	mult_rtnr_vtrafk=3.10	0,000%	0,003%	0,011%	277,6%
1,1	mult_rtnr_vtrafk=0.30	0,000%	0,000%	0,001%	260,6%
	mult_rtnr_vtrafk=0.70	0,000%	0,000%	0,000%	185,1%
	mult_rtnr_vtrafk=1.10	0,000%	0,014%	0,055%	277,4%
	mult_rtnr_vtrafk=1.50	0,000%	0,000%	0,001%	277,6%
	mult_rtnr_vtrafk=1.90	0,000%	0,000%	0,000%	277,6%
	mult_rtnr_vtrafk=2.30	0,000%	0,000%	0,000%	0,0%
	mult_rtnr_vtrafk=2.70	0,000%	0,016%	0,060%	277,4%
	mult_rtnr_vtrafk=3.10	0,000%	0,000%	0,000%	129,9%
1,2	mult_rtnr_vtrafk=0.30	0,002%	0,003%	0,005%	38,0%
	mult_rtnr_vtrafk=0.70	0,001%	0,003%	0,004%	57,7%
	mult_rtnr_vtrafk=1.10	0,002%	0,003%	0,004%	41,9%
	mult_rtnr_vtrafk=1.50	0,002%	0,003%	0,005%	35,4%
	mult_rtnr_vtrafk=1.90	0,002%	0,008%	0,014%	76,3%
	mult_rtnr_vtrafk=2.30	0,003%	0,006%	0,010%	58,3%
	mult_rtnr_vtrafk=2.70	0,004%	0,006%	0,008%	29,4%
	mult_rtnr_vtrafk=3.10	0,006%	0,008%	0,010%	24,4%
1,3	mult_rtnr_vtrafk=0.30	0,137%	0,164%	0,191%	16,4%
	mult_rtnr_vtrafk=0.70	0,103%	0,125%	0,148%	17,8%
	mult_rtnr_vtrafk=1.10	0,093%	0,116%	0,139%	19,6%
	mult_rtnr_vtrafk=1.50	0,103%	0,122%	0,140%	15,3%
	mult_rtnr_vtrafk=1.90	0,105%	0,132%	0,158%	19,9%
	mult_rtnr_vtrafk=2.30	0,123%	0,143%	0,163%	14,1%
	mult_rtnr_vtrafk=2.70	0,136%	0,158%	0,180%	14,0%
	mult_rtnr_vtrafk=3.10	0,155%	0,177%	0,199%	12,4%
1,4	mult_rtnr_vtrafk=0.30	2,491%	2,536%	2,580%	1,8%
	mult_rtnr_vtrafk=0.70	1,886%	1,925%	1,964%	2,0%
	mult_rtnr_vtrafk=1.10	1,633%	1,685%	1,737%	3,1%
	mult_rtnr_vtrafk=1.50	1,570%	1,615%	1,660%	2,8%
	mult_rtnr_vtrafk=1.90	1,546%	1,598%	1,651%	3,3%
	mult_rtnr_vtrafk=2.30	1,561%	1,607%	1,654%	2,9%
	mult_rtnr_vtrafk=2.70	1,585%	1,642%	1,699%	3,5%
	mult_rtnr_vtrafk=3.10	1,619%	1,673%	1,727%	3,2%
1,5	mult_rtnr_vtrafk=0.30	7,566%	7,643%	7,720%	1,0%
	mult_rtnr_vtrafk=0.70	5,951%	6,007%	6,063%	0,9%
	mult_rtnr_vtrafk=1.10	5,217%	5,265%	5,312%	0,9%
	mult_rtnr_vtrafk=1.50	5,033%	5,069%	5,106%	0,7%
	mult_rtnr_vtrafk=1.90	4,943%	4,974%	5,005%	0,6%
	mult_rtnr_vtrafk=2.30	4,895%	4,936%	4,978%	0,8%
	mult_rtnr_vtrafk=2.70	4,881%	4,913%	4,946%	0,7%
	mult_rtnr_vtrafk=3.10	4,873%	4,900%	4,927%	0,6%
1,6	mult_rtnr_vtrafk=0.30	11,938%	11,991%	12,044%	0,4%
	mult_rtnr_vtrafk=0.70	9,831%	9,859%	9,888%	0,3%
	mult_rtnr_vtrafk=1.10	8,947%	8,971%	8,996%	0,3%
	mult_rtnr_vtrafk=1.50	8,562%	8,604%	8,646%	0,5%
	mult_rtnr_vtrafk=1.90	8,396%	8,429%	8,461%	0,4%
	mult_rtnr_vtrafk=2.30	8,280%	8,324%	8,368%	0,5%
	mult_rtnr_vtrafk=2.70	8,218%	8,255%	8,292%	0,4%
	mult_rtnr_vtrafk=3.10	8,156%	8,208%	8,260%	0,6%
1,7	mult_rtnr_vtrafk=0.30	15,852%	15,901%	15,950%	0,3%
	mult_rtnr_vtrafk=0.70	13,396%	13,425%	13,453%	0,2%
	mult_rtnr_vtrafk=1.10	12,160%	12,222%	12,284%	0,5%
	mult_rtnr_vtrafk=1.50	11,789%	11,846%	11,903%	0,5%
	mult_rtnr_vtrafk=1.90	11,629%	11,683%	11,737%	0,5%
	mult_rtnr_vtrafk=2.30	11,534%	11,599%	11,664%	0,6%
	mult_rtnr_vtrafk=2.70	11,491%	11,540%	11,589%	0,4%
	mult_rtnr_vtrafk=3.10	11,473%	11,524%	11,575%	0,4%
1,8	mult_rtnr_vtrafk=0.30	19,420%	19,463%	19,506%	0,2%
	mult_rtnr_vtrafk=0.70	16,423%	16,457%	16,490%	0,2%
	mult_rtnr_vtrafk=1.10	15,269%	15,304%	15,340%	0,2%
	mult_rtnr_vtrafk=1.50	14,989%	15,037%	15,085%	0,3%
	mult_rtnr_vtrafk=1.90	14,857%	14,920%	14,983%	0,4%
	mult_rtnr_vtrafk=2.30	14,839%	14,865%	14,890%	0,2%
	mult_rtnr_vtrafk=2.70	14,804%	14,842%	14,880%	0,3%
	mult_rtnr_vtrafk=3.10	14,774%	14,816%	14,858%	0,3%
1,9	mult_rtnr_vtrafk=0.30	22,547%	22,628%	22,710%	0,4%
	mult_rtnr_vtrafk=0.70	18,965%	19,045%	19,124%	0,4%
	mult_rtnr_vtrafk=1.10	18,255%	18,309%	18,364%	0,3%
	mult_rtnr_vtrafk=1.50	18,059%	18,115%	18,170%	0,3%
	mult_rtnr_vtrafk=1.90	17,947%	18,022%	18,097%	0,4%
	mult_rtnr_vtrafk=2.30	17,922%	17,979%	18,035%	0,3%
	mult_rtnr_vtrafk=2.70	17,905%	17,946%	17,987%	0,2%
	mult_rtnr_vtrafk=3.10	17,880%	17,934%	17,988%	0,3%
2	mult_rtnr_vtrafk=0.30	25,388%	25,482%	25,577%	0,4%
	mult_rtnr_vtrafk=0.70	21,563%	21,646%	21,728%	0,4%
	mult_rtnr_vtrafk=1.10	21,093%	21,186%	21,279%	0,4%
	mult_rtnr_vtrafk=1.50	20,959%	21,035%	21,110%	0,4%
	mult_rtnr_vtrafk=1.90	20,864%	20,956%	21,048%	0,4%
	mult_rtnr_vtrafk=2.30	20,831%	20,912%	20,994%	0,4%
	mult_rtnr_vtrafk=2.70	20,810%	20,896%	20,983%	0,4%
	mult_rtnr_vtrafk=3.10	20,789%	20,878%	20,966%	0,4%

Tab. G-16 – Intervalos de confiança para a probabilidade de bloqueio global no RTNR versus o factor do  $Vtrafk$  (sistema 2).

reserva	0
fluxos	2

alfa 5%

Average of valor				Limite Inferior	Média	Limite Superior	Conf (Rel)
carga	item	metodo	idcorrida				
1	TpbiClas{0}	DAR	mult_dar_res	0,000%	0,001%	0,001%	28,8%
		DIRECTO	mult	0,350%	0,356%	0,362%	1,6%
	TpbiClas{1}	DAR	mult_dar_res	0,001%	0,002%	0,003%	70,6%
		DIRECTO	mult	0,724%	0,740%	0,757%	2,2%
1,1	TpbiClas{0}	DAR	mult_dar_res	0,005%	0,007%	0,008%	16,8%
		DIRECTO	mult	0,605%	0,620%	0,635%	2,4%
	TpbiClas{1}	DAR	mult_dar_res	0,017%	0,022%	0,027%	21,6%
		DIRECTO	mult	1,232%	1,268%	1,304%	2,8%
1,2	TpbiClas{0}	DAR	mult_dar_res	0,058%	0,064%	0,070%	9,2%
		DIRECTO	mult	1,170%	1,206%	1,242%	3,0%
	TpbiClas{1}	DAR	mult_dar_res	0,233%	0,258%	0,283%	9,6%
		DIRECTO	mult	2,332%	2,413%	2,495%	3,4%
1,3	TpbiClas{0}	DAR	mult_dar_res	0,374%	0,400%	0,425%	6,4%
		DIRECTO	mult	2,172%	2,219%	2,267%	2,1%
	TpbiClas{1}	DAR	mult_dar_res	1,384%	1,462%	1,540%	5,3%
		DIRECTO	mult	4,269%	4,354%	4,439%	2,0%
1,4	TpbiClas{0}	DAR	mult_dar_res	1,295%	1,326%	1,357%	2,3%
		DIRECTO	mult	3,558%	3,592%	3,627%	1,0%
	TpbiClas{1}	DAR	mult_dar_res	4,085%	4,176%	4,266%	2,2%
		DIRECTO	mult	6,845%	6,970%	7,095%	1,8%
1,5	TpbiClas{0}	DAR	mult_dar_res	3,121%	3,151%	3,181%	0,9%
		DIRECTO	mult	5,363%	5,384%	5,405%	0,4%
	TpbiClas{1}	DAR	mult_dar_res	8,285%	8,359%	8,432%	0,9%
		DIRECTO	mult	10,163%	10,240%	10,317%	0,8%
1,6	TpbiClas{0}	DAR	mult_dar_res	5,699%	5,743%	5,788%	0,8%
		DIRECTO	mult	7,667%	7,699%	7,730%	0,4%
	TpbiClas{1}	DAR	mult_dar_res	13,537%	13,634%	13,731%	0,7%
		DIRECTO	mult	14,382%	14,452%	14,522%	0,5%
1,7	TpbiClas{0}	DAR	mult_dar_res	8,567%	8,620%	8,674%	0,6%
		DIRECTO	mult	10,234%	10,285%	10,335%	0,5%
	TpbiClas{1}	DAR	mult_dar_res	18,939%	18,987%	19,035%	0,3%
		DIRECTO	mult	18,991%	19,030%	19,069%	0,2%
1,8	TpbiClas{0}	DAR	mult_dar_res	11,474%	11,526%	11,579%	0,5%
		DIRECTO	mult	12,869%	12,925%	12,981%	0,4%
	TpbiClas{1}	DAR	mult_dar_res	23,915%	23,970%	24,026%	0,2%
		DIRECTO	mult	23,485%	23,538%	23,590%	0,2%
1,9	TpbiClas{0}	DAR	mult_dar_res	14,229%	14,273%	14,317%	0,3%
		DIRECTO	mult	15,407%	15,475%	15,543%	0,4%
	TpbiClas{1}	DAR	mult_dar_res	28,392%	28,454%	28,516%	0,2%
		DIRECTO	mult	27,731%	27,804%	27,877%	0,3%
2	TpbiClas{0}	DAR	mult_dar_res	16,813%	16,907%	17,001%	0,6%
		DIRECTO	mult	17,867%	17,968%	18,069%	0,6%
	TpbiClas{1}	DAR	mult_dar_res	32,416%	32,519%	32,623%	0,3%
		DIRECTO	mult	31,682%	31,803%	31,925%	0,4%

Tab. G-17 – Intervalos de confiança para a probabilidade de bloqueio global por classe para o método DAR e DIRECTO (sistema 2).

reserva 0

alfa

5%

Average of valor			Limite Inferior	Média	Limite Superior	Conf (Rel)
carga	item	metodo				
1	TpblClas{0}	DIRECTO	0,350%	0,356%	0,362%	1,6%
		RTNR	0,000%	0,000%	0,000%	0,0%
	TpblClas{1}	DIRECTO	0,724%	0,740%	0,757%	2,2%
		RTNR	0,000%	0,000%	0,000%	0,0%
1,1	TpblClas{0}	DIRECTO	0,605%	0,620%	0,635%	2,4%
		RTNR	0,000%	0,000%	0,000%	277,6%
	TpblClas{1}	DIRECTO	1,232%	1,268%	1,304%	2,8%
		RTNR	0,000%	0,000%	0,000%	277,6%
1,2	TpblClas{0}	DIRECTO	1,170%	1,206%	1,242%	3,0%
		RTNR	0,001%	0,003%	0,004%	60,4%
	TpblClas{1}	DIRECTO	2,332%	2,413%	2,495%	3,4%
		RTNR	0,002%	0,004%	0,005%	46,8%
1,3	TpblClas{0}	DIRECTO	2,172%	2,219%	2,267%	2,1%
		RTNR	0,080%	0,111%	0,142%	28,1%
	TpblClas{1}	DIRECTO	4,269%	4,354%	4,439%	2,0%
		RTNR	0,144%	0,179%	0,214%	19,6%
1,4	TpblClas{0}	DIRECTO	3,558%	3,592%	3,627%	1,0%
		RTNR	1,619%	1,680%	1,741%	3,7%
	TpblClas{1}	DIRECTO	6,845%	6,970%	7,095%	1,8%
		RTNR	1,646%	1,696%	1,745%	2,9%
1,5	TpblClas{0}	DIRECTO	5,363%	5,384%	5,405%	0,4%
		RTNR	5,655%	5,754%	5,853%	1,7%
	TpblClas{1}	DIRECTO	10,163%	10,240%	10,317%	0,8%
		RTNR	2,595%	2,742%	2,889%	5,4%
1,6	TpblClas{0}	DIRECTO	7,667%	7,699%	7,730%	0,4%
		RTNR	9,688%	9,705%	9,723%	0,2%
	TpblClas{1}	DIRECTO	14,382%	14,452%	14,522%	0,5%
		RTNR	5,195%	5,304%	5,413%	2,1%
1,7	TpblClas{0}	DIRECTO	10,234%	10,285%	10,335%	0,5%
		RTNR	12,617%	12,693%	12,769%	0,6%
	TpblClas{1}	DIRECTO	18,991%	19,030%	19,069%	0,2%
		RTNR	9,807%	9,872%	9,937%	0,7%
1,8	TpblClas{0}	DIRECTO	12,869%	12,925%	12,981%	0,4%
		RTNR	15,368%	15,404%	15,439%	0,2%
	TpblClas{1}	DIRECTO	23,485%	23,538%	23,590%	0,2%
		RTNR	14,765%	14,809%	14,854%	0,3%
1,9	TpblClas{0}	DIRECTO	15,407%	15,475%	15,543%	0,4%
		RTNR	18,057%	18,112%	18,167%	0,3%
	TpblClas{1}	DIRECTO	27,731%	27,804%	27,877%	0,3%
		RTNR	19,180%	19,294%	19,409%	0,6%
2	TpblClas{0}	DIRECTO	17,867%	17,968%	18,069%	0,6%
		RTNR	20,673%	20,753%	20,833%	0,4%
	TpblClas{1}	DIRECTO	31,682%	31,803%	31,925%	0,4%
		RTNR	23,186%	23,348%	23,511%	0,7%

Tab. G-18 – Intervalos de confiança para a probabilidade de bloqueio global por classe para o método RTNR e DIRECTO (sistema 2).

fluxos		2		alfa	5%		
Average of valor			Limite Inferior	Média	Limite Superior	Conf (Rel)	
carga	item	metodo					
1	TpblClas{0}	DAR	0,000%	0,000%	0,001%	38,1%	
		RTNR	0,000%	0,004%	0,015%	277,6%	
	TpblClas{1}	DAR	0,001%	0,002%	0,003%	56,2%	
		RTNR	0,000%	0,000%	0,000%	0,0%	
1,1	TpblClas{0}	DAR	0,003%	0,004%	0,005%	16,8%	
		RTNR	0,000%	0,000%	0,000%	185,1%	
	TpblClas{1}	DAR	0,015%	0,016%	0,018%	8,8%	
		RTNR	0,000%	0,000%	0,000%	277,6%	
1,2	TpblClas{0}	DAR	0,041%	0,046%	0,050%	9,6%	
		RTNR	0,001%	0,003%	0,004%	57,6%	
	TpblClas{1}	DAR	0,159%	0,178%	0,196%	10,2%	
		RTNR	0,002%	0,004%	0,006%	40,2%	
1,3	TpblClas{0}	DAR	0,353%	0,379%	0,405%	7,0%	
		RTNR	0,085%	0,109%	0,134%	22,0%	
	TpblClas{1}	DAR	1,013%	1,083%	1,153%	6,5%	
		RTNR	0,147%	0,178%	0,209%	17,3%	
1,4	TpblClas{0}	DAR	1,825%	1,860%	1,895%	1,9%	
		RTNR	1,619%	1,678%	1,737%	3,5%	
	TpblClas{1}	DAR	3,141%	3,220%	3,299%	2,5%	
		RTNR	1,576%	1,620%	1,664%	2,7%	
1,5	TpblClas{0}	DAR	4,445%	4,469%	4,494%	0,6%	
		RTNR	5,664%	5,731%	5,799%	1,2%	
	TpblClas{1}	DAR	6,246%	6,314%	6,383%	1,1%	
		RTNR	2,621%	2,705%	2,788%	3,1%	
1,6	TpblClas{0}	DAR	7,553%	7,592%	7,632%	0,5%	
		RTNR	9,600%	9,635%	9,671%	0,4%	
	TpblClas{1}	DAR	10,140%	10,244%	10,348%	1,0%	
		RTNR	5,260%	5,346%	5,433%	1,6%	
1,7	TpblClas{0}	DAR	10,663%	10,726%	10,788%	0,6%	
		RTNR	12,561%	12,640%	12,719%	0,6%	
	TpblClas{1}	DAR	14,636%	14,673%	14,710%	0,2%	
		RTNR	9,878%	9,921%	9,964%	0,4%	
1,8	TpblClas{0}	DAR	13,697%	13,738%	13,779%	0,3%	
		RTNR	15,354%	15,400%	15,447%	0,3%	
	TpblClas{1}	DAR	19,090%	19,144%	19,198%	0,3%	
		RTNR	14,764%	14,808%	14,851%	0,3%	
1,9	TpblClas{0}	DAR	16,499%	16,545%	16,591%	0,3%	
		RTNR	18,049%	18,107%	18,165%	0,3%	
	TpblClas{1}	DAR	23,289%	23,360%	23,432%	0,3%	
		RTNR	19,162%	19,277%	19,392%	0,6%	
2	TpblClas{0}	DAR	19,141%	19,224%	19,307%	0,4%	
		RTNR	20,678%	20,758%	20,839%	0,4%	
	TpblClas{1}	DAR	27,157%	27,267%	27,376%	0,4%	
		RTNR	23,178%	23,314%	23,450%	0,6%	

Tab. G-19 – Intervalos de confiança para a probabilidade de bloqueio global por classe para o método RTNR e DAR (sistema 2).

metodo	DAR
fluxos	2
reserva	0
carga	1,4

alfa 5%

Average of valor			Limite Inferior	Média	Limite Superior	Conf (Rel)
origem	destino	item				
1	1	Ppbl{0}	0,000%	0,000%	0,000%	0,0%
		Ppbl{1}	0,000%	0,000%	0,000%	0,0%
	2	Ppbl{0}	5,276%	5,497%	5,717%	4,0%
		Ppbl{1}	15,716%	16,392%	17,069%	4,1%
	3	Ppbl{0}	2,041%	2,269%	2,496%	10,0%
		Ppbl{1}	5,798%	6,703%	7,607%	13,5%
	4	Ppbl{0}	0,112%	0,128%	0,145%	12,9%
		Ppbl{1}	0,253%	0,324%	0,394%	21,7%
	5	Ppbl{0}	0,942%	1,054%	1,167%	10,7%
		Ppbl{1}	3,257%	3,520%	3,783%	7,5%
	6	Ppbl{0}	2,182%	2,243%	2,304%	2,7%
		Ppbl{1}	5,380%	5,615%	5,849%	4,2%
2	1	Ppbl{0}	5,264%	5,457%	5,649%	3,5%
		Ppbl{1}	16,129%	16,667%	17,205%	3,2%
	2	Ppbl{0}	0,000%	0,000%	0,000%	0,0%
		Ppbl{1}	0,000%	0,000%	0,000%	0,0%
	3	Ppbl{0}	0,087%	0,102%	0,116%	14,6%
		Ppbl{1}	0,252%	0,315%	0,378%	20,0%
	4	Ppbl{0}	4,434%	4,522%	4,611%	2,0%
		Ppbl{1}	13,308%	13,686%	14,063%	2,8%
	5	Ppbl{0}	0,794%	0,872%	0,949%	8,9%
		Ppbl{1}	3,420%	3,644%	3,868%	6,2%
	6	Ppbl{0}	2,192%	2,259%	2,326%	3,0%
		Ppbl{1}	7,163%	7,303%	7,443%	1,9%
3	1	Ppbl{0}	2,149%	2,256%	2,363%	4,8%
		Ppbl{1}	5,936%	6,836%	7,737%	13,2%
	2	Ppbl{0}	0,085%	0,106%	0,126%	19,5%
		Ppbl{1}	0,303%	0,333%	0,364%	9,1%
	3	Ppbl{0}	0,000%	0,000%	0,000%	0,0%
		Ppbl{1}	0,000%	0,000%	0,000%	0,0%
	4	Ppbl{0}	4,424%	4,866%	5,308%	9,1%
		Ppbl{1}	16,363%	17,867%	19,371%	8,4%
	5	Ppbl{0}	0,315%	0,363%	0,411%	13,2%
		Ppbl{1}	0,749%	1,086%	1,423%	31,0%
	6	Ppbl{0}	0,958%	1,073%	1,187%	10,6%
		Ppbl{1}	4,682%	5,083%	5,484%	7,9%
4	1	Ppbl{0}	0,104%	0,121%	0,138%	14,2%
		Ppbl{1}	0,275%	0,338%	0,400%	18,5%
	2	Ppbl{0}	4,484%	4,578%	4,672%	2,1%
		Ppbl{1}	13,110%	13,700%	14,291%	4,3%
	3	Ppbl{0}	4,736%	4,853%	4,971%	2,4%
		Ppbl{1}	16,569%	18,182%	19,794%	8,9%
	4	Ppbl{0}	0,000%	0,000%	0,000%	0,0%
		Ppbl{1}	0,000%	0,000%	0,000%	0,0%
	5	Ppbl{0}	0,398%	0,422%	0,445%	5,5%
		Ppbl{1}	1,181%	1,272%	1,363%	7,1%
	6	Ppbl{0}	1,103%	1,176%	1,248%	6,2%
		Ppbl{1}	4,158%	4,205%	4,251%	1,1%
5	1	Ppbl{0}	0,972%	1,062%	1,152%	8,5%
		Ppbl{1}	2,794%	3,224%	3,655%	13,4%
	2	Ppbl{0}	0,811%	0,859%	0,907%	5,6%
		Ppbl{1}	3,681%	3,740%	3,798%	1,6%
	3	Ppbl{0}	0,273%	0,369%	0,465%	26,1%
		Ppbl{1}	1,027%	1,282%	1,537%	19,9%
	4	Ppbl{0}	0,376%	0,415%	0,454%	9,3%
		Ppbl{1}	1,104%	1,198%	1,292%	7,9%
	5	Ppbl{0}	0,000%	0,000%	0,000%	0,0%
		Ppbl{1}	0,000%	0,000%	0,000%	0,0%
	6	Ppbl{0}	0,000%	0,000%	0,000%	0,0%
		Ppbl{1}	0,000%	0,000%	0,000%	0,0%
6	1	Ppbl{0}	2,205%	2,271%	2,336%	2,9%
		Ppbl{1}	5,088%	5,419%	5,750%	6,1%
	2	Ppbl{0}	2,220%	2,305%	2,390%	3,7%
		Ppbl{1}	7,167%	7,368%	7,569%	2,7%
	3	Ppbl{0}	1,014%	1,095%	1,175%	7,4%
		Ppbl{1}	4,743%	5,129%	5,515%	7,5%
	4	Ppbl{0}	1,077%	1,137%	1,197%	5,3%
		Ppbl{1}	3,908%	4,046%	4,184%	3,4%
	5	Ppbl{0}	0,000%	0,000%	0,000%	0,0%
		Ppbl{1}	0,000%	0,000%	0,000%	0,0%
	6	Ppbl{0}	0,000%	0,000%	0,000%	0,0%
		Ppbl{1}	0,000%	0,000%	0,000%	0,0%

Tab. G-20 – Intervalos de confiança para a probabilidade de bloqueio ponto a ponto, por classe, para o método DAR (sistema 2).

metodo	RTNR
fluxos	2
reserva	0
carga	1,4

alfa 5%

Average of valor			Limite Inferior	Média	Limite Superior	Conf (Rel)
origem	destino	item				
1	1	Ppbl(0)	0,000%	0,000%	0,000%	0,0%
		Ppbl(1)	0,000%	0,000%	0,000%	0,0%
	2	Ppbl(0)	4,742%	5,051%	5,360%	6,1%
		Ppbl(1)	4,500%	4,860%	5,221%	7,4%
	3	Ppbl(0)	4,196%	4,383%	4,569%	4,3%
		Ppbl(1)	13,758%	14,543%	15,327%	5,4%
	4	Ppbl(0)	0,368%	0,390%	0,412%	5,6%
		Ppbl(1)	0,432%	0,505%	0,578%	14,5%
	5	Ppbl(0)	1,377%	1,498%	1,620%	8,1%
		Ppbl(1)	1,573%	1,826%	2,080%	13,9%
	6	Ppbl(0)	2,283%	2,398%	2,512%	4,8%
		Ppbl(1)	3,329%	3,566%	3,803%	6,6%
2	1	Ppbl(0)	6,239%	6,619%	6,998%	5,7%
		Ppbl(1)	5,697%	6,178%	6,659%	7,8%
	2	Ppbl(0)	0,000%	0,000%	0,000%	0,0%
		Ppbl(1)	0,000%	0,000%	0,000%	0,0%
	3	Ppbl(0)	1,509%	1,629%	1,748%	7,3%
		Ppbl(1)	2,347%	2,504%	2,661%	6,3%
	4	Ppbl(0)	4,273%	4,495%	4,716%	4,9%
		Ppbl(1)	3,519%	3,649%	3,780%	3,6%
	5	Ppbl(0)	2,111%	2,179%	2,247%	3,1%
		Ppbl(1)	1,463%	1,849%	2,236%	20,9%
	6	Ppbl(0)	2,755%	2,887%	3,020%	4,6%
		Ppbl(1)	2,439%	2,642%	2,846%	7,7%
3	1	Ppbl(0)	4,739%	5,166%	5,593%	8,3%
		Ppbl(1)	12,531%	14,092%	15,654%	11,1%
	2	Ppbl(0)	1,319%	1,483%	1,647%	11,1%
		Ppbl(1)	2,105%	2,331%	2,558%	9,7%
	3	Ppbl(0)	0,000%	0,000%	0,000%	0,0%
		Ppbl(1)	0,000%	0,000%	0,000%	0,0%
	4	Ppbl(0)	3,635%	3,993%	4,352%	9,0%
		Ppbl(1)	4,634%	5,549%	6,465%	16,5%
	5	Ppbl(0)	1,724%	1,876%	2,027%	8,1%
		Ppbl(1)	4,036%	4,430%	4,825%	8,9%
	6	Ppbl(0)	1,935%	2,103%	2,270%	8,0%
		Ppbl(1)	3,734%	4,406%	5,078%	15,2%
4	1	Ppbl(0)	0,398%	0,433%	0,469%	8,3%
		Ppbl(1)	0,478%	0,515%	0,552%	7,2%
	2	Ppbl(0)	4,136%	4,355%	4,574%	5,0%
		Ppbl(1)	3,302%	3,429%	3,555%	3,7%
	3	Ppbl(0)	3,530%	3,694%	3,857%	4,4%
		Ppbl(1)	5,360%	5,882%	6,403%	8,9%
	4	Ppbl(0)	0,000%	0,000%	0,000%	0,0%
		Ppbl(1)	0,000%	0,000%	0,000%	0,0%
	5	Ppbl(0)	1,045%	1,096%	1,148%	4,7%
		Ppbl(1)	1,075%	1,105%	1,135%	2,7%
	6	Ppbl(0)	1,947%	2,060%	2,172%	5,5%
		Ppbl(1)	0,978%	1,058%	1,138%	7,5%
5	1	Ppbl(0)	1,345%	1,517%	1,689%	11,3%
		Ppbl(1)	1,653%	1,867%	2,080%	11,5%
	2	Ppbl(0)	1,807%	1,905%	2,004%	5,2%
		Ppbl(1)	1,299%	1,461%	1,622%	11,0%
	3	Ppbl(0)	1,529%	1,659%	1,789%	7,8%
		Ppbl(1)	3,935%	4,460%	4,984%	11,8%
	4	Ppbl(0)	0,939%	0,970%	1,002%	3,3%
		Ppbl(1)	0,975%	1,034%	1,094%	5,7%
	5	Ppbl(0)	0,000%	0,000%	0,000%	0,0%
		Ppbl(1)	0,000%	0,000%	0,000%	0,0%
	6	Ppbl(0)	0,000%	0,000%	0,000%	0,0%
		Ppbl(1)	0,000%	0,000%	0,000%	0,0%
6	1	Ppbl(0)	2,341%	2,498%	2,654%	6,3%
		Ppbl(1)	2,968%	3,349%	3,730%	11,4%
	2	Ppbl(0)	2,576%	2,644%	2,712%	2,6%
		Ppbl(1)	2,321%	2,467%	2,614%	5,9%
	3	Ppbl(0)	1,845%	1,960%	2,075%	5,8%
		Ppbl(1)	4,238%	4,665%	5,093%	9,2%
	4	Ppbl(0)	1,401%	1,617%	1,833%	13,4%
		Ppbl(1)	0,837%	0,943%	1,049%	11,3%
	5	Ppbl(0)	0,000%	0,000%	0,000%	0,0%
		Ppbl(1)	0,000%	0,000%	0,000%	0,0%
	6	Ppbl(0)	0,000%	0,000%	0,000%	0,0%
		Ppbl(1)	0,000%	0,000%	0,000%	0,0%

Tab. G-21 – Intervalos de confiança para a probabilidade de bloqueio ponto a ponto, por classe, para o método RTNR (sistema 2).

## G.3 Intervalos de confiança da experiência com o Sistema 3

item	Tpbl	alfa	5%
reserva	0		
idcorrida	mult		

Average of valor		Limite Inferior	Média	Limite Superior	Conf (Rel)
carga	metodo				
1	DAR	0,010%	0,012%	0,014%	18,0%
	DIRECTO	0,442%	0,453%	0,465%	2,5%
	RTNR	0,000%	0,007%	0,016%	133,9%
1,1	DAR	0,027%	0,028%	0,030%	5,4%
	DIRECTO	0,683%	0,704%	0,724%	2,9%
	RTNR	0,007%	0,008%	0,009%	12,3%
1,2	DAR	0,075%	0,082%	0,088%	7,8%
	DIRECTO	1,271%	1,293%	1,315%	1,7%
	RTNR	0,027%	0,029%	0,031%	6,4%
1,3	DAR	0,296%	0,308%	0,319%	3,7%
	DIRECTO	2,311%	2,345%	2,378%	1,4%
	RTNR	0,109%	0,127%	0,145%	14,3%
1,4	DAR	1,049%	1,083%	1,117%	3,1%
	DIRECTO	3,690%	3,748%	3,807%	1,6%
	RTNR	0,707%	0,747%	0,788%	5,4%
1,5	DAR	2,745%	2,783%	2,820%	1,4%
	DIRECTO	5,365%	5,432%	5,499%	1,2%
	RTNR	2,891%	2,967%	3,043%	2,6%
1,6	DAR	5,184%	5,253%	5,322%	1,3%
	DIRECTO	7,246%	7,312%	7,378%	0,9%
	RTNR	6,071%	6,196%	6,321%	2,0%
1,7	DAR	7,993%	8,046%	8,099%	0,7%
	DIRECTO	9,238%	9,302%	9,367%	0,7%
	RTNR	9,276%	9,415%	9,554%	1,5%
1,8	DAR	10,724%	10,823%	10,923%	0,9%
	DIRECTO	11,315%	11,367%	11,419%	0,5%
	RTNR	12,279%	12,382%	12,485%	0,8%
1,9	DAR	13,559%	13,608%	13,657%	0,4%
	DIRECTO	13,408%	13,469%	13,530%	0,5%
	RTNR	15,098%	15,163%	15,228%	0,4%
2	DAR	16,364%	16,384%	16,405%	0,1%
	DIRECTO	15,515%	15,548%	15,580%	0,2%
	RTNR	17,692%	17,740%	17,788%	0,3%

Tab. G-22 – Intervalos de confiança para a congestão de chamadas na rede (sistema 3).

item	Tpbl
reserva	0

alfa

5%

Average of valor			Limite Inferior	Média	Limite Superior	Conf (Rel)
carga	metodo	idcorrida				
1	DAR	mult	0,010%	0,012%	0,014%	18,0%
		mult_dar_res	0,027%	0,029%	0,031%	6,2%
		mult_dar_res_b	0,014%	0,017%	0,019%	13,0%
	DIRECTO	mult	0,442%	0,453%	0,465%	2,5%
1,1	DAR	mult	0,027%	0,028%	0,030%	5,4%
		mult_dar_res	0,052%	0,054%	0,055%	2,7%
		mult_dar_res_b	0,040%	0,043%	0,045%	6,3%
	DIRECTO	mult	0,683%	0,704%	0,724%	2,9%
1,2	DAR	mult	0,075%	0,082%	0,088%	7,8%
		mult_dar_res	0,127%	0,133%	0,139%	4,7%
		mult_dar_res_b	0,133%	0,137%	0,142%	3,3%
	DIRECTO	mult	1,271%	1,293%	1,315%	1,7%
1,3	DAR	mult	0,296%	0,308%	0,319%	3,7%
		mult_dar_res	0,456%	0,473%	0,490%	3,6%
		mult_dar_res_b	0,505%	0,520%	0,535%	2,9%
	DIRECTO	mult	2,311%	2,345%	2,378%	1,4%
1,4	DAR	mult	1,049%	1,083%	1,117%	3,1%
		mult_dar_res	1,276%	1,314%	1,353%	3,0%
		mult_dar_res_b	1,401%	1,444%	1,487%	3,0%
	DIRECTO	mult	3,690%	3,748%	3,807%	1,6%
1,5	DAR	mult	2,745%	2,783%	2,820%	1,4%
		mult_dar_res	2,708%	2,745%	2,782%	1,3%
		mult_dar_res_b	2,932%	2,981%	3,029%	1,6%
	DIRECTO	mult	5,365%	5,432%	5,499%	1,2%
1,6	DAR	mult	5,184%	5,253%	5,322%	1,3%
		mult_dar_res	4,612%	4,682%	4,753%	1,5%
		mult_dar_res_b	4,973%	5,033%	5,093%	1,2%
	DIRECTO	mult	7,246%	7,312%	7,378%	0,9%
1,7	DAR	mult	7,993%	8,046%	8,099%	0,7%
		mult_dar_res	6,839%	6,905%	6,971%	1,0%
		mult_dar_res_b	7,252%	7,317%	7,382%	0,9%
	DIRECTO	mult	9,238%	9,302%	9,367%	0,7%
1,8	DAR	mult	10,724%	10,823%	10,923%	0,9%
		mult_dar_res	9,163%	9,250%	9,338%	0,9%
		mult_dar_res_b	9,645%	9,708%	9,770%	0,6%
	DIRECTO	mult	11,315%	11,367%	11,419%	0,5%
1,9	DAR	mult	13,559%	13,608%	13,657%	0,4%
		mult_dar_res	11,572%	11,632%	11,691%	0,5%
		mult_dar_res_b	12,018%	12,070%	12,123%	0,4%
	DIRECTO	mult	13,408%	13,469%	13,530%	0,5%
2	DAR	mult	16,364%	16,384%	16,405%	0,1%
		mult_dar_res	13,933%	13,972%	14,012%	0,3%
		mult_dar_res_b	14,340%	14,360%	14,380%	0,1%
	DIRECTO	mult	15,515%	15,548%	15,580%	0,2%

Tab. G-23 – Intervalos de confiança para a congestão de chamadas na rede, com o DAR, para reserva de circuitos fixa diferente de 0 (sistema 3).

item	Tpbl
reserva	0

alfa

5%

Average of valor			Limite Inferior	Média	Limite Superior	Conf (Rel)
carga	metodo	idcorrida				
1	DIRECTO	mult	0,442%	0,453%	0,465%	2,5%
	RTNR	mult	0,000%	0,007%	0,016%	133,9%
		mult_a	0,000%	0,003%	0,007%	101,7%
1,1	DIRECTO	mult	0,683%	0,704%	0,724%	2,9%
	RTNR	mult	0,007%	0,008%	0,009%	12,3%
		mult_a	0,008%	0,008%	0,009%	10,7%
1,2	DIRECTO	mult	1,271%	1,293%	1,315%	1,7%
	RTNR	mult	0,027%	0,029%	0,031%	6,4%
		mult_a	0,026%	0,028%	0,030%	6,8%
1,3	DIRECTO	mult	2,311%	2,345%	2,378%	1,4%
	RTNR	mult	0,109%	0,127%	0,145%	14,3%
		mult_a	0,113%	0,120%	0,126%	5,2%
1,4	DIRECTO	mult	3,690%	3,748%	3,807%	1,6%
	RTNR	mult	0,707%	0,747%	0,788%	5,4%
		mult_a	0,797%	0,867%	0,936%	8,0%
1,5	DIRECTO	mult	5,365%	5,432%	5,499%	1,2%
	RTNR	mult	2,891%	2,967%	3,043%	2,6%
		mult_a	4,762%	4,845%	4,928%	1,7%
1,6	DIRECTO	mult	7,246%	7,312%	7,378%	0,9%
	RTNR	mult	6,071%	6,196%	6,321%	2,0%
		mult_a	8,030%	8,134%	8,238%	1,3%
1,7	DIRECTO	mult	9,238%	9,302%	9,367%	0,7%
	RTNR	mult	9,276%	9,415%	9,554%	1,5%
		mult_a	11,031%	11,114%	11,197%	0,7%
1,8	DIRECTO	mult	11,315%	11,367%	11,419%	0,5%
	RTNR	mult	12,279%	12,382%	12,485%	0,8%
		mult_a	13,851%	13,948%	14,044%	0,7%
1,9	DIRECTO	mult	13,408%	13,469%	13,530%	0,5%
	RTNR	mult	15,098%	15,163%	15,228%	0,4%
		mult_a	16,590%	16,677%	16,763%	0,5%
2	DIRECTO	mult	15,515%	15,548%	15,580%	0,2%
	RTNR	mult	17,692%	17,740%	17,788%	0,3%
		mult_a	19,119%	19,210%	19,302%	0,5%

Tab. G-24 – Intervalos de confiança para a congestão de chamadas na rede, com o RTNR, para duas diferentes escolhas dos valores da largura de banda mínima garantida (sistema 3).

item	Tpbl
fluxos	4

alfa

5%

Average of valor				Limite Inferior	Média	Limite Superior	Conf (Rel)
carga	idcorrida	metodo	reserva				
1	mult	DAR	0	0,010%	0,012%	0,014%	18,0%
			1	0,010%	0,012%	0,014%	18,0%
		RTNR	0	0,000%	0,007%	0,016%	133,9%
			1	0,000%	0,003%	0,005%	88,0%
	mult_dar_res	DAR	0	0,027%	0,029%	0,031%	6,2%
1,1	mult	DAR	0	0,027%	0,028%	0,030%	5,4%
			1	0,028%	0,029%	0,030%	3,8%
		RTNR	0	0,007%	0,008%	0,009%	12,3%
			1	0,009%	0,009%	0,009%	1,8%
	mult_dar_res	DAR	0	0,052%	0,054%	0,055%	2,7%
1,2	mult	DAR	0	0,075%	0,082%	0,088%	7,8%
			1	0,078%	0,081%	0,084%	4,0%
		RTNR	0	0,027%	0,029%	0,031%	6,4%
			1	0,027%	0,028%	0,029%	2,6%
	mult_dar_res	DAR	0	0,127%	0,133%	0,139%	4,7%
1,3	mult	DAR	0	0,296%	0,308%	0,319%	3,7%
			1	0,331%	0,347%	0,364%	4,9%
		RTNR	0	0,109%	0,127%	0,145%	14,3%
			1	0,098%	0,139%	0,180%	29,6%
	mult_dar_res	DAR	0	0,456%	0,473%	0,490%	3,6%
1,4	mult	DAR	0	1,049%	1,083%	1,117%	3,1%
			1	1,505%	1,556%	1,606%	3,2%
		RTNR	0	0,707%	0,747%	0,788%	5,4%
			1	0,733%	0,765%	0,798%	4,2%
	mult_dar_res	DAR	0	1,276%	1,314%	1,353%	3,0%
1,5	mult	DAR	0	2,745%	2,783%	2,820%	1,4%
			1	4,134%	4,210%	4,287%	1,8%
		RTNR	0	2,891%	2,967%	3,043%	2,6%
			1	2,886%	2,975%	3,063%	3,0%
	mult_dar_res	DAR	0	2,708%	2,745%	2,782%	1,3%
1,6	mult	DAR	0	5,184%	5,253%	5,322%	1,3%
			1	7,744%	7,848%	7,952%	1,3%
		RTNR	0	6,071%	6,196%	6,321%	2,0%
			1	6,067%	6,150%	6,233%	1,4%
	mult_dar_res	DAR	0	4,612%	4,682%	4,753%	1,5%

Tab. G-25 – Intervalos de confiança para a probabilidade de bloqueio global de chamadas na rede, para os métodos DAR e RTNR (sistema 3).

item	TpblCirc
fluxos	4
reserva	0

alfa

5%

Average of valor			Limite Inferior	Média	Limite Superior	Conf (Rel)
carga	metodo	idcorrida				
1	DAR	mult	0,139%	0,162%	0,185%	14,4%
		mult_dar_res	0,397%	0,415%	0,433%	4,4%
	DIRECTO	mult	1,005%	1,026%	1,047%	2,1%
	RTNR	mult	0,000%	0,092%	0,241%	161,7%
1,1	DAR	mult	0,329%	0,343%	0,357%	4,2%
		mult_dar_res	0,617%	0,624%	0,632%	1,2%
	DIRECTO	mult	1,396%	1,444%	1,493%	3,4%
	RTNR	mult	0,112%	0,131%	0,150%	14,5%
1,2	DAR	mult	0,597%	0,634%	0,671%	5,9%
		mult_dar_res	0,924%	0,962%	1,001%	4,0%
	DIRECTO	mult	2,143%	2,172%	2,202%	1,3%
	RTNR	mult	0,391%	0,403%	0,415%	2,9%
1,3	DAR	mult	1,247%	1,278%	1,309%	2,4%
		mult_dar_res	1,613%	1,658%	1,702%	2,7%
	DIRECTO	mult	3,302%	3,332%	3,363%	0,9%
	RTNR	mult	0,840%	0,978%	1,116%	14,1%
1,4	DAR	mult	2,703%	2,762%	2,821%	2,1%
		mult_dar_res	2,955%	3,002%	3,050%	1,6%
	DIRECTO	mult	4,830%	4,893%	4,956%	1,3%
	RTNR	mult	2,320%	2,420%	2,519%	4,1%
1,5	DAR	mult	5,589%	5,681%	5,772%	1,6%
		mult_dar_res	5,203%	5,276%	5,348%	1,4%
	DIRECTO	mult	6,840%	6,927%	7,013%	1,3%
	RTNR	mult	5,588%	5,705%	5,822%	2,1%
1,6	DAR	mult	9,595%	9,754%	9,912%	1,6%
		mult_dar_res	8,163%	8,316%	8,470%	1,8%
	DIRECTO	mult	9,373%	9,505%	9,637%	1,4%
	RTNR	mult	9,080%	9,220%	9,360%	1,5%
1,7	DAR	mult	13,914%	14,072%	14,229%	1,1%
		mult_dar_res	11,571%	11,725%	11,878%	1,3%
	DIRECTO	mult	12,286%	12,431%	12,575%	1,2%
	RTNR	mult	12,576%	12,751%	12,927%	1,4%
1,8	DAR	mult	17,968%	18,134%	18,299%	0,9%
		mult_dar_res	15,044%	15,200%	15,356%	1,0%
	DIRECTO	mult	15,367%	15,496%	15,624%	0,8%
	RTNR	mult	15,944%	16,096%	16,249%	0,9%
1,9	DAR	mult	21,860%	21,969%	22,079%	0,5%
		mult_dar_res	18,391%	18,502%	18,613%	0,6%
	DIRECTO	mult	18,466%	18,596%	18,726%	0,7%
	RTNR	mult	19,149%	19,267%	19,386%	0,6%
2	DAR	mult	25,496%	25,569%	25,642%	0,3%
		mult_dar_res	21,600%	21,663%	21,727%	0,3%
	DIRECTO	mult	21,471%	21,570%	21,669%	0,5%
	RTNR	mult	22,131%	22,213%	22,295%	0,4%

Tab. G-26 – Intervalos de confiança para a probabilidade de bloqueio global de circuitos na rede (sistema 3).

item	TpblCirc
fluxos	4
reserva	0

alfa 5%

Average of valor			Limite Inferior	Média	Limite Superior	Conf (Rel)
carga	metodo	idcorrida				
1	DAR	mult	0,139%	0,162%	0,185%	14,4%
		mult_dar_res	0,397%	0,415%	0,433%	4,4%
		mult_dar_res_b	0,154%	0,175%	0,196%	12,0%
	DIRECTO	mult	1,005%	1,026%	1,047%	2,1%
1,1	DAR	mult	0,329%	0,343%	0,357%	4,2%
		mult_dar_res	0,617%	0,624%	0,632%	1,2%
		mult_dar_res_b	0,345%	0,363%	0,381%	5,0%
	DIRECTO	mult	1,396%	1,444%	1,493%	3,4%
1,2	DAR	mult	0,597%	0,634%	0,671%	5,9%
		mult_dar_res	0,924%	0,962%	1,001%	4,0%
		mult_dar_res_b	0,670%	0,707%	0,745%	5,3%
	DIRECTO	mult	2,143%	2,172%	2,202%	1,3%
1,3	DAR	mult	1,247%	1,278%	1,309%	2,4%
		mult_dar_res	1,613%	1,658%	1,702%	2,7%
		mult_dar_res_b	1,388%	1,425%	1,463%	2,6%
	DIRECTO	mult	3,302%	3,332%	3,363%	0,9%
1,4	DAR	mult	2,703%	2,762%	2,821%	2,1%
		mult_dar_res	2,955%	3,002%	3,050%	1,6%
		mult_dar_res_b	2,794%	2,860%	2,926%	2,3%
	DIRECTO	mult	4,830%	4,893%	4,956%	1,3%
1,5	DAR	mult	5,589%	5,681%	5,772%	1,6%
		mult_dar_res	5,203%	5,276%	5,348%	1,4%
		mult_dar_res_b	5,096%	5,183%	5,269%	1,7%
	DIRECTO	mult	6,840%	6,927%	7,013%	1,3%
1,6	DAR	mult	9,595%	9,754%	9,912%	1,6%
		mult_dar_res	8,163%	8,316%	8,470%	1,8%
		mult_dar_res_b	8,197%	8,327%	8,457%	1,6%
	DIRECTO	mult	9,373%	9,505%	9,637%	1,4%
1,7	DAR	mult	13,914%	14,072%	14,229%	1,1%
		mult_dar_res	11,571%	11,725%	11,878%	1,3%
		mult_dar_res_b	11,613%	11,768%	11,923%	1,3%
	DIRECTO	mult	12,286%	12,431%	12,575%	1,2%
1,8	DAR	mult	17,968%	18,134%	18,299%	0,9%
		mult_dar_res	15,044%	15,200%	15,356%	1,0%
		mult_dar_res_b	15,121%	15,247%	15,373%	0,8%
	DIRECTO	mult	15,367%	15,496%	15,624%	0,8%
1,9	DAR	mult	21,860%	21,969%	22,079%	0,5%
		mult_dar_res	18,391%	18,502%	18,613%	0,6%
		mult_dar_res_b	18,477%	18,586%	18,694%	0,6%
	DIRECTO	mult	18,466%	18,596%	18,726%	0,7%
2	DAR	mult	25,496%	25,569%	25,642%	0,3%
		mult_dar_res	21,600%	21,663%	21,727%	0,3%
		mult_dar_res_b	21,648%	21,707%	21,766%	0,3%
	DIRECTO	mult	21,471%	21,570%	21,669%	0,5%

Tab. G-27 – Intervalos de confiança para a probabilidade de bloqueio global de circuitos na rede, com o DAR, para diferentes valores da reserva de circuitos fixa (sistema 3).

item	TpblCirc
fluxos	4
reserva	0

alfa

5%

Average of valor			Limite Inferior	Média	Limite Superior	Conf (Rel)
carga	metodo	idcorrida				
1	DIRECTO	mult	1,005%	1,026%	1,047%	2,1%
	RTNR	mult	0,000%	0,092%	0,241%	161,7%
		mult_a	0,024%	0,029%	0,034%	16,2%
1,1	DIRECTO	mult	1,396%	1,444%	1,493%	3,4%
	RTNR	mult	0,112%	0,131%	0,150%	14,5%
		mult_a	0,123%	0,136%	0,150%	10,0%
1,2	DIRECTO	mult	2,143%	2,172%	2,202%	1,3%
	RTNR	mult	0,391%	0,403%	0,415%	2,9%
		mult_a	0,378%	0,419%	0,461%	9,9%
1,3	DIRECTO	mult	3,302%	3,332%	3,363%	0,9%
	RTNR	mult	0,840%	0,978%	1,116%	14,1%
		mult_a	0,905%	0,923%	0,941%	2,0%
1,4	DIRECTO	mult	4,830%	4,893%	4,956%	1,3%
	RTNR	mult	2,320%	2,420%	2,519%	4,1%
		mult_a	2,419%	2,558%	2,697%	5,4%
1,5	DIRECTO	mult	6,840%	6,927%	7,013%	1,3%
	RTNR	mult	5,588%	5,705%	5,822%	2,1%
		mult_a	6,714%	6,817%	6,921%	1,5%
1,6	DIRECTO	mult	9,373%	9,505%	9,637%	1,4%
	RTNR	mult	9,080%	9,220%	9,360%	1,5%
		mult_a	10,184%	10,339%	10,493%	1,5%
1,7	DIRECTO	mult	12,286%	12,431%	12,575%	1,2%
	RTNR	mult	12,576%	12,751%	12,927%	1,4%
		mult_a	13,523%	13,640%	13,758%	0,9%
1,8	DIRECTO	mult	15,367%	15,496%	15,624%	0,8%
	RTNR	mult	15,944%	16,096%	16,249%	0,9%
		mult_a	16,731%	16,863%	16,995%	0,8%
1,9	DIRECTO	mult	18,466%	18,596%	18,726%	0,7%
	RTNR	mult	19,149%	19,267%	19,386%	0,6%
		mult_a	19,796%	19,921%	20,047%	0,6%
2	DIRECTO	mult	21,471%	21,570%	21,669%	0,5%
	RTNR	mult	22,131%	22,213%	22,295%	0,4%
		mult_a	22,694%	22,793%	22,891%	0,4%

Tab. G-28 – Intervalos de confiança para a probabilidade de bloqueio global de circuitos na rede, com o RTNR, para duas diferentes escolhas dos valores da largura de banda mínima garantida (sistema 3).

metodo	RTNR	alfa	5%		
item	Tpbl				
fluxos	4				
Average of valor		Limite Inferior	Média	Limite Superior	Conf (Rel)
carga	idcorrida				
1	mult_rtnr_vtrafk=0.30	0,001%	0,002%	0,002%	23,8%
	mult_rtnr_vtrafk=0.70	0,001%	0,003%	0,005%	71,8%
	mult_rtnr_vtrafk=1.10	0,001%	0,002%	0,002%	29,3%
	mult_rtnr_vtrafk=1.50	0,001%	0,002%	0,003%	53,9%
	mult_rtnr_vtrafk=1.90	0,001%	0,002%	0,002%	34,0%
	mult_rtnr_vtrafk=2.30	0,000%	0,004%	0,009%	137,8%
	mult_rtnr_vtrafk=2.70	0,000%	0,005%	0,010%	114,0%
1,1	mult_rtnr_vtrafk=3.10	0,001%	0,002%	0,002%	46,4%
	mult_rtnr_vtrafk=0.30	0,007%	0,008%	0,009%	10,4%
	mult_rtnr_vtrafk=0.70	0,007%	0,008%	0,009%	8,5%
	mult_rtnr_vtrafk=1.10	0,008%	0,008%	0,009%	6,9%
	mult_rtnr_vtrafk=1.50	0,008%	0,008%	0,008%	4,0%
	mult_rtnr_vtrafk=1.90	0,007%	0,008%	0,010%	12,6%
	mult_rtnr_vtrafk=2.30	0,007%	0,007%	0,008%	9,7%
1,2	mult_rtnr_vtrafk=2.70	0,007%	0,008%	0,009%	7,5%
	mult_rtnr_vtrafk=3.10	0,008%	0,008%	0,009%	8,1%
	mult_rtnr_vtrafk=0.30	0,028%	0,029%	0,029%	1,4%
	mult_rtnr_vtrafk=0.70	0,026%	0,027%	0,029%	3,9%
	mult_rtnr_vtrafk=1.10	0,026%	0,028%	0,029%	4,4%
	mult_rtnr_vtrafk=1.50	0,027%	0,029%	0,030%	4,3%
	mult_rtnr_vtrafk=1.90	0,028%	0,029%	0,030%	3,5%
1,3	mult_rtnr_vtrafk=2.30	0,028%	0,029%	0,031%	9,5%
	mult_rtnr_vtrafk=2.70	0,028%	0,029%	0,031%	5,5%
	mult_rtnr_vtrafk=3.10	0,030%	0,031%	0,032%	2,8%
	mult_rtnr_vtrafk=0.30	0,130%	0,134%	0,139%	3,6%
	mult_rtnr_vtrafk=0.70	0,119%	0,124%	0,128%	3,9%
	mult_rtnr_vtrafk=1.10	0,117%	0,120%	0,123%	2,5%
	mult_rtnr_vtrafk=1.50	0,123%	0,127%	0,130%	2,7%
1,4	mult_rtnr_vtrafk=1.90	0,129%	0,135%	0,140%	3,9%
	mult_rtnr_vtrafk=2.30	0,142%	0,146%	0,150%	2,7%
	mult_rtnr_vtrafk=2.70	0,152%	0,162%	0,171%	5,9%
	mult_rtnr_vtrafk=3.10	0,161%	0,168%	0,175%	4,2%
	mult_rtnr_vtrafk=0.30	0,835%	0,870%	0,906%	4,1%
	mult_rtnr_vtrafk=0.70	0,712%	0,758%	0,804%	6,1%
	mult_rtnr_vtrafk=1.10	0,697%	0,751%	0,805%	7,2%
1,5	mult_rtnr_vtrafk=1.50	0,731%	0,794%	0,856%	7,9%
	mult_rtnr_vtrafk=1.90	0,789%	0,829%	0,870%	4,9%
	mult_rtnr_vtrafk=2.30	0,829%	0,877%	0,925%	5,5%
	mult_rtnr_vtrafk=2.70	0,860%	0,945%	1,030%	9,0%
	mult_rtnr_vtrafk=3.10	0,960%	1,003%	1,047%	4,3%
	mult_rtnr_vtrafk=0.30	3,313%	3,400%	3,487%	2,6%
	mult_rtnr_vtrafk=0.70	2,936%	3,020%	3,104%	2,8%
1,6	mult_rtnr_vtrafk=1.10	2,896%	2,954%	3,013%	2,0%
	mult_rtnr_vtrafk=1.50	2,915%	2,992%	3,069%	2,6%
	mult_rtnr_vtrafk=1.90	2,962%	3,038%	3,114%	2,5%
	mult_rtnr_vtrafk=2.30	3,085%	3,134%	3,184%	1,6%
	mult_rtnr_vtrafk=2.70	3,110%	3,223%	3,337%	3,5%
	mult_rtnr_vtrafk=3.10	3,204%	3,272%	3,340%	2,1%
	mult_rtnr_vtrafk=0.30	7,056%	7,159%	7,263%	1,4%
1,7	mult_rtnr_vtrafk=0.70	6,186%	6,302%	6,418%	1,8%
	mult_rtnr_vtrafk=1.10	6,058%	6,173%	6,289%	1,9%
	mult_rtnr_vtrafk=1.50	6,090%	6,171%	6,253%	1,3%
	mult_rtnr_vtrafk=1.90	6,061%	6,196%	6,330%	2,2%
	mult_rtnr_vtrafk=2.30	6,113%	6,254%	6,394%	2,2%
	mult_rtnr_vtrafk=2.70	6,191%	6,321%	6,451%	2,1%
	mult_rtnr_vtrafk=3.10	6,278%	6,378%	6,478%	1,6%
1,8	mult_rtnr_vtrafk=0.30	10,392%	10,516%	10,640%	1,2%
	mult_rtnr_vtrafk=0.70	9,310%	9,428%	9,545%	1,2%
	mult_rtnr_vtrafk=1.10	9,292%	9,403%	9,514%	1,2%
	mult_rtnr_vtrafk=1.50	9,236%	9,357%	9,478%	1,3%
	mult_rtnr_vtrafk=1.90	9,275%	9,392%	9,509%	1,2%
	mult_rtnr_vtrafk=2.30	9,321%	9,448%	9,575%	1,3%
	mult_rtnr_vtrafk=2.70	9,372%	9,477%	9,581%	1,1%
1,9	mult_rtnr_vtrafk=3.10	9,410%	9,521%	9,632%	1,2%
	mult_rtnr_vtrafk=0.30	13,644%	13,746%	13,847%	0,7%
	mult_rtnr_vtrafk=0.70	12,308%	12,420%	12,531%	0,9%
	mult_rtnr_vtrafk=1.10	12,261%	12,377%	12,493%	0,9%
	mult_rtnr_vtrafk=1.50	12,239%	12,355%	12,472%	0,9%
	mult_rtnr_vtrafk=1.90	12,300%	12,383%	12,466%	0,7%
	mult_rtnr_vtrafk=2.30	12,340%	12,420%	12,500%	0,6%
2	mult_rtnr_vtrafk=2.70	12,321%	12,434%	12,546%	0,9%
	mult_rtnr_vtrafk=3.10	12,368%	12,466%	12,563%	0,8%
	mult_rtnr_vtrafk=0.30	16,781%	16,871%	16,960%	0,5%
	mult_rtnr_vtrafk=0.70	15,100%	15,172%	15,243%	0,5%
	mult_rtnr_vtrafk=1.10	15,082%	15,171%	15,259%	0,6%
	mult_rtnr_vtrafk=1.50	15,038%	15,134%	15,230%	0,6%
	mult_rtnr_vtrafk=1.90	15,120%	15,179%	15,239%	0,4%
2	mult_rtnr_vtrafk=2.30	15,093%	15,176%	15,258%	0,5%
	mult_rtnr_vtrafk=2.70	15,109%	15,190%	15,272%	0,5%
	mult_rtnr_vtrafk=3.10	15,139%	15,223%	15,306%	0,6%
	mult_rtnr_vtrafk=0.30	19,685%	19,789%	19,892%	0,5%
	mult_rtnr_vtrafk=0.70	17,650%	17,718%	17,787%	0,4%
	mult_rtnr_vtrafk=1.10	17,668%	17,730%	17,792%	0,3%
	mult_rtnr_vtrafk=1.50	17,654%	17,711%	17,767%	0,3%
2	mult_rtnr_vtrafk=1.90	17,675%	17,744%	17,813%	0,4%
	mult_rtnr_vtrafk=2.30	17,649%	17,716%	17,783%	0,4%
	mult_rtnr_vtrafk=2.70	17,696%	17,760%	17,825%	0,4%
	mult_rtnr_vtrafk=3.10	17,719%	17,776%	17,833%	0,3%

Tab. G-29 – Intervalos de confiança para a probabilidade de bloqueio global no RTNR versus o factor do  $VTtraf_k$  (sistema 3).

reserva	0
fluxos	4

alfa 5%

Average of valor				Limite Inferior	Média	Limite Superior	Conf (Rel)
carga	item	metodo	idcorrida				
1	TpblClas{0}	DAR	mult_dar_res	0,001%	0,001%	0,002%	32,8%
		DIRECTO	mult	0,326%	0,342%	0,358%	4,7%
	TpblClas{1}	DAR	mult_dar_res	0,002%	0,004%	0,006%	49,2%
		DIRECTO	mult	0,675%	0,694%	0,712%	2,7%
	TpblClas{2}	DAR	mult_dar_res	0,128%	0,152%	0,175%	15,4%
		DIRECTO	mult	1,141%	1,187%	1,233%	3,9%
TpblClas{3}	DAR	mult_dar_res	5,905%	6,175%	6,444%	4,4%	
	DIRECTO	mult	7,714%	8,053%	8,392%	4,2%	
1,1	TpblClas{0}	DAR	mult_dar_res	0,003%	0,004%	0,005%	23,2%
		DIRECTO	mult	0,527%	0,548%	0,569%	3,8%
	TpblClas{1}	DAR	mult_dar_res	0,022%	0,023%	0,025%	7,6%
		DIRECTO	mult	1,064%	1,082%	1,101%	1,7%
	TpblClas{2}	DAR	mult_dar_res	0,421%	0,455%	0,490%	7,5%
		DIRECTO	mult	1,567%	1,637%	1,707%	4,3%
TpblClas{3}	DAR	mult_dar_res	8,614%	8,739%	8,864%	1,4%	
	DIRECTO	mult	9,873%	10,436%	10,998%	5,4%	
1,2	TpblClas{0}	DAR	mult_dar_res	0,033%	0,035%	0,038%	6,6%
		DIRECTO	mult	1,031%	1,050%	1,069%	1,8%
	TpblClas{1}	DAR	mult_dar_res	0,159%	0,176%	0,192%	9,3%
		DIRECTO	mult	2,047%	2,071%	2,096%	1,2%
	TpblClas{2}	DAR	mult_dar_res	0,971%	1,032%	1,094%	6,0%
		DIRECTO	mult	2,072%	2,212%	2,353%	6,4%
TpblClas{3}	DAR	mult_dar_res	11,609%	12,095%	12,580%	4,0%	
	DIRECTO	mult	12,278%	12,641%	13,004%	2,9%	
1,3	TpblClas{0}	DAR	mult_dar_res	0,222%	0,230%	0,239%	3,8%
		DIRECTO	mult	1,935%	1,967%	1,999%	1,6%
	TpblClas{1}	DAR	mult_dar_res	0,945%	0,986%	1,028%	4,2%
		DIRECTO	mult	3,762%	3,823%	3,884%	1,6%
	TpblClas{2}	DAR	mult_dar_res	2,213%	2,347%	2,481%	5,7%
		DIRECTO	mult	2,914%	3,020%	3,127%	3,5%
TpblClas{3}	DAR	mult_dar_res	15,363%	15,761%	16,160%	2,5%	
	DIRECTO	mult	14,158%	14,668%	15,179%	3,5%	
1,4	TpblClas{0}	DAR	mult_dar_res	0,749%	0,780%	0,810%	3,9%
		DIRECTO	mult	3,141%	3,193%	3,245%	1,6%
	TpblClas{1}	DAR	mult_dar_res	2,859%	2,944%	3,028%	2,9%
		DIRECTO	mult	6,038%	6,144%	6,249%	1,7%
	TpblClas{2}	DAR	mult_dar_res	4,446%	4,593%	4,741%	3,2%
		DIRECTO	mult	3,984%	4,089%	4,194%	2,6%
TpblClas{3}	DAR	mult_dar_res	20,056%	20,324%	20,593%	1,3%	
	DIRECTO	mult	17,323%	17,724%	17,724%	1,1%	
1,5	TpblClas{0}	DAR	mult_dar_res	1,759%	1,785%	1,810%	1,4%
		DIRECTO	mult	4,596%	4,657%	4,719%	1,3%
	TpblClas{1}	DAR	mult_dar_res	5,862%	5,939%	6,016%	1,3%
		DIRECTO	mult	8,741%	8,826%	8,912%	1,0%
	TpblClas{2}	DAR	mult_dar_res	8,023%	8,208%	8,392%	2,3%
		DIRECTO	mult	5,632%	5,826%	6,021%	3,3%
TpblClas{3}	DAR	mult_dar_res	28,582%	28,817%	29,051%	0,8%	
	DIRECTO	mult	22,600%	22,989%	23,378%	1,7%	
1,6	TpblClas{0}	DAR	mult_dar_res	3,176%	3,227%	3,278%	1,6%
		DIRECTO	mult	6,211%	6,264%	6,317%	0,8%
	TpblClas{1}	DAR	mult_dar_res	9,471%	9,597%	9,722%	1,3%
		DIRECTO	mult	11,588%	11,686%	11,784%	0,8%
	TpblClas{2}	DAR	mult_dar_res	12,730%	13,035%	13,340%	2,3%
		DIRECTO	mult	8,665%	8,883%	9,100%	2,4%
TpblClas{3}	DAR	mult_dar_res	39,443%	40,447%	41,451%	2,5%	
	DIRECTO	mult	31,664%	32,404%	33,145%	2,3%	
1,7	TpblClas{0}	DAR	mult_dar_res	4,914%	4,962%	5,009%	1,0%
		DIRECTO	mult	7,893%	7,945%	7,997%	0,7%
	TpblClas{1}	DAR	mult_dar_res	13,297%	13,396%	13,496%	0,7%
		DIRECTO	mult	14,513%	14,598%	14,682%	0,6%
	TpblClas{2}	DAR	mult_dar_res	18,265%	18,598%	18,932%	1,8%
		DIRECTO	mult	12,858%	13,075%	13,291%	1,7%
TpblClas{3}	DAR	mult_dar_res	52,307%	53,040%	53,773%	1,4%	
	DIRECTO	mult	43,084%	44,059%	45,033%	2,2%	

Tab. G-30 – Intervalos de confiança para a probabilidade de bloqueio global por classe para o método DAR e DIRECTO (sistema 3).

metodo	DAR
reserva	0
fluxos	4

alfa 5%

Average of valor			Limite Inferior	Média	Limite Superior	Conf (Rel)	
carga	item	idcorrida					
1	TpbiClas{0}	mult_dar_res	0,001%	0,001%	0,002%	32,8%	
		mult_dar_res_b	0,002%	0,003%	0,004%	27,8%	
	TpbiClas{1}	mult_dar_res	0,002%	0,004%	0,006%	49,2%	
		mult_dar_res_b	0,013%	0,018%	0,023%	26,3%	
	TpbiClas{2}	mult_dar_res	0,128%	0,152%	0,175%	15,4%	
		mult_dar_res_b	0,035%	0,055%	0,074%	35,5%	
	TpbiClas{3}	mult_dar_res	5,905%	6,175%	6,444%	4,4%	
		mult_dar_res_b	2,266%	2,544%	2,821%	10,9%	
	1,1	TpbiClas{0}	mult_dar_res	0,003%	0,004%	0,005%	23,2%
			mult_dar_res_b	0,009%	0,011%	0,013%	17,5%
TpbiClas{1}		mult_dar_res	0,022%	0,023%	0,025%	7,6%	
		mult_dar_res_b	0,054%	0,058%	0,063%	7,8%	
TpbiClas{2}		mult_dar_res	0,421%	0,455%	0,490%	7,5%	
		mult_dar_res_b	0,145%	0,180%	0,215%	19,4%	
TpbiClas{3}		mult_dar_res	8,614%	8,739%	8,864%	1,4%	
		mult_dar_res_b	4,757%	5,031%	5,304%	5,4%	
1,2		TpbiClas{0}	mult_dar_res	0,033%	0,035%	0,038%	6,6%
			mult_dar_res_b	0,056%	0,059%	0,061%	4,3%
	TpbiClas{1}	mult_dar_res	0,159%	0,176%	0,192%	9,3%	
		mult_dar_res_b	0,262%	0,276%	0,290%	5,1%	
	TpbiClas{2}	mult_dar_res	0,971%	1,032%	1,094%	6,0%	
		mult_dar_res_b	0,416%	0,436%	0,455%	4,4%	
	TpbiClas{3}	mult_dar_res	11,609%	12,095%	12,580%	4,0%	
		mult_dar_res_b	8,176%	8,740%	9,304%	6,5%	
	1,3	TpbiClas{0}	mult_dar_res	0,222%	0,230%	0,239%	3,8%
			mult_dar_res_b	0,290%	0,301%	0,312%	3,6%
TpbiClas{1}		mult_dar_res	0,945%	0,986%	1,028%	4,2%	
		mult_dar_res_b	1,171%	1,217%	1,264%	3,8%	
TpbiClas{2}		mult_dar_res	2,213%	2,347%	2,481%	5,7%	
		mult_dar_res_b	1,018%	1,106%	1,193%	7,9%	
TpbiClas{3}		mult_dar_res	15,363%	15,761%	16,160%	2,5%	
		mult_dar_res_b	12,729%	13,141%	13,553%	3,1%	
1,4		TpbiClas{0}	mult_dar_res	0,749%	0,780%	0,810%	3,9%
			mult_dar_res_b	0,903%	0,934%	0,965%	3,3%
	TpbiClas{1}	mult_dar_res	2,859%	2,944%	3,028%	2,9%	
		mult_dar_res_b	3,270%	3,366%	3,463%	2,9%	
	TpbiClas{2}	mult_dar_res	4,446%	4,593%	4,741%	3,2%	
		mult_dar_res_b	2,532%	2,651%	2,771%	4,5%	
	TpbiClas{3}	mult_dar_res	20,056%	20,324%	20,593%	1,3%	
		mult_dar_res_b	18,761%	19,071%	19,382%	1,6%	
	1,5	TpbiClas{0}	mult_dar_res	1,759%	1,785%	1,810%	1,4%
			mult_dar_res_b	2,009%	2,049%	2,089%	2,0%
TpbiClas{1}		mult_dar_res	5,862%	5,939%	6,016%	1,3%	
		mult_dar_res_b	6,514%	6,603%	6,692%	1,3%	
TpbiClas{2}		mult_dar_res	8,023%	8,208%	8,392%	2,3%	
		mult_dar_res_b	5,403%	5,469%	5,534%	1,2%	
TpbiClas{3}		mult_dar_res	28,582%	28,817%	29,051%	0,8%	
		mult_dar_res_b	27,432%	28,039%	28,645%	2,2%	
1,6		TpbiClas{0}	mult_dar_res	3,176%	3,227%	3,278%	1,6%
			mult_dar_res_b	3,568%	3,615%	3,663%	1,3%
	TpbiClas{1}	mult_dar_res	9,471%	9,597%	9,722%	1,3%	
		mult_dar_res_b	10,320%	10,410%	10,501%	0,9%	
	TpbiClas{2}	mult_dar_res	12,730%	13,035%	13,340%	2,3%	
		mult_dar_res_b	9,568%	9,885%	10,202%	3,2%	
	TpbiClas{3}	mult_dar_res	39,443%	40,447%	41,451%	2,5%	
		mult_dar_res_b	39,680%	40,453%	41,227%	1,9%	
	1,7	TpbiClas{0}	mult_dar_res	4,914%	4,962%	5,009%	1,0%
			mult_dar_res_b	5,383%	5,431%	5,478%	0,9%
TpbiClas{1}		mult_dar_res	13,297%	13,396%	13,496%	0,7%	
		mult_dar_res_b	14,125%	14,226%	14,327%	0,7%	
TpbiClas{2}		mult_dar_res	18,265%	18,598%	18,932%	1,8%	
		mult_dar_res_b	14,927%	15,269%	15,612%	2,2%	
TpbiClas{3}		mult_dar_res	52,307%	53,040%	53,773%	1,4%	
		mult_dar_res_b	52,370%	53,169%	53,967%	1,5%	

Tab. G-31 – Intervalos de confiança para a probabilidade de bloqueio global por classe para o método DAR, para duas diferentes escolhas dos valores da reserva de circuitos fixa (sistema 3).

reserva	0
idcorrida	mult

alfa

5%

Average of valor			Limite Inferior	Média	Limite Superior	Conf (Rel)
carga	item	metodo				
1	TpblClas{0}	DIRECTO	0,326%	0,342%	0,358%	4,7%
		RTNR	0,000%	0,000%	0,000%	0,0%
	TpblClas{1}	DIRECTO	0,675%	0,694%	0,712%	2,7%
		RTNR	0,000%	0,000%	0,000%	0,0%
	TpblClas{2}	DIRECTO	1,141%	1,187%	1,233%	3,9%
		RTNR	0,000%	0,066%	0,244%	267,7%
TpblClas{3}	DIRECTO	7,714%	8,053%	8,392%	4,2%	
	RTNR	0,000%	1,312%	3,724%	183,8%	
1,1	TpblClas{0}	DIRECTO	0,527%	0,548%	0,569%	3,8%
		RTNR	0,000%	0,000%	0,000%	194,2%
	TpblClas{1}	DIRECTO	1,064%	1,082%	1,101%	1,7%
		RTNR	0,000%	0,000%	0,000%	0,0%
	TpblClas{2}	DIRECTO	1,567%	1,637%	1,707%	4,3%
		RTNR	0,011%	0,019%	0,027%	44,1%
TpblClas{3}	DIRECTO	9,873%	10,436%	10,998%	5,4%	
	RTNR	1,716%	2,024%	2,333%	15,2%	
1,2	TpblClas{0}	DIRECTO	1,031%	1,050%	1,069%	1,8%
		RTNR	0,002%	0,003%	0,004%	35,6%
	TpblClas{1}	DIRECTO	2,047%	2,071%	2,096%	1,2%
		RTNR	0,004%	0,006%	0,008%	33,0%
	TpblClas{2}	DIRECTO	2,072%	2,212%	2,353%	6,4%
		RTNR	0,079%	0,094%	0,108%	15,5%
TpblClas{3}	DIRECTO	12,278%	12,641%	13,004%	2,9%	
	RTNR	5,921%	6,128%	6,335%	3,4%	
1,3	TpblClas{0}	DIRECTO	1,935%	1,967%	1,999%	1,6%
		RTNR	0,033%	0,055%	0,076%	38,8%
	TpblClas{1}	DIRECTO	3,762%	3,823%	3,884%	1,6%
		RTNR	0,050%	0,089%	0,128%	44,1%
	TpblClas{2}	DIRECTO	2,914%	3,020%	3,127%	3,5%
		RTNR	0,617%	0,665%	0,712%	7,1%
TpblClas{3}	DIRECTO	14,158%	14,668%	15,179%	3,5%	
	RTNR	11,279%	13,329%	15,380%	15,4%	
1,4	TpblClas{0}	DIRECTO	3,141%	3,193%	3,245%	1,6%
		RTNR	0,509%	0,537%	0,566%	5,3%
	TpblClas{1}	DIRECTO	6,038%	6,144%	6,249%	1,7%
		RTNR	0,648%	0,703%	0,758%	7,9%
	TpblClas{2}	DIRECTO	3,984%	4,089%	4,194%	2,6%
		RTNR	3,705%	3,938%	4,171%	5,9%
TpblClas{3}	DIRECTO	17,323%	17,524%	17,724%	1,1%	
	RTNR	22,222%	22,845%	23,468%	2,7%	
1,5	TpblClas{0}	DIRECTO	4,596%	4,657%	4,719%	1,3%
		RTNR	2,642%	2,718%	2,795%	2,8%
	TpblClas{1}	DIRECTO	8,741%	8,826%	8,912%	1,0%
		RTNR	1,907%	1,993%	2,079%	4,3%
	TpblClas{2}	DIRECTO	5,632%	5,826%	6,021%	3,3%
		RTNR	10,557%	10,890%	11,224%	3,1%
TpblClas{3}	DIRECTO	22,600%	22,989%	23,378%	1,7%	
	RTNR	35,033%	35,904%	36,776%	2,4%	
1,6	TpblClas{0}	DIRECTO	6,211%	6,264%	6,317%	0,8%
		RTNR	6,056%	6,186%	6,316%	2,1%
	TpblClas{1}	DIRECTO	11,588%	11,686%	11,784%	0,8%
		RTNR	3,493%	3,614%	3,735%	3,3%
	TpblClas{2}	DIRECTO	8,665%	8,883%	9,100%	2,4%
		RTNR	15,641%	15,770%	15,899%	0,8%
TpblClas{3}	DIRECTO	31,664%	32,404%	33,145%	2,3%	
	RTNR	43,686%	43,925%	44,164%	0,5%	
1,7	TpblClas{0}	DIRECTO	7,893%	7,945%	7,997%	0,7%
		RTNR	9,306%	9,447%	9,587%	1,5%
	TpblClas{1}	DIRECTO	14,513%	14,598%	14,682%	0,6%
		RTNR	6,189%	6,323%	6,456%	2,1%
	TpblClas{2}	DIRECTO	12,858%	13,075%	13,291%	1,7%
		RTNR	19,878%	20,126%	20,374%	1,2%
TpblClas{3}	DIRECTO	43,084%	44,059%	45,033%	2,2%	
	RTNR	50,633%	51,169%	51,704%	1,0%	
1,8	TpblClas{0}	DIRECTO	9,642%	9,683%	9,724%	0,4%
		RTNR	12,209%	12,304%	12,399%	0,8%
	TpblClas{1}	DIRECTO	17,467%	17,537%	17,608%	0,4%
		RTNR	9,496%	9,639%	9,781%	1,5%
	TpblClas{2}	DIRECTO	17,800%	18,085%	18,370%	1,6%
		RTNR	23,328%	23,588%	23,848%	1,1%
TpblClas{3}	DIRECTO	54,827%	55,618%	56,408%	1,4%	
	RTNR	58,055%	58,711%	59,368%	1,1%	

Tab. G-32 – Intervalos de confiança para a probabilidade de bloqueio global por classe para o método RTNR e DIRECTO (sistema 3).

metodo	RTNR
reserva	0
fluxos	4

alfa 5%

Average of valor			Limite Inferior	Média	Limite Superior	Conf (Rel)
carga	item	idcorrida				
1	TpblClas{0}	mult	0,000%	0,000%	0,000%	0,0%
		mult_a	0,000%	0,002%	0,006%	274,7%
	TpblClas{1}	mult	0,000%	0,000%	0,000%	0,0%
		mult_a	0,000%	0,001%	0,006%	277,6%
	TpblClas{2}	mult	0,000%	0,066%	0,244%	267,7%
		mult_a	0,000%	0,002%	0,006%	215,0%
	TpblClas{3}	mult	0,000%	1,312%	3,724%	183,8%
		mult_a	0,329%	0,432%	0,536%	24,0%
1,1	TpblClas{0}	mult	0,000%	0,000%	0,000%	194,2%
		mult_a	0,000%	0,000%	0,000%	166,5%
	TpblClas{1}	mult	0,000%	0,000%	0,000%	0,0%
		mult_a	0,000%	0,000%	0,000%	113,3%
	TpblClas{2}	mult	0,011%	0,019%	0,027%	44,1%
		mult_a	0,009%	0,015%	0,021%	38,9%
	TpblClas{3}	mult	1,716%	2,024%	2,333%	15,2%
		mult_a	1,899%	2,115%	2,330%	10,2%
1,2	TpblClas{0}	mult	0,002%	0,003%	0,004%	35,6%
		mult_a	0,001%	0,002%	0,003%	47,2%
	TpblClas{1}	mult	0,004%	0,006%	0,008%	33,0%
		mult_a	0,002%	0,005%	0,007%	54,4%
	TpblClas{2}	mult	0,079%	0,094%	0,108%	15,5%
		mult_a	0,067%	0,074%	0,082%	10,3%
	TpblClas{3}	mult	5,921%	6,128%	6,335%	3,4%
		mult_a	5,794%	6,440%	7,087%	10,0%
1,3	TpblClas{0}	mult	0,033%	0,055%	0,076%	38,8%
		mult_a	0,047%	0,050%	0,054%	6,8%
	TpblClas{1}	mult	0,050%	0,089%	0,128%	44,1%
		mult_a	0,068%	0,081%	0,095%	16,7%
	TpblClas{2}	mult	0,617%	0,665%	0,712%	7,1%
		mult_a	0,584%	0,669%	0,755%	12,8%
	TpblClas{3}	mult	11,279%	13,329%	15,380%	15,4%
		mult_a	12,363%	12,527%	12,692%	1,3%
1,4	TpblClas{0}	mult	0,509%	0,537%	0,566%	5,3%
		mult_a	0,609%	0,666%	0,723%	8,5%
	TpblClas{1}	mult	0,648%	0,703%	0,758%	7,9%
		mult_a	0,684%	0,729%	0,774%	6,2%
	TpblClas{2}	mult	3,705%	3,938%	4,171%	5,9%
		mult_a	3,369%	4,307%	5,245%	21,8%
	TpblClas{3}	mult	22,222%	22,845%	23,468%	2,7%
		mult_a	22,440%	23,007%	23,575%	2,5%
1,5	TpblClas{0}	mult	2,642%	2,718%	2,795%	2,8%
		mult_a	4,815%	4,891%	4,968%	1,6%
	TpblClas{1}	mult	1,907%	1,993%	2,079%	4,3%
		mult_a	2,934%	3,163%	3,392%	7,2%
	TpblClas{2}	mult	10,557%	10,890%	11,224%	3,1%
		mult_a	8,969%	9,263%	9,556%	3,2%
	TpblClas{3}	mult	35,033%	35,904%	36,776%	2,4%
		mult_a	32,418%	32,780%	33,143%	1,1%
1,6	TpblClas{0}	mult	6,056%	6,186%	6,316%	2,1%
		mult_a	8,209%	8,302%	8,395%	1,1%
	TpblClas{1}	mult	3,493%	3,614%	3,735%	3,3%
		mult_a	5,330%	5,528%	5,727%	3,6%
	TpblClas{2}	mult	15,641%	15,770%	15,899%	0,8%
		mult_a	13,650%	13,872%	14,094%	1,6%
	TpblClas{3}	mult	43,686%	43,925%	44,164%	0,5%
		mult_a	38,712%	39,315%	39,917%	1,5%
1,7	TpblClas{0}	mult	9,306%	9,447%	9,587%	1,5%
		mult_a	11,269%	11,335%	11,401%	0,6%
	TpblClas{1}	mult	6,189%	6,323%	6,456%	2,1%
		mult_a	7,752%	7,926%	8,101%	2,2%
	TpblClas{2}	mult	19,878%	20,126%	20,374%	1,2%
		mult_a	17,610%	18,030%	18,451%	2,3%
	TpblClas{3}	mult	50,633%	51,169%	51,704%	1,0%
		mult_a	46,064%	46,558%	47,052%	1,1%
1,8	TpblClas{0}	mult	12,209%	12,304%	12,399%	0,8%
		mult_a	13,988%	14,091%	14,194%	0,7%
	TpblClas{1}	mult	9,496%	9,639%	9,781%	1,5%
		mult_a	10,774%	10,865%	10,955%	0,8%
	TpblClas{2}	mult	23,328%	23,588%	23,848%	1,1%
		mult_a	21,606%	21,875%	22,144%	1,2%
	TpblClas{3}	mult	58,055%	58,711%	59,368%	1,1%
		mult_a	53,195%	53,778%	54,361%	1,1%

Tab. G-33 – Intervalos de confiança para a probabilidade de bloqueio global por classe para o RTNR, para duas diferentes escolhas dos valores da largura de banda mínima garantida (sistema 3).

reserva	0
fluxos	4

alfa 5%

Average of valor carga				Limite Inferior	Média	Limite Superior	Conf (Rel)
item	metodo	idcorrida					
1	TpbIClas{0}	DAR	mult_dar_res	0,001%	0,001%	0,002%	32,8%
		RTNR	mult	0,000%	0,000%	0,000%	0,0%
	TpbIClas{1}	DAR	mult_dar_res	0,002%	0,004%	0,006%	49,2%
		RTNR	mult	0,000%	0,000%	0,000%	0,0%
	TpbIClas{2}	DAR	mult_dar_res	0,128%	0,152%	0,175%	15,4%
		RTNR	mult	0,000%	0,066%	0,244%	267,7%
TpbIClas{3}	DAR	mult_dar_res	5,905%	6,175%	6,444%	4,4%	
	RTNR	mult	0,000%	1,312%	3,724%	183,8%	
1.1	TpbIClas{0}	DAR	mult_dar_res	0,003%	0,004%	0,005%	23,2%
		RTNR	mult	0,000%	0,000%	0,000%	194,2%
	TpbIClas{1}	DAR	mult_dar_res	0,022%	0,023%	0,025%	7,6%
		RTNR	mult	0,000%	0,000%	0,000%	0,0%
	TpbIClas{2}	DAR	mult_dar_res	0,421%	0,455%	0,490%	7,5%
		RTNR	mult	0,011%	0,019%	0,027%	44,1%
TpbIClas{3}	DAR	mult_dar_res	8,614%	8,739%	8,864%	1,4%	
	RTNR	mult	1,716%	2,024%	2,333%	15,2%	
1.2	TpbIClas{0}	DAR	mult_dar_res	0,033%	0,035%	0,038%	6,6%
		RTNR	mult	0,002%	0,003%	0,004%	35,6%
	TpbIClas{1}	DAR	mult_dar_res	0,159%	0,176%	0,192%	9,3%
		RTNR	mult	0,004%	0,006%	0,008%	33,0%
	TpbIClas{2}	DAR	mult_dar_res	0,971%	1,032%	1,094%	6,0%
		RTNR	mult	0,079%	0,094%	0,108%	15,5%
TpbIClas{3}	DAR	mult_dar_res	11,609%	12,095%	12,580%	4,0%	
	RTNR	mult	5,921%	6,128%	6,335%	3,4%	
1.3	TpbIClas{0}	DAR	mult_dar_res	0,222%	0,230%	0,239%	3,8%
		RTNR	mult	0,033%	0,055%	0,076%	38,8%
	TpbIClas{1}	DAR	mult_dar_res	0,945%	0,986%	1,028%	4,2%
		RTNR	mult	0,050%	0,089%	0,128%	44,1%
	TpbIClas{2}	DAR	mult_dar_res	2,213%	2,347%	2,481%	5,7%
		RTNR	mult	0,617%	0,665%	0,712%	7,1%
TpbIClas{3}	DAR	mult_dar_res	15,363%	15,761%	16,160%	2,5%	
	RTNR	mult	11,279%	13,329%	15,380%	15,4%	
1.4	TpbIClas{0}	DAR	mult_dar_res	0,749%	0,780%	0,810%	3,9%
		RTNR	mult	0,509%	0,537%	0,566%	5,3%
	TpbIClas{1}	DAR	mult_dar_res	2,859%	2,944%	3,028%	2,9%
		RTNR	mult	0,648%	0,703%	0,758%	7,9%
	TpbIClas{2}	DAR	mult_dar_res	4,446%	4,593%	4,741%	3,2%
		RTNR	mult	3,705%	3,938%	4,171%	5,9%
TpbIClas{3}	DAR	mult_dar_res	20,056%	20,324%	20,593%	1,3%	
	RTNR	mult	22,222%	22,845%	23,468%	2,7%	
1.5	TpbIClas{0}	DAR	mult_dar_res	1,759%	1,785%	1,810%	1,4%
		RTNR	mult	2,642%	2,718%	2,795%	2,8%
	TpbIClas{1}	DAR	mult_dar_res	5,862%	5,939%	6,016%	1,3%
		RTNR	mult	1,907%	1,993%	2,079%	4,3%
	TpbIClas{2}	DAR	mult_dar_res	8,023%	8,208%	8,392%	2,3%
		RTNR	mult	10,567%	10,890%	11,224%	3,1%
TpbIClas{3}	DAR	mult_dar_res	28,582%	28,817%	29,051%	0,8%	
	RTNR	mult	35,033%	35,904%	36,776%	2,4%	
1.6	TpbIClas{0}	DAR	mult_dar_res	3,176%	3,227%	3,278%	1,6%
		RTNR	mult	6,056%	6,186%	6,316%	2,1%
	TpbIClas{1}	DAR	mult_dar_res	9,471%	9,597%	9,722%	1,3%
		RTNR	mult	3,493%	3,614%	3,735%	3,3%
	TpbIClas{2}	DAR	mult_dar_res	12,730%	13,035%	13,340%	2,3%
		RTNR	mult	15,641%	15,770%	15,899%	0,8%
TpbIClas{3}	DAR	mult_dar_res	39,443%	40,447%	41,451%	2,5%	
	RTNR	mult	43,686%	43,925%	44,164%	0,5%	
1.7	TpbIClas{0}	DAR	mult_dar_res	4,914%	4,962%	5,009%	1,0%
		RTNR	mult	9,306%	9,447%	9,587%	1,5%
	TpbIClas{1}	DAR	mult_dar_res	13,297%	13,396%	13,496%	0,7%
		RTNR	mult	6,189%	6,323%	6,456%	2,1%
	TpbIClas{2}	DAR	mult_dar_res	18,265%	18,598%	18,932%	1,8%
		RTNR	mult	19,878%	20,126%	20,374%	1,2%
TpbIClas{3}	DAR	mult_dar_res	52,307%	53,040%	53,773%	1,4%	
	RTNR	mult	50,633%	51,169%	51,704%	1,0%	
1.8	TpbIClas{0}	DAR	mult_dar_res	6,785%	6,856%	6,926%	1,0%
		RTNR	mult	12,209%	12,304%	12,399%	0,8%
	TpbIClas{1}	DAR	mult_dar_res	16,959%	17,087%	17,215%	0,7%
		RTNR	mult	9,496%	9,639%	9,781%	1,5%
	TpbIClas{2}	DAR	mult_dar_res	24,321%	24,654%	24,987%	1,4%
		RTNR	mult	23,328%	23,588%	23,848%	1,1%
TpbIClas{3}	DAR	mult_dar_res	64,015%	64,522%	65,030%	0,8%	
	RTNR	mult	58,055%	58,711%	59,368%	1,1%	
1.9	TpbIClas{0}	DAR	mult_dar_res	8,810%	8,848%	8,886%	0,4%
		RTNR	mult	14,840%	14,892%	14,945%	0,4%
	TpbIClas{1}	DAR	mult_dar_res	20,498%	20,632%	20,766%	0,6%
		RTNR	mult	13,048%	13,184%	13,320%	1,0%
	TpbIClas{2}	DAR	mult_dar_res	30,016%	30,314%	30,611%	1,0%
		RTNR	mult	26,656%	27,002%	27,348%	1,3%
TpbIClas{3}	DAR	mult_dar_res	72,975%	73,587%	74,198%	0,8%	
	RTNR	mult	64,648%	65,230%	65,811%	0,9%	
2	TpbIClas{0}	DAR	mult_dar_res	10,813%	10,851%	10,888%	0,3%
		RTNR	mult	17,206%	17,254%	17,302%	0,3%
	TpbIClas{1}	DAR	mult_dar_res	23,818%	23,899%	23,981%	0,3%
		RTNR	mult	16,606%	16,684%	16,762%	0,5%
	TpbIClas{2}	DAR	mult_dar_res	35,711%	35,954%	36,196%	0,7%
		RTNR	mult	29,852%	29,954%	30,056%	0,3%
TpbIClas{3}	DAR	mult_dar_res	80,881%	81,226%	81,570%	0,4%	
	RTNR	mult	70,475%	71,111%	71,748%	0,9%	

Tab. G-34 – Intervalos de confiança para a probabilidade de bloqueio global por classe para o método DAR e RTNR (sistema 3).

metodo	DAR
reserva	0
carga	1,4
idcorrida	mult_dar_res

alfa

5%

Average of valor			Limite Inferior	Média	Limite Superior	Conf (Rel)
origem	destino	item				
1	1	Ppbl{0}	0,000%	0,000%	0,000%	0,0%
		Ppbl{1}	0,000%	0,000%	0,000%	0,0%
	2	Ppbl{0}	1,304%	1,429%	1,554%	8,8%
		Ppbl{1}	5,293%	5,651%	6,008%	6,3%
	3	Ppbl{0}	0,408%	0,487%	0,565%	16,1%
		Ppbl{1}	1,716%	2,241%	2,767%	23,5%
	4	Ppbl{0}	0,022%	0,028%	0,033%	19,9%
		Ppbl{1}	0,078%	0,089%	0,100%	12,0%
	5	Ppbl{0}	0,121%	0,170%	0,219%	28,7%
		Ppbl{1}	0,493%	0,615%	0,737%	19,8%
	6	Ppbl{0}	0,278%	0,349%	0,420%	20,4%
		Ppbl{1}	1,433%	1,586%	1,738%	9,6%
2	1	Ppbl{0}	1,303%	1,407%	1,511%	7,4%
		Ppbl{1}	5,813%	5,955%	6,096%	2,4%
	2	Ppbl{0}	0,000%	0,000%	0,000%	0,0%
		Ppbl{1}	0,000%	0,000%	0,000%	0,0%
	3	Ppbl{0}	0,000%	0,002%	0,004%	170,0%
		Ppbl{1}	0,000%	0,004%	0,015%	277,6%
	4	Ppbl{0}	3,615%	3,733%	3,851%	3,2%
		Ppbl{1}	12,027%	12,519%	13,011%	3,9%
	5	Ppbl{0}	0,399%	0,455%	0,510%	12,2%
		Ppbl{1}	2,438%	2,726%	3,015%	10,6%
	6	Ppbl{0}	1,689%	1,755%	1,820%	3,7%
		Ppbl{1}	5,800%	6,077%	6,353%	4,5%
3	1	Ppbl{0}	0,410%	0,557%	0,704%	26,4%
		Ppbl{1}	1,351%	1,866%	2,381%	27,6%
	2	Ppbl{0}	0,000%	0,002%	0,009%	277,6%
		Ppbl{1}	0,000%	0,016%	0,037%	130,5%
	3	Ppbl{0}	0,000%	0,000%	0,000%	0,0%
		Ppbl{1}	0,000%	0,000%	0,000%	0,0%
	4	Ppbl{0}	3,391%	3,782%	4,172%	10,3%
		Ppbl{1}	15,795%	16,965%	18,134%	6,9%
	5	Ppbl{0}	0,000%	0,003%	0,011%	277,6%
		Ppbl{1}	0,000%	0,058%	0,158%	170,0%
	6	Ppbl{0}	0,451%	0,540%	0,629%	16,5%
		Ppbl{1}	3,188%	3,739%	4,290%	14,7%
4	1	Ppbl{0}	0,019%	0,026%	0,032%	26,1%
		Ppbl{1}	0,072%	0,096%	0,119%	24,7%
	2	Ppbl{0}	3,588%	3,732%	3,875%	3,8%
		Ppbl{1}	12,124%	12,516%	12,909%	3,1%
	3	Ppbl{0}	3,462%	3,898%	4,333%	11,2%
		Ppbl{1}	15,219%	16,916%	18,614%	10,0%
	4	Ppbl{0}	0,000%	0,000%	0,000%	0,0%
		Ppbl{1}	0,000%	0,000%	0,000%	0,0%
	5	Ppbl{0}	0,074%	0,094%	0,115%	21,9%
		Ppbl{1}	0,292%	0,346%	0,399%	15,5%
	6	Ppbl{0}	0,685%	0,724%	0,764%	5,5%
		Ppbl{1}	2,797%	3,163%	3,530%	11,6%
5	1	Ppbl{0}	0,118%	0,161%	0,205%	27,0%
		Ppbl{1}	0,498%	0,727%	0,956%	31,5%
	2	Ppbl{0}	0,384%	0,429%	0,473%	10,5%
		Ppbl{1}	2,535%	2,742%	2,949%	7,6%
	3	Ppbl{0}	0,004%	0,012%	0,020%	69,4%
		Ppbl{1}	0,000%	0,058%	0,133%	128,8%
	4	Ppbl{0}	0,077%	0,098%	0,119%	21,4%
		Ppbl{1}	0,284%	0,348%	0,413%	18,6%
	5	Ppbl{0}	0,000%	0,000%	0,000%	0,0%
		Ppbl{1}	0,000%	0,000%	0,000%	0,0%
	6	Ppbl{0}	0,000%	0,000%	0,000%	0,0%
		Ppbl{1}	0,000%	0,000%	0,000%	0,0%
6	1	Ppbl{0}	0,242%	0,324%	0,405%	25,2%
		Ppbl{1}	1,060%	1,274%	1,487%	16,7%
	2	Ppbl{0}	1,669%	1,757%	1,844%	5,0%
		Ppbl{1}	5,759%	6,309%	6,858%	8,7%
	3	Ppbl{0}	0,423%	0,593%	0,764%	28,7%
		Ppbl{1}	3,268%	3,734%	4,199%	12,5%
	4	Ppbl{0}	0,586%	0,693%	0,799%	15,4%
		Ppbl{1}	3,071%	3,335%	3,599%	7,9%
	5	Ppbl{0}	0,000%	0,000%	0,000%	0,0%
		Ppbl{1}	0,000%	0,000%	0,000%	0,0%
	6	Ppbl{0}	0,000%	0,000%	0,000%	0,0%
		Ppbl{1}	0,000%	0,000%	0,000%	0,0%

Tab. G-35 – Intervalos de confiança para a probabilidade de bloqueio ponto a ponto por classe, para as classes 0 e 1, para o método DAR (sistema 3).

metodo	RTNR
reserva	0
carga	1,4
idcorrida	mult

alfa 5%

Average of valor			Limite Inferior	Média	Limite Superior	Conf (Rel)
origem	destino	item				
1	1	Ppbl{0}	0,000%	0,000%	0,000%	0,0%
		Ppbl{1}	0,000%	0,000%	0,000%	0,0%
	2	Ppbl{0}	2,162%	2,424%	2,685%	10,8%
		Ppbl{1}	1,894%	2,057%	2,220%	7,9%
	3	Ppbl{0}	1,828%	2,161%	2,493%	15,4%
		Ppbl{1}	5,258%	6,208%	7,157%	15,3%
	4	Ppbl{0}	0,060%	0,074%	0,088%	18,9%
		Ppbl{1}	0,096%	0,127%	0,158%	24,5%
	5	Ppbl{0}	0,399%	0,481%	0,563%	17,1%
		Ppbl{1}	0,610%	0,746%	0,882%	18,3%
	6	Ppbl{0}	0,723%	0,858%	0,993%	15,7%
		Ppbl{1}	0,900%	1,141%	1,383%	21,1%
2	1	Ppbl{0}	2,168%	2,250%	2,331%	3,6%
		Ppbl{1}	1,745%	2,151%	2,557%	18,9%
	2	Ppbl{0}	0,000%	0,000%	0,000%	0,0%
		Ppbl{1}	0,000%	0,000%	0,000%	0,0%
	3	Ppbl{0}	0,565%	0,594%	0,623%	4,9%
		Ppbl{1}	0,793%	0,975%	1,157%	18,7%
	4	Ppbl{0}	1,605%	1,685%	1,764%	4,7%
		Ppbl{1}	1,816%	1,980%	2,144%	8,3%
	5	Ppbl{0}	0,706%	0,762%	0,818%	7,3%
		Ppbl{1}	0,576%	0,737%	0,897%	21,8%
	6	Ppbl{0}	1,052%	1,120%	1,189%	6,1%
		Ppbl{1}	1,207%	1,424%	1,641%	15,2%
3	1	Ppbl{0}	1,221%	1,494%	1,767%	18,3%
		Ppbl{1}	4,886%	5,669%	6,452%	13,8%
	2	Ppbl{0}	0,425%	0,512%	0,600%	17,1%
		Ppbl{1}	0,777%	1,034%	1,290%	24,8%
	3	Ppbl{0}	0,000%	0,000%	0,000%	0,0%
		Ppbl{1}	0,000%	0,000%	0,000%	0,0%
	4	Ppbl{0}	0,513%	0,679%	0,845%	24,5%
		Ppbl{1}	1,820%	2,103%	2,386%	13,5%
	5	Ppbl{0}	0,251%	0,412%	0,573%	39,1%
		Ppbl{1}	1,452%	1,782%	2,113%	18,5%
	6	Ppbl{0}	0,313%	0,397%	0,481%	21,2%
		Ppbl{1}	1,248%	1,575%	1,902%	20,8%
4	1	Ppbl{0}	0,065%	0,080%	0,096%	19,4%
		Ppbl{1}	0,081%	0,106%	0,131%	23,5%
	2	Ppbl{0}	1,399%	1,488%	1,577%	6,0%
		Ppbl{1}	1,786%	1,940%	2,093%	7,9%
	3	Ppbl{0}	0,821%	0,912%	1,004%	10,0%
		Ppbl{1}	1,672%	2,079%	2,485%	19,6%
	4	Ppbl{0}	0,000%	0,000%	0,000%	0,0%
		Ppbl{1}	0,000%	0,000%	0,000%	0,0%
	5	Ppbl{0}	0,265%	0,290%	0,314%	8,5%
		Ppbl{1}	0,251%	0,297%	0,342%	15,3%
	6	Ppbl{0}	0,322%	0,369%	0,415%	12,7%
		Ppbl{1}	0,157%	0,238%	0,319%	34,2%
5	1	Ppbl{0}	0,340%	0,405%	0,470%	16,0%
		Ppbl{1}	0,428%	0,502%	0,577%	14,8%
	2	Ppbl{0}	0,601%	0,636%	0,672%	5,6%
		Ppbl{1}	0,564%	0,686%	0,808%	17,7%
	3	Ppbl{0}	0,350%	0,478%	0,606%	26,8%
		Ppbl{1}	1,293%	1,888%	2,483%	31,5%
	4	Ppbl{0}	0,215%	0,241%	0,268%	11,0%
		Ppbl{1}	0,273%	0,317%	0,362%	13,9%
	5	Ppbl{0}	0,000%	0,000%	0,000%	0,0%
		Ppbl{1}	0,000%	0,000%	0,000%	0,0%
	6	Ppbl{0}	0,000%	0,000%	0,000%	0,0%
		Ppbl{1}	0,000%	0,000%	0,000%	0,0%
6	1	Ppbl{0}	0,478%	0,595%	0,710%	19,5%
		Ppbl{1}	0,701%	0,841%	0,982%	16,7%
	2	Ppbl{0}	0,945%	1,001%	1,058%	5,7%
		Ppbl{1}	1,102%	1,348%	1,595%	18,3%
	3	Ppbl{0}	0,331%	0,423%	0,515%	21,8%
		Ppbl{1}	1,181%	1,681%	2,180%	29,7%
	4	Ppbl{0}	0,245%	0,300%	0,355%	18,3%
		Ppbl{1}	0,202%	0,252%	0,302%	19,8%
	5	Ppbl{0}	0,000%	0,000%	0,000%	0,0%
		Ppbl{1}	0,000%	0,000%	0,000%	0,0%
	6	Ppbl{0}	0,000%	0,000%	0,000%	0,0%
		Ppbl{1}	0,000%	0,000%	0,000%	0,0%

Tab. G-36 – Intervalos de confiança para a probabilidade de bloqueio ponto a ponto por classe, para as classes 0 e 1 para o método RTNR (sistema 3).

fluxos	4
reserva	0
carga	1,4

alfa 5%

Average of valor					Limite Inferior	Média	Limite Superior	Conf (Rel)
origem	destino	item	metodo	idcorrida				
1	2	Ppbl{2}	DAR	mult_dar_res	33,914%	34,967%	36,019%	3,0%
			RTNR	mult	18,716%	19,814%	20,912%	5,5%
		Ppbl{3}	DAR	mult_dar_res	94,746%	95,875%	97,005%	1,2%
			RTNR	mult	95,249%	96,072%	96,894%	0,9%
	3	Ppbl{2}	DAR	mult_dar_res	16,311%	17,710%	19,110%	7,9%
			RTNR	mult	32,768%	34,060%	35,353%	3,8%
		Ppbl{3}	DAR	mult_dar_res	98,002%	98,757%	99,513%	0,8%
			RTNR	mult	95,258%	97,405%	99,552%	2,2%
	4	Ppbl{2}	DAR	mult_dar_res	0,352%	0,433%	0,514%	18,7%
			RTNR	mult	0,559%	0,707%	0,855%	20,9%
		Ppbl{3}	DAR	mult_dar_res	2,001%	2,449%	2,897%	18,3%
			RTNR	mult	4,629%	5,116%	5,604%	9,5%
	5	Ppbl{2}	DAR	mult_dar_res	5,311%	5,816%	6,321%	8,7%
			RTNR	mult	4,780%	5,001%	5,222%	4,4%
		Ppbl{3}	DAR	mult_dar_res	65,768%	66,685%	67,602%	1,4%
			RTNR	mult	43,476%	47,159%	50,841%	7,8%
	6	Ppbl{2}	DAR	mult_dar_res	7,859%	8,647%	9,434%	9,1%
			RTNR	mult	8,878%	9,450%	10,022%	6,1%
		Ppbl{3}	DAR	mult_dar_res	51,823%	55,053%	58,283%	5,9%
			RTNR	mult	72,171%	74,802%	77,432%	3,5%

Tab. G-37 – Intervalos de confiança para a probabilidade de bloqueio ponto a ponto por classe, para as classes 2 e 3, para o método DAR e RTNR (sistema 3).

Um Estudo Simulacional de Redes Inter-centrais com Encaminhamento Dinâmico

metodo	DAR
idcorrida	mult_dar_res
fluxos	4
reserva	0

alfa 5%

Average of valor				Limite Inferior	Média	Limite Superior	Conf (Rel)
origem	destino	carga	item				
1	2	1	Ppbl(0)	0,006%	0,016%	0,026%	63,8%
			Ppbl(1)	0,033%	0,089%	0,144%	63,0%
			Ppbl(2)	0,910%	1,144%	1,378%	20,5%
		1,1	Ppbl(0)	41,162%	44,434%	47,705%	7,4%
			Ppbl(1)	0,037%	0,073%	0,109%	49,0%
			Ppbl(2)	0,214%	0,317%	0,420%	32,5%
		1,2	Ppbl(0)	3,310%	3,627%	3,944%	8,7%
			Ppbl(1)	59,738%	62,128%	64,518%	3,8%
			Ppbl(2)	0,150%	0,175%	0,200%	14,2%
		1,3	Ppbl(0)	0,683%	0,890%	1,098%	23,3%
			Ppbl(1)	8,419%	8,823%	9,227%	4,6%
			Ppbl(2)	75,162%	76,855%	78,547%	2,2%
		1,4	Ppbl(0)	0,452%	0,532%	0,612%	15,0%
			Ppbl(1)	2,289%	2,539%	2,790%	9,9%
			Ppbl(2)	18,975%	19,875%	20,776%	4,5%
		1,5	Ppbl(0)	86,638%	89,346%	92,054%	3,0%
			Ppbl(1)	1,304%	1,429%	1,554%	8,8%
			Ppbl(2)	5,293%	5,651%	6,008%	6,3%
		1,6	Ppbl(0)	33,914%	34,967%	36,019%	3,0%
			Ppbl(1)	94,746%	95,875%	97,005%	1,2%
			Ppbl(2)	2,461%	2,663%	2,865%	7,6%
		1,7	Ppbl(0)	9,433%	9,815%	10,196%	3,9%
			Ppbl(1)	48,389%	49,282%	50,174%	1,8%
			Ppbl(2)	98,130%	98,562%	98,994%	0,4%
		1,8	Ppbl(0)	4,007%	4,161%	4,315%	3,7%
			Ppbl(1)	13,554%	14,324%	15,095%	5,4%
			Ppbl(2)	59,211%	59,672%	60,133%	0,8%
		1,9	Ppbl(0)	99,352%	99,658%	99,964%	0,3%
			Ppbl(1)	5,637%	5,820%	6,003%	3,1%
			Ppbl(2)	17,884%	18,422%	18,961%	2,9%
		2	Ppbl(0)	66,138%	67,117%	68,096%	1,5%
			Ppbl(1)	99,876%	99,954%	100,032%	0,1%
			Ppbl(2)	7,388%	7,706%	8,024%	4,1%
		3	Ppbl(0)	21,355%	22,161%	22,968%	3,6%
			Ppbl(1)	72,439%	72,701%	72,963%	0,4%
			Ppbl(2)	99,827%	99,954%	100,081%	0,1%
		1,1	Ppbl(0)	9,564%	9,771%	9,978%	2,1%
			Ppbl(1)	25,154%	25,934%	26,713%	3,0%
			Ppbl(2)	76,331%	77,176%	78,021%	1,1%
		1,2	Ppbl(0)	100,000%	100,000%	100,000%	0,0%
			Ppbl(1)	11,232%	11,564%	11,896%	2,9%
			Ppbl(2)	28,225%	28,931%	29,636%	2,4%
		1,3	Ppbl(0)	79,996%	80,629%	81,261%	0,8%
			Ppbl(1)	100,000%	100,000%	100,000%	0,0%
			Ppbl(2)	0,000%	0,028%	0,065%	135,0%
		1,4	Ppbl(0)	0,000%	0,021%	0,078%	277,6%
			Ppbl(1)	0,678%	1,635%	2,591%	58,5%
			Ppbl(2)	79,348%	81,912%	84,477%	3,1%
		1,5	Ppbl(0)	0,000%	0,065%	0,150%	132,2%
			Ppbl(1)	0,049%	0,238%	0,424%	79,4%
			Ppbl(2)	2,570%	3,573%	4,577%	28,1%
		1,6	Ppbl(0)	87,403%	89,790%	92,177%	2,7%
			Ppbl(1)	0,101%	0,151%	0,202%	33,2%
			Ppbl(2)	0,000%	0,387%	0,801%	106,8%
		1,7	Ppbl(0)	4,656%	5,485%	6,313%	15,1%
			Ppbl(1)	89,165%	91,153%	93,141%	2,2%
			Ppbl(2)	0,090%	0,272%	0,454%	66,7%
		1,8	Ppbl(0)	0,407%	0,946%	1,485%	57,0%
			Ppbl(1)	8,101%	9,830%	11,559%	17,6%
			Ppbl(2)	95,386%	96,860%	98,333%	1,5%
		1,9	Ppbl(0)	0,408%	0,487%	0,565%	16,1%
			Ppbl(1)	1,716%	2,241%	2,767%	23,5%
			Ppbl(2)	16,311%	17,710%	19,110%	7,9%
		2	Ppbl(0)	98,002%	98,757%	99,513%	0,8%
			Ppbl(1)	0,824%	0,947%	1,070%	13,0%
			Ppbl(2)	3,659%	4,054%	4,448%	9,7%
		3	Ppbl(0)	27,079%	27,792%	28,506%	2,6%
			Ppbl(1)	99,179%	99,783%	100,386%	0,6%
			Ppbl(2)	1,404%	1,562%	1,719%	10,1%
		1,1	Ppbl(0)	5,935%	6,269%	6,602%	5,3%
			Ppbl(1)	35,429%	36,253%	37,078%	2,3%
			Ppbl(2)	99,626%	99,901%	100,176%	0,3%
		1,2	Ppbl(0)	1,953%	2,279%	2,605%	14,3%
			Ppbl(1)	7,140%	7,860%	8,580%	9,2%
			Ppbl(2)	41,006%	42,419%	43,832%	3,3%
		1,3	Ppbl(0)	100,000%	100,000%	100,000%	0,0%
			Ppbl(1)	2,959%	3,243%	3,528%	8,8%
			Ppbl(2)	8,757%	9,931%	11,106%	11,8%
		1,4	Ppbl(0)	47,357%	48,301%	49,245%	2,0%
			Ppbl(1)	100,000%	100,000%	100,000%	0,0%
			Ppbl(2)	3,463%	3,909%	4,356%	11,4%
		1,5	Ppbl(0)	10,749%	11,996%	13,242%	10,4%
			Ppbl(1)	50,244%	52,185%	54,126%	3,7%
			Ppbl(2)	100,000%	100,000%	100,000%	0,0%
		1,6	Ppbl(0)	4,532%	4,729%	4,925%	4,2%
			Ppbl(1)	11,522%	12,699%	13,877%	9,3%
			Ppbl(2)	55,355%	56,771%	58,187%	2,5%
		1,7	Ppbl(0)	100,000%	100,000%	100,000%	0,0%
			Ppbl(1)				
			Ppbl(2)				

Tab. G-38 – Intervalos de confiança para a probabilidade de bloqueio ponto a ponto por classe, para as classes 2 e 3, para o método DAR para várias cargas (sistema 3).

4	1	Ppbl(0)	0,000%	0,000%	0,000%	0,0%	
		Ppbl(1)	0,000%	0,000%	0,000%	0,0%	
		Ppbl(2)	0,000%	0,000%	0,000%	0,0%	
		Ppbl(3)	0,000%	0,000%	0,000%	0,0%	
	1,1	Ppbl(0)	0,000%	0,000%	0,000%	0,0%	
		Ppbl(1)	0,000%	0,000%	0,000%	0,0%	
		Ppbl(2)	0,000%	0,000%	0,000%	0,0%	
		Ppbl(3)	0,000%	0,000%	0,000%	0,0%	
	1,2	Ppbl(0)	0,000%	0,000%	0,000%	0,0%	
		Ppbl(1)	0,000%	0,001%	0,004%	185,4%	
		Ppbl(2)	0,000%	0,005%	0,011%	113,3%	
		Ppbl(3)	0,000%	0,032%	0,080%	149,9%	
	1,3	Ppbl(0)	0,000%	0,002%	0,003%	76,6%	
		Ppbl(1)	0,002%	0,007%	0,011%	69,1%	
		Ppbl(2)	0,025%	0,041%	0,058%	40,1%	
		Ppbl(3)	0,215%	0,359%	0,503%	40,2%	
	1,4	Ppbl(0)	0,022%	0,028%	0,033%	19,9%	
		Ppbl(1)	0,078%	0,089%	0,100%	12,0%	
		Ppbl(2)	0,352%	0,433%	0,514%	18,7%	
		Ppbl(3)	2,001%	2,449%	2,897%	18,3%	
	1,5	Ppbl(0)	0,142%	0,153%	0,164%	7,3%	
		Ppbl(1)	0,466%	0,499%	0,532%	6,5%	
		Ppbl(2)	2,048%	2,139%	2,230%	4,2%	
		Ppbl(3)	9,726%	10,330%	10,933%	5,8%	
	1,6	Ppbl(0)	0,514%	0,556%	0,597%	7,5%	
		Ppbl(1)	1,445%	1,564%	1,684%	7,6%	
		Ppbl(2)	5,300%	5,624%	5,947%	5,8%	
		Ppbl(3)	22,362%	23,557%	24,752%	5,1%	
	1,7	Ppbl(0)	1,138%	1,171%	1,205%	2,9%	
		Ppbl(1)	2,991%	3,114%	3,236%	3,9%	
		Ppbl(2)	10,125%	10,488%	10,850%	3,5%	
		Ppbl(3)	38,347%	39,087%	39,828%	1,9%	
	1,8	Ppbl(0)	2,035%	2,108%	2,182%	3,5%	
		Ppbl(1)	4,933%	5,118%	5,303%	3,6%	
		Ppbl(2)	15,816%	16,221%	16,625%	2,5%	
		Ppbl(3)	53,142%	53,652%	54,161%	0,9%	
	1,9	Ppbl(0)	3,118%	3,162%	3,206%	1,4%	
		Ppbl(1)	7,072%	7,249%	7,427%	2,4%	
		Ppbl(2)	21,396%	21,822%	22,248%	2,0%	
		Ppbl(3)	64,493%	65,379%	66,266%	1,4%	
	2	Ppbl(0)	4,302%	4,421%	4,539%	2,7%	
		Ppbl(1)	9,513%	9,796%	10,078%	2,9%	
		Ppbl(2)	27,250%	27,627%	28,004%	1,4%	
		Ppbl(3)	75,013%	75,375%	75,736%	0,5%	
	5	1	Ppbl(0)	0,000%	0,001%	0,005%	277,6%
			Ppbl(1)	0,000%	0,000%	0,000%	0,0%
			Ppbl(2)	0,000%	0,000%	0,000%	0,0%
			Ppbl(3)	0,000%	0,723%	2,058%	184,6%
1,1		Ppbl(0)	0,000%	0,000%	0,000%	0,0%	
		Ppbl(1)	0,000%	0,013%	0,048%	277,6%	
		Ppbl(2)	0,000%	0,000%	0,000%	0,0%	
		Ppbl(3)	2,505%	5,953%	9,401%	57,9%	
1,2		Ppbl(0)	0,000%	0,007%	0,019%	185,1%	
		Ppbl(1)	0,000%	0,011%	0,043%	277,6%	
		Ppbl(2)	0,104%	0,342%	0,581%	69,7%	
		Ppbl(3)	22,249%	25,587%	28,924%	13,0%	
1,3		Ppbl(0)	0,000%	0,019%	0,044%	132,5%	
		Ppbl(1)	0,044%	0,125%	0,205%	64,7%	
		Ppbl(2)	1,688%	1,958%	2,228%	13,8%	
		Ppbl(3)	45,852%	49,073%	52,294%	6,6%	
1,4		Ppbl(0)	0,121%	0,170%	0,219%	28,7%	
		Ppbl(1)	0,493%	0,615%	0,737%	19,8%	
		Ppbl(2)	5,311%	5,816%	6,321%	8,7%	
		Ppbl(3)	65,768%	66,685%	67,602%	1,4%	
1,5		Ppbl(0)	0,372%	0,413%	0,454%	9,9%	
		Ppbl(1)	1,375%	1,624%	1,873%	15,3%	
		Ppbl(2)	11,952%	12,905%	13,858%	7,4%	
		Ppbl(3)	78,238%	80,975%	83,712%	3,4%	
1,6		Ppbl(0)	0,744%	0,889%	1,034%	16,3%	
		Ppbl(1)	3,421%	3,684%	3,946%	7,1%	
		Ppbl(2)	21,485%	22,204%	22,923%	3,2%	
		Ppbl(3)	89,018%	90,351%	91,683%	1,5%	
1,7		Ppbl(0)	1,572%	1,652%	1,732%	4,9%	
		Ppbl(1)	5,668%	5,937%	6,207%	4,5%	
		Ppbl(2)	31,104%	31,388%	31,673%	0,9%	
		Ppbl(3)	94,800%	95,335%	95,870%	0,6%	
1,8		Ppbl(0)	2,250%	2,393%	2,535%	6,0%	
		Ppbl(1)	7,919%	8,515%	9,110%	7,0%	
		Ppbl(2)	39,220%	40,216%	41,211%	2,5%	
		Ppbl(3)	97,481%	98,288%	99,095%	0,8%	
1,9		Ppbl(0)	3,273%	3,405%	3,537%	3,9%	
		Ppbl(1)	10,777%	11,097%	11,417%	2,9%	
		Ppbl(2)	46,695%	47,385%	48,076%	1,5%	
		Ppbl(3)	98,272%	98,937%	99,601%	0,7%	
2		Ppbl(0)	4,405%	4,588%	4,771%	4,0%	
		Ppbl(1)	13,503%	14,028%	14,553%	3,7%	
		Ppbl(2)	53,556%	54,007%	54,458%	0,8%	
		Ppbl(3)	99,106%	99,421%	99,736%	0,3%	

Tab. G-39 – Intervalos de confiança para a probabilidade de bloqueio ponto a ponto por classe, para as classes 2 e 3, para o método DAR para várias cargas (sistema 3) - continuação.

6	1	Ppbl(0)	0,001%	0,007%	0,012%	79,4%
		Ppbl(1)	0,000%	0,029%	0,079%	171,2%
		Ppbl(2)	0,246%	0,315%	0,384%	21,8%
		Ppbl(3)	8,698%	11,368%	14,038%	23,5%
	1,1	Ppbl(0)	0,001%	0,019%	0,037%	92,3%
		Ppbl(1)	0,066%	0,151%	0,236%	56,1%
		Ppbl(2)	0,801%	1,075%	1,350%	25,5%
		Ppbl(3)	16,070%	17,794%	19,518%	9,7%
	1,2	Ppbl(0)	0,036%	0,060%	0,084%	39,4%
		Ppbl(1)	0,186%	0,236%	0,286%	21,3%
		Ppbl(2)	1,746%	2,157%	2,569%	19,1%
		Ppbl(3)	24,197%	27,253%	30,309%	11,2%
	1,3	Ppbl(0)	0,115%	0,146%	0,177%	21,1%
		Ppbl(1)	0,486%	0,733%	0,979%	33,6%
		Ppbl(2)	3,914%	4,647%	5,380%	15,8%
		Ppbl(3)	35,619%	38,415%	41,211%	7,3%
	1,4	Ppbl(0)	0,278%	0,349%	0,420%	20,4%
		Ppbl(1)	1,433%	1,586%	1,738%	9,6%
		Ppbl(2)	7,859%	8,647%	9,434%	9,1%
		Ppbl(3)	51,823%	55,053%	58,283%	5,9%
	1,5	Ppbl(0)	0,642%	0,680%	0,717%	5,5%
		Ppbl(1)	2,364%	2,583%	2,803%	8,5%
		Ppbl(2)	13,946%	15,136%	16,325%	7,9%
		Ppbl(3)	72,670%	73,346%	74,022%	0,9%
	1,6	Ppbl(0)	1,102%	1,289%	1,475%	14,5%
		Ppbl(1)	3,938%	4,552%	5,167%	13,5%
		Ppbl(2)	22,111%	22,823%	23,535%	3,1%
		Ppbl(3)	83,850%	84,802%	85,755%	1,1%
	1,7	Ppbl(0)	1,907%	2,062%	2,217%	7,5%
		Ppbl(1)	6,217%	6,392%	6,568%	2,7%
		Ppbl(2)	29,226%	30,262%	31,299%	3,4%
		Ppbl(3)	90,937%	92,200%	93,462%	1,4%
	1,8	Ppbl(0)	2,804%	2,953%	3,102%	5,1%
		Ppbl(1)	7,784%	8,447%	9,111%	7,9%
		Ppbl(2)	36,252%	37,604%	38,956%	3,6%
		Ppbl(3)	94,938%	95,626%	96,314%	0,7%
	1,9	Ppbl(0)	3,815%	4,010%	4,205%	4,9%
		Ppbl(1)	10,380%	10,977%	11,574%	5,4%
		Ppbl(2)	43,184%	44,042%	44,899%	1,9%
		Ppbl(3)	96,854%	97,534%	98,215%	0,7%
	2	Ppbl(0)	4,760%	5,009%	5,258%	5,0%
		Ppbl(1)	12,937%	13,451%	13,964%	3,8%
Ppbl(2)		49,465%	49,868%	50,272%	0,8%	
Ppbl(3)		98,219%	98,579%	98,939%	0,4%	

Tab. G-40 – Intervalos de confiança para a probabilidade de bloqueio ponto a ponto por classe, para as classes 2 e 3, para o método DAR para várias cargas (sistema 3) - conclusão.

metodo		RTNR		alfa	5%		
item		Ropl					
idcorrida		mult					
Average of valor			Limite Inferior	Média	Limite Superior	Conf (Rel)	
carga	origem	destino					
1	1	1	2	3,146%	3,415%	3,683%	7,9%
			3	0,098%	0,145%	0,192%	32,4%
			4	0,000%	0,000%	0,000%	0,0%
		5	0,071%	0,115%	0,158%	37,7%	
		6	0,007%	0,023%	0,039%	69,5%	
		2	3	0,000%	0,000%	0,000%	0,0%
			4	0,070%	0,089%	0,108%	21,0%
			5	0,000%	0,000%	0,001%	277,6%
		6	0,000%	0,004%	0,009%	136,4%	
		3	4	22,642%	23,243%	23,643%	1,7%
			5	0,000%	0,000%	0,000%	0,0%
			6	1,482%	1,646%	1,810%	10,0%
	4	5	0,000%	0,000%	0,000%	0,0%	
		6	0,084%	0,110%	0,136%	23,4%	
		6	0,000%	0,000%	0,000%	0,0%	
	1,2	1	2	11,943%	12,406%	12,870%	3,7%
			3	2,700%	2,870%	3,040%	5,9%
			4	0,000%	0,000%	0,000%	0,0%
			5	1,639%	1,898%	2,158%	13,7%
			6	1,191%	1,323%	1,454%	10,0%
			3	0,000%	0,001%	0,003%	277,6%
		2	4	3,709%	3,845%	3,981%	3,5%
			5	0,549%	0,601%	0,653%	8,6%
			6	0,771%	0,857%	0,943%	10,0%
3		4	33,776%	34,442%	35,108%	1,9%	
		5	0,000%	0,001%	0,003%	170,0%	
		6	6,220%	6,644%	7,068%	6,4%	
4	5	0,000%	0,000%	0,001%	277,6%		
	6	3,446%	3,629%	3,812%	5,0%		
	6	0,000%	0,000%	0,000%	0,0%		
1,4	1	2	28,341%	28,710%	29,078%	1,3%	
		3	17,113%	17,449%	17,784%	1,9%	
		4	0,665%	0,703%	0,742%	5,5%	
		5	16,300%	16,520%	16,741%	1,3%	
		6	13,466%	13,619%	13,773%	1,1%	
		2	3	1,492%	1,555%	1,618%	4,1%
			4	15,642%	15,861%	16,080%	1,4%
			5	10,960%	11,104%	11,249%	1,3%
		3	4	10,289%	10,498%	10,707%	2,0%
			5	47,675%	48,169%	48,663%	1,0%
			6	3,338%	3,481%	3,624%	4,1%
		4	5	23,101%	23,632%	24,163%	2,2%
	6		2,456%	2,501%	2,547%	1,8%	
	6		17,017%	17,314%	17,610%	1,7%	
	1,6	1	2	33,847%	33,934%	34,020%	0,3%
			3	19,272%	19,487%	19,701%	1,1%
			4	5,782%	5,911%	6,040%	2,2%
			5	22,955%	23,079%	23,203%	0,5%
			6	18,212%	18,631%	19,050%	2,2%
			2	3,113%	3,204%	3,295%	2,8%
		2	4	22,090%	22,285%	22,479%	0,9%
			5	17,078%	17,220%	17,362%	0,8%
			6	16,562%	16,740%	16,918%	1,1%
		3	4	53,381%	53,656%	53,932%	0,5%
5			6,123%	6,321%	6,520%	3,1%	
6			30,367%	30,626%	30,884%	0,8%	
4	5	5,623%	5,716%	5,810%	1,6%		
	6	23,837%	24,102%	24,367%	1,1%		
	6	0,000%	0,000%	0,000%	0,0%		
1,8	1	2	36,220%	36,481%	36,742%	0,7%	
		3	22,371%	22,558%	22,746%	0,8%	
		4	13,363%	13,476%	13,590%	0,8%	
		5	27,124%	27,324%	27,523%	0,7%	
		6	20,791%	21,359%	21,927%	2,7%	
		2	3	5,037%	5,186%	5,334%	2,9%
			4	28,684%	28,840%	28,996%	0,5%
			5	21,485%	21,613%	21,740%	0,6%
		6	21,788%	21,990%	22,192%	0,9%	
		3	4	56,919%	56,991%	57,063%	0,1%
			5	7,511%	7,689%	7,866%	2,3%
			6	35,711%	35,950%	36,189%	0,7%
	4	5	8,580%	8,660%	8,739%	0,9%	
		6	28,771%	29,034%	29,296%	0,9%	
		6	0,000%	0,000%	0,000%	0,0%	
	2	1	2	40,181%	40,440%	40,698%	0,6%
			3	25,230%	25,495%	25,761%	1,0%
			4	21,146%	21,328%	21,509%	0,9%
			5	29,771%	30,033%	30,296%	0,9%
			6	24,657%	25,043%	25,430%	1,5%
			3	7,444%	7,584%	7,723%	1,8%
		2	4	35,080%	35,266%	35,453%	0,5%
			5	26,398%	26,481%	26,564%	0,3%
			6	27,755%	28,056%	28,356%	1,1%
3		4	58,642%	58,874%	59,107%	0,4%	
		5	8,542%	8,874%	9,206%	3,7%	
		6	40,059%	40,257%	40,455%	0,5%	
4	5	12,553%	12,607%	12,660%	0,4%		
	6	34,806%	35,000%	35,195%	0,6%		
	6	0,003%	0,007%	0,012%	57,0%		

Tab. G-41 – Intervalos de confiança para a probabilidade de bloqueio nos feixes com o método RTNR (sistema 2).

metodo		DAR		alfa		5%	
item		Rpbl					
reserva		0					
fluxos		2					
Average of valor	origem	destino	Limite Inferior	Média	Limite Superior	Conf (Rel)	
1	1	1	2	3,121%	3,381%	3,642%	7,7%
			3	0,043%	0,057%	0,072%	25,1%
			4	0,000%	0,000%	0,000%	0,0%
		2	3	0,075%	0,111%	0,147%	32,9%
			4	0,013%	0,028%	0,043%	52,8%
			6	0,000%	0,000%	0,000%	0,0%
	3	3	0,060%	0,076%	0,093%	21,8%	
		4	0,000%	0,000%	0,001%	277,6%	
		6	0,000%	0,005%	0,011%	107,3%	
	4	4	22,534%	23,014%	23,495%	2,1%	
		5	0,000%	0,000%	0,000%	0,0%	
		6	1,008%	1,134%	1,259%	11,1%	
	5	5	0,000%	0,000%	0,000%	0,0%	
		6	0,068%	0,093%	0,117%	26,8%	
		6	0,000%	0,000%	0,000%	0,0%	
1,2	1	1	2	10,382%	10,806%	11,230%	3,9%
			3	1,127%	1,379%	1,630%	18,3%
			4	0,000%	0,000%	0,000%	0,0%
		2	5	1,503%	1,751%	1,998%	14,1%
			6	0,711%	0,752%	0,793%	5,5%
			3	0,000%	0,000%	0,000%	0,0%
	3	4	3,196%	3,312%	3,428%	3,5%	
		5	0,714%	0,774%	0,834%	7,8%	
		6	0,599%	0,683%	0,766%	12,3%	
	4	4	31,900%	32,414%	32,928%	1,6%	
		5	0,026%	0,059%	0,092%	56,3%	
		6	3,941%	4,238%	4,535%	7,0%	
	5	5	0,000%	0,004%	0,010%	161,3%	
		6	2,907%	3,146%	3,386%	7,6%	
		6	0,000%	0,000%	0,000%	0,0%	
1,4	1	1	2	26,157%	26,726%	27,295%	2,1%
			3	11,238%	11,576%	11,914%	2,9%
			4	0,167%	0,194%	0,221%	13,8%
		2	5	8,449%	8,761%	9,074%	3,6%
			6	7,115%	7,469%	7,822%	4,7%
			3	0,060%	0,081%	0,102%	26,2%
	3	4	13,604%	13,791%	13,979%	1,4%	
		5	8,641%	8,814%	8,987%	2,0%	
		6	7,305%	7,456%	7,607%	2,0%	
	4	4	48,268%	48,637%	49,006%	0,8%	
		5	2,101%	2,226%	2,351%	5,6%	
		6	15,897%	16,354%	16,810%	2,8%	
	5	5	1,085%	1,134%	1,182%	4,3%	
		6	13,300%	13,593%	13,886%	2,2%	
		6	0,000%	0,000%	0,000%	0,0%	
1,6	1	1	2	44,110%	44,368%	44,626%	0,6%
			3	30,755%	30,970%	31,185%	0,7%
			4	5,728%	5,880%	6,031%	2,6%
		2	5	28,405%	28,609%	28,813%	0,7%
			6	22,298%	22,662%	23,025%	1,6%
			3	1,415%	1,541%	1,667%	8,2%
	3	4	25,701%	25,912%	26,124%	0,8%	
		5	19,618%	19,744%	19,870%	0,6%	
		6	18,477%	18,705%	18,933%	1,2%	
	4	4	67,229%	67,517%	67,805%	0,4%	
		5	7,507%	7,803%	8,099%	3,8%	
		6	34,303%	34,565%	34,827%	0,8%	
	5	5	7,750%	7,833%	7,916%	1,1%	
		6	26,372%	26,627%	26,881%	1,0%	
		6	0,000%	0,000%	0,000%	0,0%	
1,8	1	1	2	56,238%	56,375%	56,512%	0,2%
			3	46,807%	47,102%	47,397%	0,6%
			4	15,009%	15,103%	15,196%	0,6%
		2	5	43,898%	44,092%	44,287%	0,4%
			6	34,935%	35,532%	36,128%	1,7%
			3	5,881%	6,186%	6,490%	4,9%
	3	4	36,189%	36,379%	36,568%	0,5%	
		5	29,068%	29,225%	29,382%	0,5%	
		6	28,916%	29,128%	29,341%	0,7%	
	4	4	78,162%	78,336%	78,510%	0,2%	
		5	13,947%	14,443%	14,938%	3,4%	
		6	47,200%	47,431%	47,663%	0,5%	
	5	5	17,755%	17,925%	18,095%	0,9%	
		6	37,706%	37,960%	38,215%	0,7%	
		6	0,000%	0,000%	0,000%	0,0%	
2	1	1	2	63,930%	64,214%	64,498%	0,4%
			3	57,643%	57,948%	58,253%	0,5%
			4	23,270%	23,409%	23,547%	0,6%
		2	5	53,385%	53,684%	53,983%	0,6%
			6	44,865%	45,287%	45,709%	0,9%
			3	13,336%	13,819%	14,301%	3,5%
	3	4	44,324%	44,543%	44,762%	0,5%	
		5	36,993%	37,066%	37,138%	0,2%	
		6	37,331%	37,652%	37,974%	0,9%	
	4	4	83,370%	83,576%	83,782%	0,2%	
		5	19,326%	19,995%	20,664%	3,3%	
		6	55,888%	56,144%	56,400%	0,5%	
	5	5	27,191%	27,317%	27,443%	0,5%	
		6	46,311%	46,512%	46,712%	0,4%	
		6	0,000%	0,009%	0,018%	104,8%	

Tab. G-42 – Intervalos de confiança para a probabilidade de bloqueio nos feixes com o método DAR (sistema 2).

metodo	RTNR
item	Rpbl
idcorrida	mult
reserva	0
fluxos	4

alfa 5%

Average of valor			Limite Inferior	Média	Limite Superior	Conf (Rel)		
carga	origem	destino						
1	1	1	2	2,933%	3,170%	3,406%	7,5%	
			3	1,944%	2,422%	2,899%	19,7%	
			4	0,000%	0,000%	0,000%	0,0%	
		5	1,594%	1,662%	1,731%	4,1%		
		6	0,490%	0,641%	0,792%	23,5%		
		2	3	0,000%	0,000%	0,000%	0,0%	
			4	0,109%	0,200%	0,291%	45,6%	
			5	0,002%	0,022%	0,042%	90,8%	
		6	0,024%	0,047%	0,070%	48,6%		
		3	4	22,839%	23,821%	24,803%	4,1%	
			5	0,000%	0,016%	0,035%	115,3%	
			6	1,769%	2,014%	2,258%	12,2%	
	4	5	0,000%	0,000%	0,000%	0,0%		
		6	0,146%	0,164%	0,182%	11,0%		
	1,2	1	1	2	6,165%	6,351%	6,538%	2,9%
				3	4,563%	4,864%	5,166%	6,2%
				4	0,000%	0,001%	0,002%	174,7%
			5	3,629%	3,784%	3,939%	4,1%	
			6	2,009%	2,135%	2,261%	5,9%	
			2	3	0,001%	0,010%	0,019%	85,2%
		4		4,061%	4,148%	4,235%	2,1%	
		5		0,948%	1,031%	1,113%	8,0%	
		6	1,268%	1,442%	1,616%	12,1%		
		3	4	34,609%	35,267%	35,924%	1,9%	
5			0,088%	0,149%	0,210%	40,8%		
6			7,706%	8,023%	8,339%	3,9%		
4	5	4	0,008%	0,010%	0,011%	19,7%		
		5	3,846%	3,989%	4,131%	3,6%		
		6	0,000%	0,000%	0,000%	0,0%		
	6	2	14,888%	15,336%	15,784%	2,9%		
		3	11,683%	11,973%	12,264%	2,4%		
		4	0,268%	0,298%	0,328%	10,0%		
1,4	1	1	2	8,483%	8,679%	8,875%	2,3%	
			3	6,097%	6,409%	6,721%	4,9%	
			4	0,887%	0,988%	1,088%	10,2%	
		5	14,708%	15,007%	15,307%	2,0%		
		6	9,445%	9,701%	9,956%	2,6%		
		2	3	9,041%	9,292%	9,542%	2,7%	
			4	46,885%	47,380%	47,875%	1,0%	
			5	2,161%	2,263%	2,366%	4,5%	
		6	20,915%	21,243%	21,570%	1,5%		
		3	4	1,577%	1,690%	1,802%	6,7%	
			5	15,164%	15,469%	15,774%	2,0%	
			6	0,000%	0,000%	0,000%	0,0%	
	1,6	1	1	2	20,746%	20,925%	21,104%	0,9%
				3	14,496%	14,777%	15,057%	1,9%
				4	3,550%	3,727%	3,904%	4,7%
			5	12,012%	12,138%	12,265%	1,0%	
			6	14,789%	15,123%	15,457%	2,2%	
			2	3	3,073%	3,181%	3,290%	3,4%
		4		22,567%	22,682%	22,797%	0,5%	
		5		17,673%	17,827%	17,981%	0,9%	
		6	17,066%	17,205%	17,344%	0,8%		
		3	4	52,972%	53,505%	54,039%	1,0%	
			5	5,910%	6,087%	6,263%	2,9%	
			6	30,584%	30,901%	31,218%	1,0%	
4	5	4	5,933%	6,042%	6,151%	1,8%		
		5	24,878%	25,072%	25,265%	0,8%		
		6	0,000%	0,000%	0,000%	0,0%		
	6	2	24,709%	25,077%	25,446%	1,5%		
		3	16,325%	16,765%	17,205%	2,6%		
		4	8,894%	9,131%	9,368%	2,6%		
1,8	1	1	2	14,625%	14,868%	15,110%	1,6%	
			3	18,097%	18,409%	18,721%	1,7%	
			4	5,069%	5,256%	5,443%	3,6%	
		5	28,276%	28,446%	28,615%	0,6%		
		6	22,143%	22,401%	22,659%	1,2%		
		2	3	22,023%	22,327%	22,632%	1,4%	
			4	57,948%	58,452%	58,955%	0,9%	
			5	7,433%	7,633%	7,833%	2,6%	
		3	4	35,902%	36,129%	36,356%	0,6%	
			5	9,069%	9,265%	9,462%	2,1%	
			6	29,384%	29,596%	29,808%	0,7%	
		2	1	1	2	0,000%	0,000%	0,000%
	3				28,585%	28,867%	29,149%	1,0%
	4				18,772%	19,122%	19,473%	1,8%
	5			13,880%	13,985%	14,091%	0,8%	
	6			18,019%	18,406%	18,793%	2,1%	
	2			3	7,431%	7,521%	7,611%	1,2%
			4	34,709%	34,827%	34,944%	0,3%	
			5	26,768%	26,920%	27,072%	0,6%	
	3		4	27,668%	27,825%	27,983%	0,6%	
			5	60,436%	60,638%	60,840%	0,3%	
			6	8,409%	8,603%	8,796%	2,2%	
	4		5	4	40,043%	40,383%	40,724%	0,8%
		5		12,733%	12,865%	12,997%	1,0%	
6		34,456%		34,554%	34,652%	0,3%		
6		4	0,005%	0,009%	0,013%	48,1%		

Tab. G-43 – Intervalos de confiança para a probabilidade de bloqueio nos feixes com o método RTNR (sistema 3).

metodo	DAR
item	Rpbl
idcorrida	mult_dar_res
reserva	0
fluxos	4

alfa 5%

Average of valor			Limite Inferior	Média	Limite Superior	Conf (Rel)			
carga	origem	destino							
1	1	1	2	2,594%	2,719%	2,845%	4,6%		
			3	2,181%	2,440%	2,698%	10,6%		
			4	0,000%	0,000%	0,000%	0,0%		
			5	1,169%	1,333%	1,498%	12,3%		
			6	0,509%	0,627%	0,745%	18,8%		
			6	0,000%	0,000%	0,000%	0,0%		
	2	2	3	4	0,107%	0,131%	0,156%	18,8%	
				5	0,000%	0,002%	0,004%	105,7%	
				6	0,017%	0,022%	0,026%	19,9%	
				4	4	22,670%	23,652%	24,633%	4,2%
					5	0,000%	0,010%	0,023%	122,5%
					6	1,069%	1,293%	1,517%	17,3%
	4	5	4	0,000%	0,001%	0,001%	129,7%		
			6	0,083%	0,101%	0,118%	17,2%		
	1.2	1	1	2	5,230%	5,539%	5,848%	5,6%	
				3	4,499%	4,819%	5,139%	6,6%	
				4	0,000%	0,002%	0,004%	100,4%	
				5	2,633%	2,831%	3,029%	7,0%	
				6	1,161%	1,484%	1,807%	21,7%	
				6	0,000%	0,000%	0,000%	0,0%	
2		3	3	0,000%	0,000%	0,000%	0,0%		
			4	3,066%	3,141%	3,217%	2,4%		
			5	0,526%	0,586%	0,647%	10,3%		
			6	0,549%	0,635%	0,721%	13,5%		
			4	4	32,044%	32,915%	33,785%	2,6%	
				5	0,045%	0,076%	0,107%	41,1%	
4		5	4	4,358%	4,796%	5,235%	9,1%		
			6	0,018%	0,025%	0,031%	25,6%		
1.4		1	1	2	3,124%	3,195%	3,266%	2,2%	
				3	0,000%	0,000%	0,000%	0,0%	
				4	13,967%	14,364%	14,762%	2,8%	
				5	11,489%	11,921%	12,353%	3,6%	
				6	0,138%	0,161%	0,184%	14,4%	
				6	5,489%	5,540%	5,591%	0,9%	
	2	3	3	4,271%	4,485%	4,699%	4,8%		
			4	0,026%	0,046%	0,065%	43,4%		
			5	12,520%	12,702%	12,884%	1,4%		
			6	7,436%	7,609%	7,782%	2,3%		
			4	4	6,188%	6,535%	6,883%	5,3%	
				6	46,442%	46,907%	47,371%	1,0%	
	3	5	3	1,853%	1,989%	2,125%	6,8%		
			4	14,578%	14,829%	15,081%	1,7%		
			5	0,864%	0,904%	0,943%	4,4%		
			6	12,804%	13,031%	13,259%	1,7%		
			4	4	0,000%	0,000%	0,000%	0,0%	
				6	27,069%	27,499%	27,930%	1,6%	
	1.6	1	1	2	23,942%	24,074%	24,207%	0,5%	
				3	1,761%	1,889%	2,016%	6,7%	
4				12,489%	12,924%	13,359%	3,4%		
5				11,137%	11,548%	11,959%	3,6%		
6				1,066%	1,188%	1,310%	10,3%		
6				23,761%	23,927%	24,092%	0,7%		
2		3	3	18,328%	18,465%	18,603%	0,7%		
			4	16,955%	17,246%	17,536%	1,7%		
			5	62,342%	62,601%	62,861%	0,4%		
			6	6,822%	6,966%	7,109%	2,1%		
			4	4	30,561%	31,007%	31,453%	1,4%	
				6	5,787%	5,890%	5,993%	1,7%	
3		5	3	24,255%	24,359%	24,463%	0,4%		
			4	0,000%	0,000%	0,000%	0,0%		
			5	38,188%	38,634%	39,079%	1,2%		
			6	35,428%	36,285%	37,142%	2,4%		
			4	4	5,347%	5,464%	5,582%	2,1%	
				6	23,107%	23,286%	23,466%	0,8%	
1.8		1	1	2	19,753%	20,199%	20,645%	2,2%	
				3	5,317%	5,691%	6,066%	6,6%	
	4			33,531%	33,691%	33,851%	0,5%		
	5			27,636%	27,919%	28,203%	1,0%		
	6			27,084%	27,456%	27,827%	1,4%		
	6			72,862%	73,117%	73,372%	0,3%		
	2	3	3	12,464%	12,690%	12,915%	1,8%		
			4	43,890%	44,378%	44,866%	1,1%		
			5	14,342%	14,351%	14,360%	0,1%		
			6	34,369%	34,419%	34,468%	0,1%		
			4	4	0,000%	0,000%	0,000%	0,0%	
				6	47,027%	47,328%	47,628%	0,6%	
	2	1	1	2	46,017%	46,271%	46,525%	0,5%	
				3	9,577%	9,748%	9,919%	1,8%	
				4	32,908%	33,208%	33,508%	0,9%	
				5	28,381%	28,627%	28,873%	0,9%	
				6	12,601%	13,052%	13,503%	3,5%	
				6	41,559%	41,706%	41,853%	0,4%	
		2	3	3	35,851%	35,971%	36,091%	0,3%	
				4	35,894%	36,074%	36,253%	0,5%	
5				79,268%	79,502%	79,736%	0,3%		
6				17,937%	18,239%	18,540%	1,7%		
4				4	53,531%	53,670%	53,808%	0,3%	
				6	23,027%	23,189%	23,350%	0,7%	
3		5	3	42,746%	42,810%	42,873%	0,1%		
			4	0,005%	0,008%	0,011%	40,2%		

Tab. G-44 – Intervalos de confiança para a probabilidade de bloqueio nos feixes com o método DAR (sistema 3).

# Apêndice H

## Outros Resultados da Simulação

### H.1 Experiência computacional com o Sistema 2

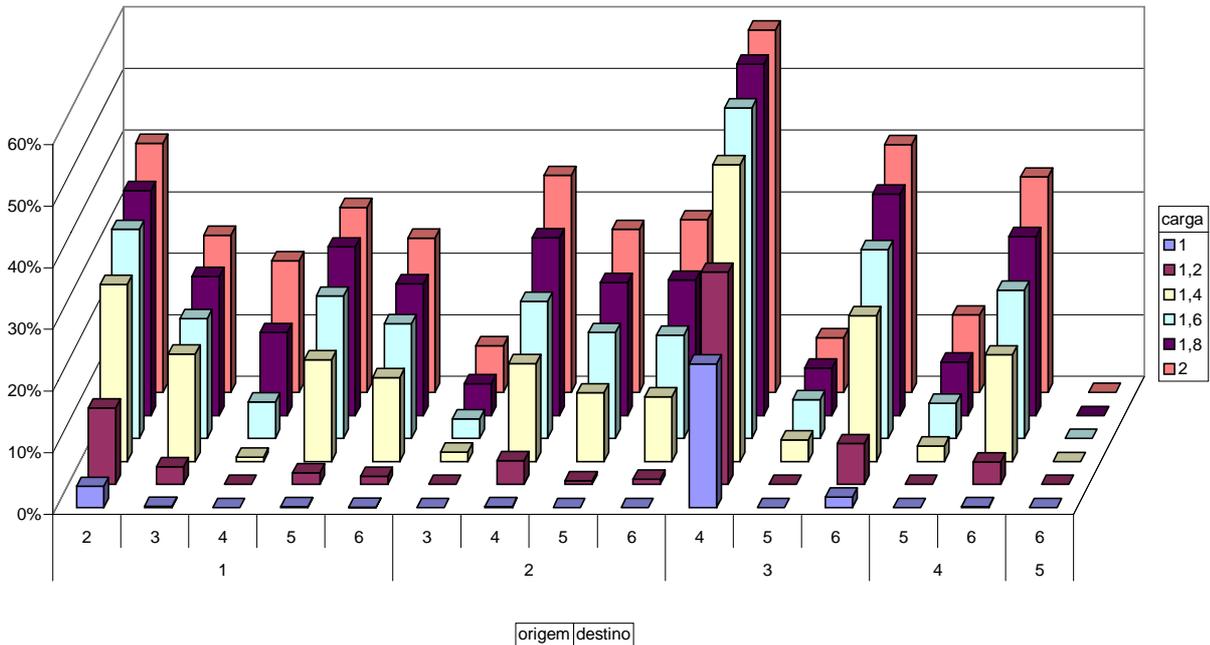
#### H.1.1 Análise da probabilidade de bloqueio nos feixes

Nos gráficos H-1 e H-2 podem ser observadas as probabilidades de bloqueio de chamadas nos feixes com o método de encaminhamento RTNR e DAR respectivamente, para vários valores do factor de sobrecarga.

Podemos observar no gráfico H-1 que a probabilidade de bloqueio nos feixes aumenta em todos os feixes à medida que aumenta o valor do factor de sobrecarga. Podemos verificar que os feixes mais sobrecarregados são os feixes entre as centrais 1-2, 2-4, 3-4, 3-6 e 4-6, os mesmos que no sistema 1, existindo porém para alguns factores de sobrecarga ligeiras excepções, conforme visível na tabela G-41 do apêndice G.

metodo|RTNR|item|Rpbl|idcorrida|mult

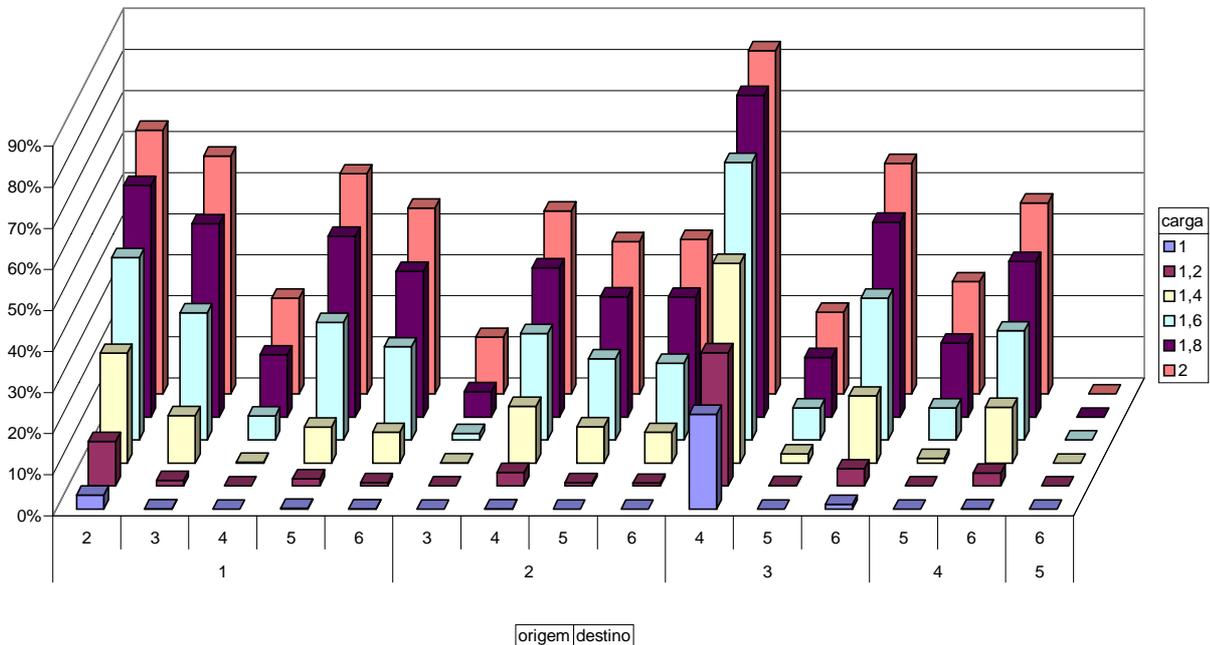
Average of valor



Gráf. H-1 – Probabilidade de bloqueio nos feixes com o método RTNR (sistema 2).

metodo|DAR|item|Rpbl|reserva|0|fluxos|2

Average of valor



Gráf. H-2 – Probabilidade de bloqueio nos feixes com o método DAR (sistema 2).

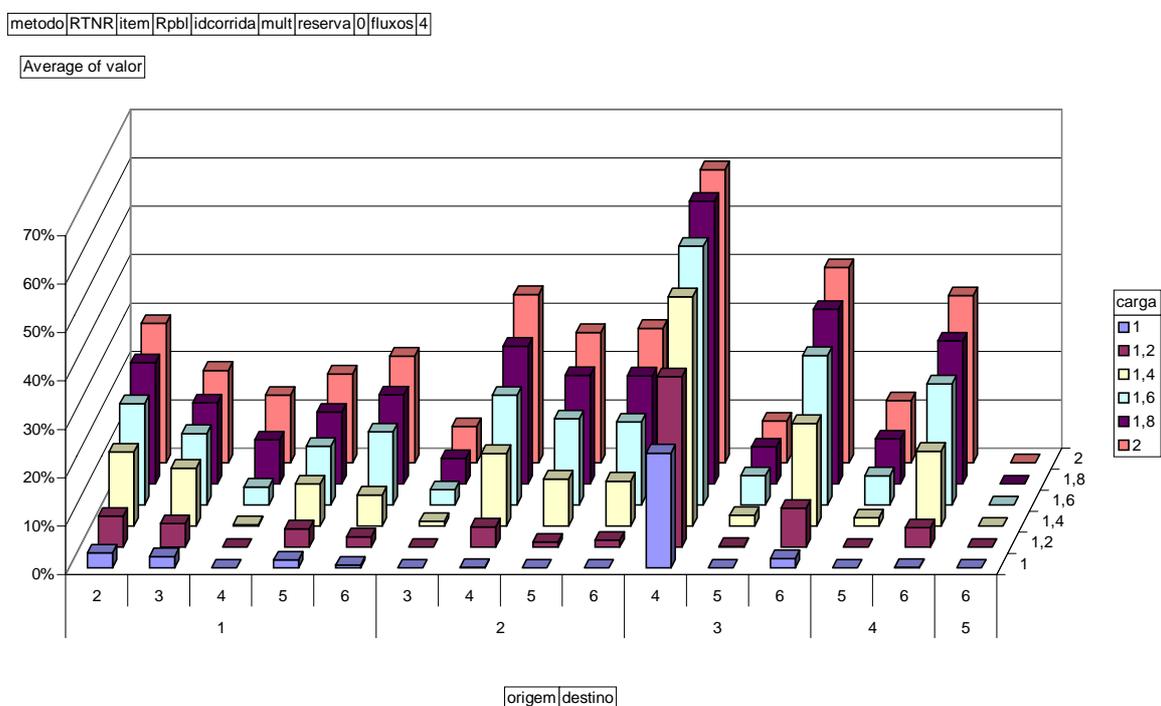
No gráfico H-2 podemos também observar que a probabilidade de bloqueio nos feixes aumenta em todos os feixes à medida que aumenta o valor do factor de sobrecarga. Aqui os feixes mais sobrecarregados

são em geral os feixes entre as centrais 1-2, 1-3, 1-5, 3-4 e 3-6, em que alguns deles são os mesmos que no sistema 1 e os outros são feixes que embora não sendo os mais sobrecarregados, no sistema 1, são feixes com grande probabilidade de bloqueio. Os intervalos de confiança para um grau de confiança de 95% correspondentes a estes valores encontram-se na tabela G-42 do apêndice G

## H.2 Experiência computacional com o Sistema 3

### H.2.1 Análise da probabilidade de bloqueio nos feixes

Nos gráficos H-3 e H-4 podem ser observadas as probabilidades de bloqueio de chamadas nos feixes com o método de encaminhamento RTNR e DAR respectivamente, para vários valores do factor de sobrecarga.

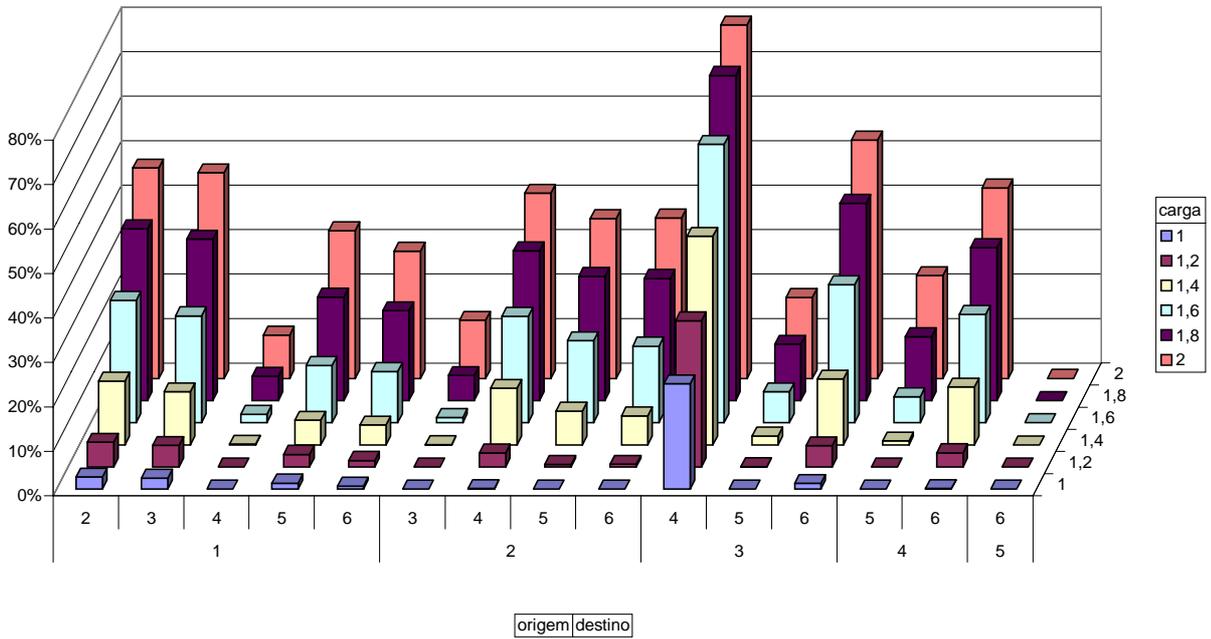


Gráf. H-3 – Probabilidade de bloqueio nos feixes com o método RTNR (sistema 3).

Podemos observar no gráfico H-3 que a probabilidade de bloqueio nos feixes aumenta em todos os feixes à medida que aumenta o valor do factor de sobrecarga. Podemos verificar que os feixes mais sobrecarregados são os feixes entre as centrais 1-2, 2-4, 3-4, 3-6 e 4-6, os mesmos que no sistema 1 e sistema 2, existindo porém para alguns factores de sobrecarga ligeiras excepções (para cargas muito leves), conforme visível na tabela G-43 do apêndice G.

metodo|DAR|item|Rpb|idcorrida|mult\_dar\_res|reserva|0|fluxos|4

Average of valor



Gráf. H-4 – Probabilidade de bloqueio nos feixes com o método DAR (sistema 3).

No gráfico H-4 podemos também observar que a probabilidade de bloqueio nos feixes aumenta em todos os feixes à medida que aumenta o valor do factor de sobrecarga. Aqui os feixes mais sobrecarregados são em geral os feixes entre as centrais 1-2, 1-3, 3-4 e 3-6, os quais também já eram os mais congestionados no sistema 2. Os intervalos de confiança para um grau de confiança de 95% correspondentes a estes valores encontram-se na tabela G-44 do apêndice G.

# Bibliografia

- [Ash90] Gerald R. Ash, "Design and Control of Networks with Dynamic Nonhierarchical Routing", *IEEE Communication Magazine*, Vol. 28, N° 10, pp. 34-40, New York, October 1990
- [Ash91] Gerald R. Ash, J.-S. Chen, A. E. Frey e B. D. Huang, "Real Time Network Routing in a Dynamic Class-of-Service Network", em *Proceedings of the 13th International Teletraffic Congress*, pp. 187-194, 1991
- [Ash93] Gerald R. Ash e BaoSheng D. Huang, "An Analitical Model for Adaptive Routing Networks", *IEEE Transactions on Communications*, Vol.41, N° 11, pp. 1748-1759, New York, November 1993
- [Ash94] Gerald R. Ash e BaoSheng D. Huang, "Comparative Evaluation of Dynamic Routing Strategies for a Worldwide Intelligent Network", em *Proceedings of the 14th International Teletraffic Congress*, pp. 677-686, 1994
- [Ash95] Gerald R. Ash, "Dynamic Network Evolution, with Examples from AT&T's Evolving Dynamic Network", *IEEE Communication Magazine*, Vol. 33, N° 7, pp. 26-39, New York, July 1995
- [Ash98] Gerald R. Ash, *Dynamic Routing in Telecommunications Networks*, Mc Graw Hill, New York, 1998
- [Banks96] Jerry Banks, John S. Carson e Barry L. Nelson, *Discrete-Event System Simulation*, Prentice Hall, 2nd Edition, New Jersey, 1996
- [Bel85] G. Bel, P. Chemouil, J. M. Garcia, F. Le Gall e J. Bernussou, "Adaptative Traffic Routing in Telephone Networks", *Large Scale Systems*, Vol. 8, N° 3, pp. 267-282, 1985

- [Berezner97] S. A. Berezner e A. E. Krzesinsky, “Hybrid Alternative Routing Strategies in Communication Networks”, em *Proceedings of the 15th International Teletraffic Congress*, pp. 169-178, 1997
- [Cameron86] H. Cameron, “Simulation of dynamic routing: critical path selection features for service and economy”, em *International Conference on Communications*, Vol. 81, pp. 55.5.1-55.5.6, 1986
- [Chemouil90] Prosper Chemouil, Janusz Filipiak e Paul Gauthier, “Performance Issues in the Design of Dynamically Controlled Circuit Switched Networks”, *IEEE Communication Magazine*, Vol. 28, Nº 10, pp. 90-95, New York, October 1990
- [Chung93] Shun-Ping Chung, Arik Kasper e Keith W. Ross, “Computing Approximate Blocking Probabilities for Large Loss Networks with State-Dependent Routing”, *IEEE/ACM Transactions on Networking*, Vol. 1, Nº 1, pp. 105-115, February 1993
- [Cooper81] Robert B. Cooper, *Introduction to Queuing Theory*, North Holland, 2nd Edition, 1981
- [Craveirinha01] José Craveirinha, Lúcia Martins, Teresa Gomes, Carlos H. Antunes e João Clímaco, “Formulation of a Multiple Objective Dynamic Routing Method using Implied Costs”, submetido para publicação, Fevereiro 2001
- [Dudewicz88] Edward J. Dudewicz e Satya N. Mishra, *Modern Mathematical Statistics*, John Wiley and Sons, New York, 1988
- [Esteves91] J. S. Esteves, “Metodologias de Análise e Cálculo Numérico em Redes de Teletráfego com Transbordos Múltiplos”, Coimbra, 1991
- [Garzia90] Mario R. Garzia, “A Study of Network Adaptative Routing”, em *Network Modeling, Simulation and Analysis*, Marcel Dekker, Inc, pp. 211-235, USA, 1990
- [Gibbens93] Richard J. Gibbens, Frank P. Kelly e Stephen R. E. Turner, “Dynamic Routing in Multiparented Networks”, *IEEE/ACM Transactions on Networking*, Vol. 1, Nº 2, pp. 261-270, New York, April 1993
- [Girard90] André Girard, *Routing and Dimensioning in Circuit-Switched Networks*, Addison-Wesley Publishing Company, U.S.A., 1990

- 
- [Gomes97] Teresa M. S. Gomes, “Um Modelo de Análise de Fiabilidade-Qualidade de Serviço em Redes Inter-Centrais”, Coimbra, 1997
- [Howard60] R. A. Howard, *Dynamic Programming and Markov Processes*, MIT Press, Cambridge, MA, 1960
- [ITU-T97a] T. Oda (Contact), *Recommendation E.529: Network dimensioning using end-to-end GOS objectives*, Study Period 1997-2000, Japan, May 1997
- [ITU-T97b] G. Ash e A. Maunder (Contacts), *Draft text for Recommendation E.1XX: Dynamic routing interworking*, USA, May 1997
- [Jobart00] Alain Jobart, “Beginners Guide to C++”, <http://www.cybersoft.org/c++/browser.html>, Setembro 2000
- [Kashper95] A. N. Kashper e Y. Watanabe, “Dynamic Routing in the Multiple Carrier International Network”, *IEEE Communication Magazine*, Vol 33, Nº 7, pp. 54-61, July 1995
- [Kawashima95] Konosuke Kawashima e Akiya Inoue, “State- and Time-Dependent Routing in the NTT Network”, *IEEE Communication Magazine*, Vol 33, Nº 7, pp. 40-47, July 1995
- [Key88] Peter B. Key e M. J. Whitehead, “Cost-Effective Use of Networks Employing Dynamic Alternative Routing”, em *Proceedings of the 12th International Teletraffic Congress*, pp. 3.1A4.1-3.1A4.11, Torino, Italy, June 1988
- [Key90] Peter B. Key e Graham A. Cope, “Distributed Dynamic Routing Schemes”, *IEEE Communication Magazine*, Vol. 28, Nº 10, pp. 54-64, New York, October 1990
- [Kobayashi78] H. Kobayashi, *Modelling and Analysis*, John Wiley, New York, 1978
- [Koussoulas93] Nick T. Koussoulas, “Performance Analysis of Circuit-Switched Networks with State-Dependent Routing”, *IEEE Transactions on Communications*, Vol.41,Nº 11, pp. 1647-1655, New York, November 1993
- [Krishnan89] K. R. Krishnan e T. J. Ott, “Forward Looking Routing: A New State-Dependent Routing Scheme”, em *Proceedings of the 12th International Teletraffic Congress*, pp. 1026-1032, Torino, Italy, 1989

- [Krishnan90] K. R. Krishnan, "Markov Decision Algorithms for Dynamic Routing", *IEEE Communication Magazine*, Vol. 28, N° 10, pp. 66-69, New York, October 1990
- [Krishnan91] K. R. Krishnan, "Adaptative State-Dependent Traffic Routing Using On-Line Trunk-Group Measurements", em *Proceedings of the 13th International Teletraffic Congress*, pp. 407-411, 1991
- [Larman98] Craig Larman, *Applying UML and patterns: an introduction to object oriented analysis and design*, Prentice-Hall, USA, 1998
- [Law91] Averill M. Law e W. David Kelton, *Simulation Modeling & Analysis*, McGraw-Hill, Inc, 2nd Edition, U.S.A., 1991
- [Law94] Averill M. Law e Michael G. McComas, "Simulation Software for Communications Networks: The State of the Art", *IEEE Communication Magazine*, Vol. 32, N° 3, pp. 44-50, New York, March 1994
- [Mase90] Kenichi Mase e Hisao Yamamoto, "Advanced Traffic Control Methods for Network Management", *IEEE Communication Magazine*, Vol. 28, N° 10, pp. 82-88, New York, October 1990
- [Mitra91a] Debasis Mitra e Judith B. Seery, "Comparative Evaluations of Randomized and Dynamic Routing Strategies for Circuit-Switched Networks", *IEEE Transactions on Communications*, Vol. 39, N° 1, pp. 102-116, New York, January 1991
- [Mitra91b] Debasis Mitra, Richard J. Gibbens e B. D. Huang, "Analysis and Optimal Design of Aggregated\_Least-Busy-Alternative Routing on Symmetric Loss Networks with Trunk Reservations", em *Proceedings of the 13th International Teletraffic Congress*, pp. 477-482, 1991
- [Mitra93] Debasis Mitra, Richard J. Gibbens e Baosheng D. Huang, "State-Dependent Routing on Symmetric Loss Networks with Trunk Reservations - I", *IEEE Transactions on Communications*, Vol. 41, N° 2, pp. 400-411, New York, February 1993
- [Oda97] Toshikane Oda e Prosper Chemouil, "Dynamic Routing Interworking", em *Proceedings of the 15th International Teletraffic Congress*, Hot Topic Session, pp. 1-11, 1997
- [Olsson75] Olsson, *Computation - ITU Seminar on Network Planning*, Part II, Nova Delhi, 1975

- 
- [Rational00] Rational Corp., “Rational Rose, Visual Modelling, UML, Object Oriented, Component-Based Development”, <http://www.rational.com/products/rose/>, Julho, 2000
- [Régnier90] Jean Régnier e W. Hugh Cameron, “State-Dependent Dynamic Traffic Management for Telephone Networks”, *IEEE Communication Magazine*, Vol. 28, Nº 10, pp. 42-53, New York, October 1990
- [Régnier95] Jean Régnier, François Bédard, Jean Choquette e Alain Caron, “Dynamically Controlled Routing in Networks with Non-DCR-Compliant Switches”, *IEEE Communication Magazine*, Vol 33, Nº 7, pp. 48-52, July 1995
- [Simões99] Carlos. A. T. Simões, “Sistemas de Teletráfego em RDIS-BE - Modelos e Algoritmos”, Coimbra, 1999
- [Stroustrup97] Bjarne Stroustrup, *The C++ Programming Language*, Addison Wesley, USA, 1997
- [Syski59] R. Syski, *Introduction to Congestion Theory in Telephone Systems*, Oliver and Boyd, Edinburgh, 1959
- [Topkis86] Donald M. Topkis, “Reordering Heuristics for Routing in Communications Networks”, *Journal of Applied Probability*, Vol. 23, pp. 130-143, Israel, 1986
- [Varga00] András Varga, “OMNeT++ Discrete Event Simulation System - User Manual”, <http://www.hit.bme.hu/phd/vargaa/opp-docs/usman.htm>, Julho 2000
- [Vargas96] Cesar Vargas, Manjunath V. Hedge, Morteza Nareghi-Pour e Paul S. Min, “Shadow Prices for LLR e ALBA”, *IEEE/ACM Transactions on Networking*, Vol. 4, Nº 5, pp. 796-807, October 1996
- [Watanabe90] Yu Watanabe e Toshikane Oda, “Dynamic Routing Schemes for International Networks”, *IEEE Communication Magazine*, Vol. 28, Nº 10, pp. 70-75, New York, October 1990
- [Wolf90] Richard B. Wolf, “Advanced Techniques For Managing Telecommunications Networks”, *IEEE Communication Magazine*, Vol. 28, Nº 10, pp. 76-81, New York, October 1990

- [Yum87] Tak-kin G. Yum e Mischa Schwartz, "Comparison of Routing Procedures for Circuit-Switched Traffic in Nonhierarchical Networks", *IEEE Transactions on Communications*, Vol. Com-35, N° 5, pp. 535-544, New York, May 1987