

Paulo José Lopes Machado Portugal

**Avaliação da Confiança no Funcionamento de Redes
de Campo**

**Contribuição no Domínio dos Sistemas Industriais de
Controlo**

Dissertação submetida para a obtenção do grau de
Doutor em Engenharia Electrotécnica e de Computadores
pela Faculdade de Engenharia da Universidade do Porto

Trabalho realizado sob a orientação do Professor Doutor

Adriano da Silva Carvalho

Professor Associado com Agregação
da Faculdade de Engenharia da Universidade do Porto

Universidade do Porto
Faculdade de Engenharia
Departamento de Engenharia Electrotécnica e de Computadores
2004

APOIOS

O autor foi durante parte da realização desta dissertação bolseiro
da Fundação para a Ciência e a Tecnologia

PROGRAMA PRAXIS XXI / BD / 13799 / 97

*À memória do meu Pai por todo seu o esforço e dedicação realizados no sentido
de me propiciar uma melhor educação.*

AGRADECIMENTOS

Em primeiro lugar, desejo exprimir a minha amizade e o meu reconhecimento ao Professor Doutor Adriano da Silva Carvalho, meu orientador desde o início da minha carreira académica, pela confiança depositada, pela forma atenta e esclarecida como sempre orientou a minha actividade científica e pedagógica e pelo empenho que colocou na obtenção das condições que possibilitaram a preparação desta dissertação.

A todos os amigos e colegas que me apoiaram e incentivaram ao longo da preparação desta dissertação, em particular aos colegas da Secção de Automação, Controlo e Sistemas de Produção Industrial pelas excelentes relações de camaradagem e pelo bom ambiente de trabalho que sempre existiu entre nós.

Mais do que todos à minha família – Mãe, Fernando, Linda e João - pelo seu constante apoio e paciência sempre demonstradas.

“...there are no fault-free systems — either hardware or software — there are only systems which have not yet failed....”

Jean-Claude Laprie, 1992

| Resumo

As redes locais industriais constituem presentemente a estrutura sobre a qual assentam as arquitecturas dos sistemas de controlo distribuídos. A rápida evolução tecnológica nesta área, criou as condições necessárias que permitiram o desenvolvimento de aplicações de controlo cada vez mais complexas e sofisticadas. Em muitos destes sistemas, o comportamento na presença de falhas ou de avarias torna-se crítico, quer em termos dos custos decorrentes da indisponibilidade do processo, quer em termos do seu impacto sobre as pessoas e o ambiente. Torna-se assim necessário depositar maior confiança no funcionamento destes sistemas durante todo o seu ciclo de vida. Este aspecto torna-se particularmente importante ao nível das arquitecturas baseadas em redes de campo, dado que estas, ao realizarem a interface directa com o processo industrial potenciam, na ocorrência de falhas, as maiores consequências sobre os mesmos.

O processo de validação de um sistema com estas características é uma tarefa considerável, devido ao vasto conjunto de domínios de conhecimento envolvidos e às complexas interações existentes entre os vários componentes do sistema. Entre as várias metodologias utilizadas neste processo de validação, a utilização de modelos de avaliação estocásticos assume especial relevância devido à sua capacidade de prever o comportamento do sistema na presença de falhas. A sua flexibilidade aliada ao seu crescente poder de modelação têm permitido a sua utilização com sucesso na avaliação de sistemas cada vez mais complexos.

Ao nível das redes de campo, os trabalhos desenvolvidos no contexto da avaliação da confiança no funcionamento são relativamente reduzidos. Na sua maioria focam-se em aspectos específicos, e não permitem obter uma visão global sobre o comportamento da rede. Contudo, a avaliação segundo esta perspectiva é fundamental dado que a rede opera como uma entidade única. O objectivo desta dissertação centrou-se assim em preencher parte desta lacuna, nomeadamente através do desenvolvimento de modelos estocásticos que permitam avaliar, a partir de diferentes perspectivas, a confiança no funcionamento de arquitecturas de controlo suportadas por redes de campo.

Para as arquitecturas de controlo onde a informação disponível é escassa, nomeadamente durante as fases iniciais de concepção e desenvolvimento, é proposta uma metodologia com o objectivo de desenvolver modelos preliminares de avaliação. Estes são baseados em cadeias de Markov e incluem os efeitos de falhas permanentes ao nível dos componentes mais importantes da rede, de falhas transitórias durante as comunicações e de avarias em modo-comum em ambos os cenários. Para além de uma solução analítica, é possível também extrair dos modelos expressões das medidas de interesse e identificar os parâmetros mais importantes, facilitando assim o processo de avaliação.

Para arquitecturas de controlo mais complexas, é proposta uma metodologia com o objectivo de desenvolver um modelo que inclua diferentes perspectivas sobre o funcionamento da rede, e que assim possibilite a obtenção de um modelo integrado de avaliação. Os modelos propostos são baseados em redes de Petri Estocásticas e permitem obter uma solução analítica do problema. Estes incluem os efeitos de falhas permanentes e transitórias e de avarias em modo-comum, que afectam diferentes tipos de componentes situados em diferentes níveis funcionais. São considerados vários de modos de avaria da rede, em função da respectiva gravidade. Os aspectos relacionados com a dimensão do espaço de estados e com a natureza dos processos estocásticos envolvidos foram especialmente equacionados, sendo discutidas e utilizadas técnicas específicas para a sua resolução.

Dada a importância sobre a operação do sistema dos efeitos das falhas transitórias que ocorrem durante as comunicações, são desenvolvidos modelos rigorosos e detalhados para a avaliação das redes Profibus-DP e CAN neste contexto. Os modelos são também baseados em redes de Petri Estocásticas, mas em que a solução é obtida através de simulação. Neste contexto, são discutidos e analisados os vários factores que condicionam este tipo de soluções.

Esta página foi intencionalmente deixada em branco

| Abstract

Industrial communication networks are presently the backbone of distributed control systems architectures. The fast technological evolution in this domain has created the necessary conditions that allowed the development and use of complex and high performance control applications. In many of these systems the behavior in the presence of faults and failures becomes critical, due to the costs of the system unavailability, and the impact to the people and environment. So its necessary that these systems be dependable during all their life-cycle. This becomes particularly important for Fieldbus networks, since their direct interface with the industrial process enables faults to have a major impact on the system operation.

The validation process of a system with these characteristics is a complex task, due to the complex interactions between system components and the large number of knowledge domains involved. The use of stochastic model based techniques to evaluate dependability has been gaining wide acceptance, since it enables to foreseen the system behavior in the the presence of faults. Their flexibility and modeling power shows that is feasible to used it to evaluate dependability of complex systems.

Research on dependability evaluation of fieldbus networks has been somewhat limited. Most of the works focuses at particular aspects without a global perspective of the network behavior. This perspective is fundamental, because the network operates as a single entity. The thesis main goal is to fill this gap, by developing stochastic models which enable dependability evaluation of control architectures supported by fieldbus networks to be performed using different viewpoints.

For control architectures with limited available data, especially during the development phases, a methodology is proposed to develop preliminary evaluation models. These models are based on Markov chains, and they include the effects of permanent faults on the most important system components, transient fault during communications and common-mode failures in both scenarios. Models solution are analytic ones and expressions of dependability measures can be extracted from it.

For complex control architectures, it is proposed an approach where different viewpoints of the network behavior are merged into an integrated dependability model. These are analytical models based on Stochastic Petri Nets. They include the effects of permanent and transient faults, and common-mode failures affecting different components at different functional levels. Several system failure modes are considered with different severities. The problems related to the size of the state space and to the nature of the involved stochastic processes have been especially equated.

Since transient faults that occur during communications have a major importance in the network operation, they are developed accurate and detailed dependability models to evaluate the behavior of Profibus-DP and CAN in these scenarios. The models are also based on Stochastic Petri Nets, but simulation is used to obtain results. The limitations and problems of using this approach are analyzed and discussed.

Esta página foi intencionalmente deixada em branco

Résumé

Les réseaux locaux industriels représentent actuellement la structure sur laquelle sont basées les architectures des systèmes de contrôle distribués. La rapide évolution technologique dans ce secteur, a créé les conditions nécessaires qui ont permis le développement d'applications de contrôle de plus en plus complexes et sophistiquées. Dans beaucoup de ces systèmes, le comportement en présence de fautes ou de pannes se rend critique, dû aux coûts de l'indisponibilité de système, et de l'impact pour l'environnement et du peuple. Se rend ainsi nécessaire déposer plus grande confiance en le fonctionnement de ces systèmes pendant tout son cycle de vie. Cet aspect se rend particulièrement important au niveau des architectures basées sur des réseaux de terrain, puisque leur interface directe avec le processus industriel permet à des fautes d'avoir un impact important sur l'exploitation du système.

Le processus de validation d'un système avec ces caractéristiques est une tâche considérable, dû au vaste ensemble de domaines de connaissance engagés et aux complexes interactions existantes entre les plusieurs composantes du système. Entre les plusieurs méthodologies utilisées dans ce processus de validation, l'utilisation de modèles d'évaluation stochastiques suppose spéciale importance, dû à sa capacité de déprévoir le comportement du système en présence des fautes. En outre, sa flexibilité alliée à son croissant pouvoir de modelage ont permis son utilisation avec succès dans l'évaluation de systèmes de plus en plus complexes.

Au niveau des réseaux de terrain, les travaux développés dans le contexte de l'évaluation de la sûreté en le fonctionnement relativement sont réduits. Pour ils la plupart focalisent seulement dans des aspects spécifiques, et ils ne permettent pas d'obtenir une vision globale sur le comportement du réseaux. Cette perspective est fondamentale, parce que le réseau fonctionne comme entité simple. L'objectif de cette dissertation s'est centré ainsi à remplir cette lacune, notamment à travers le développement de modèles stochastiques qui permettent d'évaluer de forme intégrée, y compris de différentes perspectives, la sûreté de fonctionnement d'architectures de contrôle supportées par des réseaux de terrain.

Pour des architectures de commande avec peu de données disponibles, notamment pendant les phases initiales de conception, est proposée une méthodologie avec l'objectif de développer des modèles préliminaires d'évaluation. Ces modèles sont basés sur des chaînes de Markov, et ils incluent les effets des fautes permanents sur les composants de système les plus importants, le fautes pendant les communications et les fautes de origine-commune dans les deux scénarios. La solution de modèles sont analytique, et des expressions des mesures de fiabilité peuvent être extraites à partir d'elle, facilitant ainsi le processus d'évaluation.

Pour des architectures de contrôle plus complexes, est proposée une méthodologie avec l'objectif de développer un modèle qui inclue de différentes perspectives sur le fonctionnement du réseau, et ainsi rende possible l'obtention d'un modèle intégré d'évaluation. Les modèles proposés sont basés sur des réseaux de Petri Stochastiques et permettent d'obtenir une solution analytique du problème. Ceux-ci incluent les effets des fautes permanentes, transitoires et de origine-commune, laquelle affectent de différents types de composantes les différents niveaux. Sont considérés plusieurs de manières de panne du filet, en fonction de la respective gravité. Les aspects rapportés avec la dimension de l'espace d'états et avec la nature des procédures aléatoires impliquées surtout ont été égalés, étant discutés et utilisés des techniques spécifiques pour leur résolution.

Donnée de l'importance sur l'opération du système de l'effet des fautes transitoires qui se produisent pendant les communications, sont développés des modèles détaillés et rigoureux pour l'évaluation des réseaux Profibus-DP et CAN dans ce contexte. Les modèles aussi sont basés sur des réseaux de Petri Stochastiques, mais dans que la solution est obtenue à travers simulation. Dans ce contexte, ils sont discutés et analysés les plusieurs facteurs qui conditionnent ce type de solutions.

Esta página foi intencionalmente deixada em branco

| Índice

Resumo.....	xi
Abstract.....	xiii
Résumé.....	xv
Índice.....	xvii
Índice de Figuras.....	xxiii
Índice de Tabelas.....	xxvii
 Capítulo 1	
Introdução.....	1
1.1 Contexto.....	1
1.2 Sistema de Controlo.....	3
1.3 Evolução das Tecnologias de Controlo.....	5
1.4 Arquitectura das Comunicações Industriais.....	9
1.5 Redes de Campo.....	11
1.6 Motivação.....	14
1.7 Contribuições da Dissertação.....	21
1.8 Estrutura da Dissertação.....	22
1.9 Publicações Resultantes da Dissertação.....	24
 Capítulo 2	
Conceitos e Técnicas de Avaliação da Confiança no	
Funcionamento.....	25
2.1 Introdução.....	25
2.2 Conceitos e Terminologia Associados à Confiança no Funcionamento.....	26
2.2.1 Definição de Confiança no Funcionamento.....	27

2.2.2 Impedimentos à Confiança no Funcionamento.....	29
2.2.2.1 Falhas.....	30
2.2.2.2 Erros.....	32
2.2.2.3 Avarias.....	32
2.2.2.4 Patologia das Avarias.....	34
2.2.3 Meios Para Obter Confiança no Funcionamento.....	34
2.2.3.1 Prevenção de Falhas.....	35
2.2.3.2 Tolerância a Falhas.....	35
2.2.3.2.1 Processamento de Erros.....	36
2.2.3.2.2 Tratamento de Falhas.....	37
2.2.3.3 Supressão de Falhas.....	37
2.2.3.4 Previsão de Falhas.....	38
2.2.4 Atributos da Confiança no Funcionamento.....	38
2.3 Técnicas de Avaliação da Confiança no Funcionamento.....	39
2.3.1 Formalismos de Modelação Estocásticos.....	41
2.4 Medidas da Confiança no Funcionamento.....	47
2.4.1 Relações Fundamentais.....	48
2.4.1.1 Probabilidade de Avaria.....	48
2.4.1.2 Taxa de Avaria.....	48
2.4.1.3 Fiabilidade.....	51
2.4.1.4 Segurança.....	51
2.4.1.5 Disponibilidade.....	52
2.4.1.6 Taxa de Reparação.....	52
2.4.1.7 Capacidade de Manutenção.....	52
2.4.1.8 Valores Esperados.....	52
2.5 Processos Estocásticos.....	53
2.5.1 Processos de Renovação.....	56
2.5.1.1 Processo de Poisson.....	57
2.5.2 Processos de Renovação de Markov.....	58
2.5.3 Processos de Markov Discretos.....	60
2.5.4 Processos de Markov Contínuos.....	60
2.5.5 Análise de Processos Markovianos.....	61
2.5.5.1 Cadeias de Markov Discretas.....	61
2.5.5.2 Cadeias de Markov Contínuas.....	64
2.5.6 Processos de Semi-Markov.....	67
2.5.7 Processos de Markov Regenerativos.....	68
2.5.8 Processos Generalizados de Semi-Markov.....	69
2.5.9 Análise de Processos Não-Markovianos.....	70
2.5.9.1 Expansão por Fases.....	71
2.5.9.2 Variáveis Suplementares.....	72
2.5.9.3 Teoria da Renovação	73
2.6 Modelos de Recompensa.....	75
2.6.1 Relação com as Medidas da Confiança no Funcionamento.....	77
2.7 Redes de Petri.....	78
2.7.1 Redes de Petri Ordinárias.....	79
2.7.2 Redes de Petri Extendidas.....	81
2.7.3 Redes de Petri Temporizadas e Estocásticas.....	83

2.7.3.1 Política de Disparo.....	85
2.7.3.2 Política de Memória.....	85
2.7.3.3 Política de Concorrência.....	87
2.7.3.4 Redes de Petri Estocásticas.....	88
2.7.4 Redes de Petri Estocásticas Markovianas.....	92
2.7.5 Redes de Petri Estocásticas Não-Markovianas.....	93
2.8 Simulação.....	97
2.9 Conclusões.....	106

Capítulo 3

Avaliação Preliminar da Confiança no Funcionamento.....109

3.1 Introdução.....	109
3.2 Discussão dos Trabalhos Relevantes Previamente Desenvolvidos.....	111
3.3 Metodologia Proposta.....	113
3.4 Modelo de Falhas Permanentes.....	118
3.4.1 Método dos Estados de Funcionamento Crítico.....	119
3.4.1.1 Descrição das Condições de Aplicação.....	120
3.4.2 Pressupostos Assumidos.....	124
3.4.3 Architecturas Avaliadas.....	126
3.4.4 Modelo mR : m nós redundantes.....	127
3.4.5 Modelo mD : m nós dispensáveis.....	129
3.4.6 Avarias em Modo-Comum.....	130
3.4.7 Modelo mRCS : mR Modo-comum simplificado.....	132
3.4.8 Modelo mRCC : mR Modo-comum complexo.....	133
3.4.9 Modelo mRD : mR Dual.....	136
3.4.10 Modelo mRDCC : mR Dual modo-comum complexo.....	138
3.4.11 Extensões dos Modelos.....	141
3.4.12 Caso de Estudo.....	142
3.4.12.1 Avaliação de Resultados.....	145
3.5 Modelo de Falhas Transitórias.....	145
3.5.1 Pressupostos Assumidos.....	146
3.5.2 Camada Física.....	147
3.5.2.1 Falhas Pontuais.....	147
3.5.2.2 Falhas com Duração Associada.....	149
3.5.2.3 Meio Físico de Transmissão Redundante.....	151
3.5.2.3.1 Avarias em Modo-Comum.....	152
3.5.3 Camada de Ligação de Dados.....	154
3.5.3.1 Modelo da Transmissão das Mensagens.....	154
3.5.3.2 Modelos de Avaria da Rede.....	157
3.5.3.2.1 Transmissão Periódica de Mensagens.....	160
3.5.3.2.2 Transmissão Aperiódica de Mensagens.....	166
3.5.4 Caso de Estudo.....	168
3.5.4.1 Avaliação de Resultados.....	176
3.6 Conclusões.....	177

Capítulo 4

Avaliação Integrada da Confiança no Funcionamento.....179

4.1 Introdução.....	179
4.2 Definição do Problema.....	180
4.2.1 Características dos Sistemas de Controlo Suportados por Redes de Campo.....	182
4.2.2 Discussão dos Trabalhos Relevantes Previamente Desenvolvidos.....	184
4.2.3 Modelação Utilizando Redes de Petri Estocásticas.....	189
4.2.3.1 Aspectos Prévios ao Desenvolvimento de Modelos Baseados em Redes de Petri.....	190
4.2.3.1.1 Dimensão do Espaço de Estados.....	190
4.2.3.1.2 Ordem de Grandeza dos Parâmetros.....	193
4.2.4 Metodologia Utilizada no Desenvolvimento dos Modelos.....	193
4.3 Modelos Desenvolvidos.....	200
4.3.1 Modelo HW.....	200
4.3.2 Modelo SW.....	207
4.3.2.1 Modelo SW Local.....	210
4.3.2.2 Modelo SW Global.....	211
4.3.3 Modelo CL.....	214
4.3.4 Modelo BUS.....	217
4.3.5 Modelos das Reparações.....	223
4.3.5.1 Reparação Local.....	223
4.3.5.2 Reparação Global.....	225
4.3.6 Avarias em Modo-Comum.....	230
4.3.7 Notas Finais Sobre os Modelos.....	231
4.4 Caso de Estudo.....	233
4.4.1 Modelo de Avaliação Completo.....	234
4.4.2 Definição das Medidas de Interesse.....	235
4.4.3 Resultados Obtidos.....	236
4.5 Conclusões.....	240

Capítulo 5

Avaliação da Rede Profibus-DP em Cenários de Falhas

Transitórias.....243

5.1 Introdução.....	243
5.2 O Profibus.....	244
5.2.1 Camada Física.....	245
5.2.2 Camada de Ligação de Dados.....	246
5.2.3 Camada de Aplicação.....	248
5.3 Discussão dos Trabalhos Relevantes Previamente Desenvolvidos.....	249
5.3.1 Comportamento em Cenários de Falhas Durante as Comunicações.....	249
5.3.2 Escalonamento de Mensagens em Sistemas de Tempo-Real.....	251
5.4 Metodologia Proposta.....	252
5.4.1 Pressupostos Assumidos.....	253
5.4.2 Ferramenta da Modelação.....	255
5.5 Modelos Desenvolvidos.....	256

5.5.1 Modelo do Mestre.....	257
5.5.1.1 Passagem do Token.....	258
5.5.1.2 Transmissão do Token.....	261
5.5.1.3 Rotação do Token.....	265
5.5.1.4 Gestão do Anel.....	268
5.5.1.5 Re-Inicialização do Anel.....	271
5.5.1.6 Re-Inicialização do Mestre.....	272
5.5.1.7 Transmissão de Mensagens.....	272
5.5.1.8 Gestão dos Escravos.....	275
5.5.2 Modelo do Escravo.....	280
5.5.3 Modelo Global.....	282
5.5.4 Parâmetros do Modelo.....	283
5.5.4.1 Probabilidades Associadas às Transições.....	283
5.5.4.2 Duração das Transições.....	287
5.5.5 Definição das Medidas de Interesse.....	287
5.5.6 Modelo de Avaliação Completo.....	288
5.6 Caso de Estudo.....	289
5.6.1 Definição dos Valores dos Parâmetros.....	289
5.6.2 Caracterização do Desempenho do Modelo.....	292
5.6.3 Resultados Obtidos.....	294
5.7 Conclusões.....	298

Capítulo 6

Avaliação da Rede CAN em Cenários de Falhas Transitórias...301

6.1 Introdução.....	301
6.2 O CAN.....	302
6.2.1 Camada Física.....	302
6.2.2 Camada de Ligação de Dados.....	304
6.2.3 Camada de Aplicação.....	308
6.3 Discussão dos Trabalhos Relevantes Previamente Desenvolvidos.....	310
6.3.1 Comportamento Geral em Cenários de Falhas.....	311
6.3.2 Escalonamento de Mensagens em Cenários de Falhas Durante as Comunicações.....	313
6.4 Metodologia Proposta.....	315
6.4.1 Pressupostos Assumidos.....	316
6.5 Modelos Desenvolvidos.....	318
6.5.1 Modelo da Ocorrência das Falhas.....	318
6.5.2 Modelo da Transmissão de uma Mensagem.....	320
6.5.3 Modelo da Sinalização dos Erros.....	323
6.5.4 Extensão do Modelo a Várias Mensagens.....	325
6.5.5 Definição das Medidas de Interesse.....	325
6.6 Caso de Estudo.....	326
6.6.1 Parâmetros do Modelo.....	326
6.6.1.1 Definição das Mensagens.....	326
6.6.1.2 Definição dos Cenários de Ocorrência de Falhas.....	327
6.6.2 Solução do Modelo.....	328

6.6.2.1 Redes de Petri Discretas Determinísticas e Estocásticas.....	328
6.6.2.2 Expansão por Fases Total.....	330
6.6.2.3 Expansão por Fases Parcial.....	331
6.6.2.4 Simulação.....	331
6.6.3 Caracterização do Desempenho do Modelo.....	332
6.6.4 Resultados Obtidos.....	334
6.7 Conclusões.....	337

Capítulo 7

Conclusões.....	339

7.1 Perspectivas de Trabalhos Futuros.....	344

Anexo 1.....	349

Anexo 2.....	353

Anexo 3.....	381

Bibliografia.....	409

| Índice de Figuras

Figura 1.1 - Contexto da dissertação.....	2
Figura 1.2 - Sistema de controlo.....	3
Figura 1.3 - Exemplos de arquitecturas de controlo.....	4
Figura 1.4 - Relação entre os níveis de controlo e arquitectura de comunicações.....	9
Figura 1.5 - Stack de comunicações das redes de campo.....	11
Figura 1.6 - Categorias de redes de campo.....	13
Figura 1.7 - Exemplos de soluções comerciais existentes.....	14
Figura 1.8 - Ambiente de validação da arquitectura GUARDS.....	19
Figura 2.1 - Conceito de confiança no funcionamento.....	27
Figura 2.2 - Classes elementares de falhas.....	30
Figura 2.3 - Classes combinadas de falhas.....	32
Figura 2.4 - Modos de avaria.....	33
Figura 2.5 - Relação causal entre falhas, erros e avarias.....	34
Figura 2.6 - Poder de modelação das várias técnicas de avaliação da confiança no funcionamento.....	46
Figura 2.7 - Alternância entre estados do sistema.....	48
Figura 2.8 - Evolução típica da taxa de avarias.....	49
Figura 2.9 - Exemplo da realização de um processo estocástico.....	54
Figura 2.10 - Relações e hierarquia entre as várias classes de processos de Markov.....	56
Figura 2.11 - Realização de um processo de renovação.....	57
Figura 2.12 - Propriedades do processo de Poisson.....	58
Figura 2.13 - Realização de um processo de Renovação de Markov.....	59
Figura 2.14 - Exemplo de uma cadeia de Markov Discreta.....	62
Figura 2.15 - Exemplo de uma cadeia de Markov Contínua.....	65
Figura 2.16 - Realização de um processo de Semi-Markov.....	67
Figura 2.17 - Realização de um processo Regenerativo de Markov.....	68
Figura 2.18 - Realização de um processo Generalizado de Semi-Markov.....	70
Figura 2.19 - Expansão por fases.....	71
Figura 2.20 - Conceito de função de recompensa.....	75

Figura 2.21 - Exemplo de uma rede de Petri.....	78
Figura 2.22 - Exemplo do disparo de uma transição temporizada.....	84
Figura 2.23 - Políticas de preempção.....	87
Figura 3.1 - Estrutura da rede.....	113
Figura 3.2 - Noção de deadline associada à transmissão de uma mensagem.....	116
Figura 3.3 - Aplicação do método dos estados de funcionamento críticos: sistemas não reparáveis.....	121
Figura 3.4 - Cadeia de Markov reduzida: sistemas não reparáveis.....	122
Figura 3.5 - Aplicação do método dos estados de funcionamento críticos: sistemas reparáveis.....	122
Figura 3.6 - Cadeia de Markov reduzida: sistemas reparáveis.....	123
Figura 3.7 - Factor de cobertura.....	125
Figura 3.8 - Cadeia de Markov associada ao modelo mR.....	128
Figura 3.9 - Cadeia de Markov associada ao modelo mD.....	130
Figura 3.10 - Cadeia de Markov associada ao modelo mRCS.....	133
Figura 3.11 - Cadeia de Markov associada ao modelo mRCC.....	134
Figura 3.12 - Cadeia de Markov associada ao modelo mRD.....	137
Figura 3.13 - Cadeia de Markov associada ao modelo mRDCC.....	139
Figura 3.14 - Modelo mRD vs. mR: indisponibilidade (cn).....	143
Figura 3.15 - Modelo mRD vs. mR: indisponibilidade (λn).....	143
Figura 3.16 - Modelo mRCC vs. mR: indisponibilidade (cn).....	144
Figura 3.17 - Modelo mRCC vs. mR: indisponibilidade (λcn).....	144
Figura 3.18 - Probabilidade de transmissão de uma trama com sucesso: duração variável (exponencial) vs. fixa.....	148
Figura 3.19 - Falhas que ocorrem previamente à transmissão.....	149
Figura 3.20 - Duração das falhas: variáveis vs. pontuais.....	151
Figura 3.21 - Relações entre parâmetros temporais.....	158
Figura 3.22 - Exemplo da determinação das probabilidades das sequências.....	163
Figura 3.23 - Exemplo sem sincronização inicial.....	165
Figura 3.24 - Exemplo com sincronização inicial.....	166
Figura 3.25 - Probabilidades das sequências das falhas.....	169
Figura 3.26 - Indisponibilidade: meio simples (não redundante).....	170
Figura 3.27 - Indisponibilidade: meio dual.....	170
Figura 3.28 - Indisponibilidade: meio dual – falhas em modo-comum.....	171
Figura 3.29 - Indisponibilidade comparativa.....	171
Figura 3.30 - MTTF: meio simples.....	172
Figura 3.31 - MTTF: meio dual.....	172
Figura 3.32 - MTTF: meio dual – falhas em modo-comum.....	173
Figura 3.33 - MTTF comparativo.....	173
Figura 3.34 - MTBF: meio simples.....	174

Figura 3.35 - MTBF: meio dual.....	174
Figura 3.36 - MTBF: meio dual – falhas em modo-comum.....	175
Figura 3.37 - MTBF comparativo.....	175
Figura 4.1 - Características estruturais e funcionais da rede campo.....	182
Figura 4.2 - Representação gráfica das SAN.....	195
Figura 4.3 - Modelo HW.....	201
Figura 4.4 - Aplicação da técnica de decomposição comportamental.....	204
Figura 4.5 - Arquitectura dos modelos HW e SW.....	208
Figura 4.6 - Componente tolerante a falhas ideal.....	209
Figura 4.7 - Composição de modelos HW e SW.....	209
Figura 4.8 - Modelo SW Local.....	210
Figura 4.9 - Modelo SW Global.....	212
Figura 4.10 - Exemplo das condições de avaria da rede.....	213
Figura 4.11 - Arquitectura do modelo CL.....	214
Figura 4.12 - Modelo CL.....	215
Figura 4.13 - Exemplo das condições de avaria da rede incluindo o cluster.....	216
Figura 4.14 - Arquitectura do modelo BUS.....	219
Figura 4.15 - Modelo BUS.....	222
Figura 4.16 - Modelo da reparação local.....	224
Figura 4.17 - Modelo da reparação global.....	225
Figura 4.18 - Versão alternativa do modelo global.....	229
Figura 4.19 - Avarias em modo-comum.....	231
Figura 4.20 - Modelo de avaliação integrado da rede de campo.....	234
Figura 5.1 - Arquitectura de comunicações do Profibus-DP.....	245
Figura 5.2 - Mecanismo de acesso ao meio.....	247
Figura 5.3 - Diagrama de estados do Profibus.....	257
Figura 5.4 - Passagem do token.....	258
Figura 5.5 - Trama do token.....	259
Figura 5.6 - Transmissão do token.....	262
Figura 5.7 - Rotação do token.....	265
Figura 5.8 - Mecanismo de transmissão de mensagens.....	267
Figura 5.9 - Gestão do anel.....	268
Figura 5.10 - Re-inicialização do anel.....	271
Figura 5.11 - Re-inicialização do mestre.....	272
Figura 5.12 - Transmissão de mensagens.....	274
Figura 5.13 - Sequência de mensagens.....	277
Figura 5.14 - Gestão dos escravos.....	278
Figura 5.15 - Modelo do escravo.....	280
Figura 5.16 - Modelo global.....	282

Figura 5.17 - Formato das tramas do Profibus.....	284
Figura 5.18 - Composição do modelo.....	288
Figura 6.1 - Arquitectura de comunicações do CAN.....	303
Figura 6.2 - CAN 2.0A: formato de uma trama de dados.....	304
Figura 6.3 - Mecanismo de acesso ao meio.....	305
Figura 6.4 - Trama de erro.....	307
Figura 6.5 - Trama de sobrecarga.....	308
Figura 6.6 - CANOpen: modelo dos dispositivos.....	309
Figura 6.7 - DeviceNet: modelo de objectos.....	310
Figura 6.8 - Ocorrência das falhas.....	316
Figura 6.9 - Modelo da ocorrência das falhas.....	318
Figura 6.10 - Modelo da transmissão de uma mensagem.....	321
Figura 6.11 - Modelo da sinalização dos erros.....	324

| Índice de Tabelas

Tabela 1.1 - Perfis e protocolos de acordo com o IEC 61158 e o IEC 61784.....	12
Tabela 1.2 - Níveis de integridade (SIL) de acordo com o IEC 61508.....	17
Tabela 2.1 - Comparação entre técnicas de avaliação da confiança no funcionamento....	45
Tabela 2.2 - Medidas da confiança no funcionamento.....	47
Tabela 2.3 - Coeficiente de variação.....	72
Tabela 3.1 - Terminologia associada às avarias de causa-comum.....	131
Tabela 4.1 - Exemplo do código da porta i_hw_f.....	202
Tabela 4.2 - Exemplo do código da porta o_cg.....	206
Tabela 4.3 - Exemplo do código da porta o_cg_2.....	212
Tabela 4.4 - Meio simples - Condição de avaria segura: > 1 nó.....	237
Tabela 4.5 - Meio simples - Condição de avaria segura: > 2 nós.....	237
Tabela 4.6 - Meio dual - Condição de avaria segura : > 1 nó.....	237
Tabela 4.7 - Meio dual - Condição de avaria segura: > 2 nós.....	237
Tabela 4.8 - Meio simples - Condição de avaria segura: > 1 nó – 2 Clusters c/ 2 nós....	238
Tabela 4.9 - Meio simples - Condição de avaria segura: > 2 nós - 2 Clusters c/ 2 nós....	238
Tabela 4.10 - Meio dual - Condição de avaria segura: > 1 nó - 2 Clusters c/ 2 nós.....	238
Tabela 4.11 - Meio dual - Condição de avaria segura: > 2 nós – 2 Clusters c/ 2 nós.....	238
Tabela 4.12 - Comparação de resultados.....	239
Tabela 5.1 - Parâmetros da transição t_TT.....	286
Tabela 5.2 - Parâmetros das transições t_e_da e t_e_sa.....	286
Tabela 5.3 - Estrutura da rede de campo.....	289
Tabela 5.4 - Parâmetros associados à transmissão das mensagens.....	290
Tabela 5.5 - Parâmetros associados à operação da rede.....	291
Tabela 5.6 - Tempo real necessário para executar 1 simulação.....	294
Tabela 5.7 - Tempo real de uma simulação completa.....	296
Tabela 6.1 - SDS: serviços da camada de aplicação.....	309
Tabela 6.2 - Conjunto de mensagens utilizado.....	327
Tabela 6.3 - Cenários de ocorrência das falhas.....	328
Tabela 6.4 - Tempo real necessário para executar 1 simulação.....	334

Tabela 6.5 - Tempo real de uma simulação completa.....335

Capítulo 1

Introdução

1.1 Contexto

A dinâmica dos mercados actuais exige cada vez mais às empresas a criação de produtos de melhor qualidade, possuindo maiores funcionalidades, incorporando as mais recentes inovações tecnológicas e cujos processos de desenvolvimento e de produção sejam realizados de forma cada vez mais rápida. Este desafio nas áreas da inovação e da competitividade obrigou as empresas a concentrar esforços na modernização tecnológica dos seus processos de fabrico, nomeadamente na automatização dos mesmos. Deste esforço integrado resultaram, para além de produtos mais competitivos, o desenvolvimento de soluções tecnológicas avançadas que à medida que se tornaram cada vez mais comuns e acessíveis passaram também a ser incorporadas nos próprios processos de fabrico.

Durante as últimas três décadas assistiu-se a uma evolução sem paralelo na área dos sistemas de controlo, nomeadamente ao nível dos respectivos processos de concepção, implementação e operação. Isto deveu-se, em grande parte, aos novos desenvolvimentos quer em áreas tecnológicas, tais como a micro-electrónica e as telecomunicações, quer em áreas associadas à gestão e à integração de sistemas, bem como ao desejo de disponibilizar aos utilizadores finais equipamentos com com maiores funcionalidades a custos mais reduzidos.

Introdução

Este desenvolvimento reflectiu-se também ao nível das comunicações industriais, através da substituição progressiva das tradicionais comunicações ponto-a-ponto pelas redes locais industriais (LANs – *Local Area Networks*). Embora inicialmente os motivos desta mudança estivessem relacionados com aspectos económicos, tais como a redução da cablagem e dos custos de manutenção, resultaram posteriormente em enormes vantagens ao nível da descentralização do controlo dos processos, na facilidade de instalação e configuração, na elevada flexibilidade de utilização e na melhoria do desempenho dos sistemas de controlo.

A crescente descentralização ao nível das funções de controlo, bem como o aumento da utilização em sistemas de controlo industriais de dispositivos inteligentes baseados em microprocessadores ou microcontroladores, criou as condições necessárias para o desenvolvimento e proliferação das redes de campo (*Fieldbuses*). Estas são um tipo específico de rede local industrial, com o objectivo de interligar controladores, sensores e actuadores, que realizam a interface directa com o processo industrial.

As redes de campo têm vindo a ser progressivamente utilizadas como elemento central e estruturante das arquitecturas de controlo de sistemas industriais, que incluem, entre os seus muitos requisitos, a necessidade de depositar confiança justificada no seu funcionamento. Ao nível dos processos industriais, nomeadamente nas indústrias de manufatura e de processos, para além dos custos decorrentes da ocorrência de avarias pode também estar em causa a segurança (*safety*) de pessoas ou do ambiente. A crescente importância destes sistemas, associada ao aumento da respectiva complexidade, implica a necessidade de se dispor de metodologias que permitam avaliar o respectivo comportamento em cenários de ocorrência de falhas (*faults*) ao nível dos seus componentes ou de avarias (*failures*) do próprio sistema (Fig. 1.1).

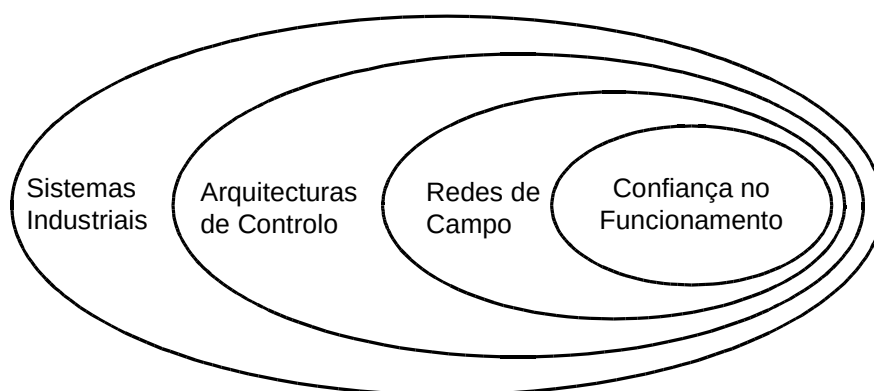


Figura 1.1 - Contexto da dissertação.

1.2 Sistema de Controlo

O funcionamento de forma correcta e segura de um processo industrial, de qualquer natureza, é assegurado por intermédio de um sistema de controlo apropriado. Independentemente da dimensão ou da complexidade do processo em causa, o respectivo sistema de controlo pode ser decomposto em três subsistemas com funções bem definidas: o processo controlado¹, o controlador e o operador humano [Kop97] (Fig. 1.2).

O controlador é um equipamento que interage com o seu ambiente através de duas interfaces com características distintas. A interface com o processo controlado é definida como a interface de instrumentação e a interface com o operador humano como a interface homem-máquina.

A interface de instrumentação consiste num conjunto de sensores e actuadores que transformam os sinais físicos do processo controlado em sinais com características apropriadas para serem utilizados pelo controlador, e vice-versa. A interface homem-máquina consiste num conjunto de dispositivos de entrada e saída, que permitem a interacção com um operador humano. Tipicamente, esta interacção realiza-se ao nível da definição de parâmetros do processo e da supervisão da respectiva evolução.

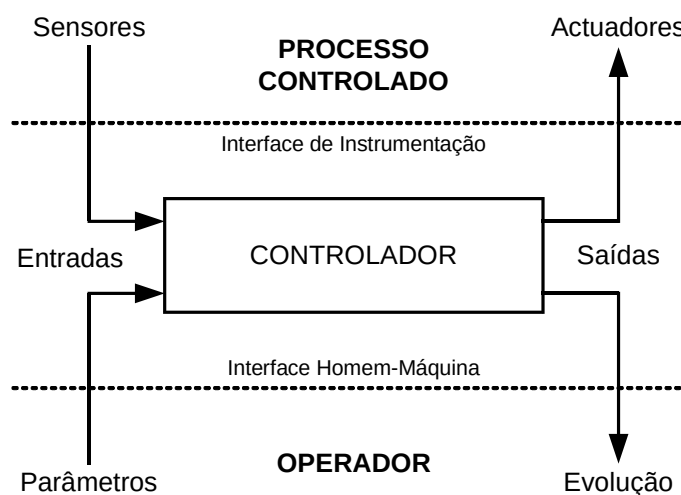


Figura 1.2 - Sistema de controlo.

A função do controlador é controlar a evolução do processo através da execução de um algoritmo de controlo adequado. A partir do processamento da informação

¹ Também referido como objecto controlado. Dado o contexto da discussão adoptou-se o termo processo.

Introdução

obtida, quer directamente do estado do processo através da interface de instrumentação, quer fornecida pelo operador humano através da interface homem-máquina, o algoritmo de controlo produz um conjunto de comandos que são enviados para o processo através da interface de instrumentação. Para realizar estas funções o controlador dispõe de uma estrutura funcional, baseada na utilização de equipamentos adequados ao processo em causa, que suporta a execução do algoritmo de controlo.

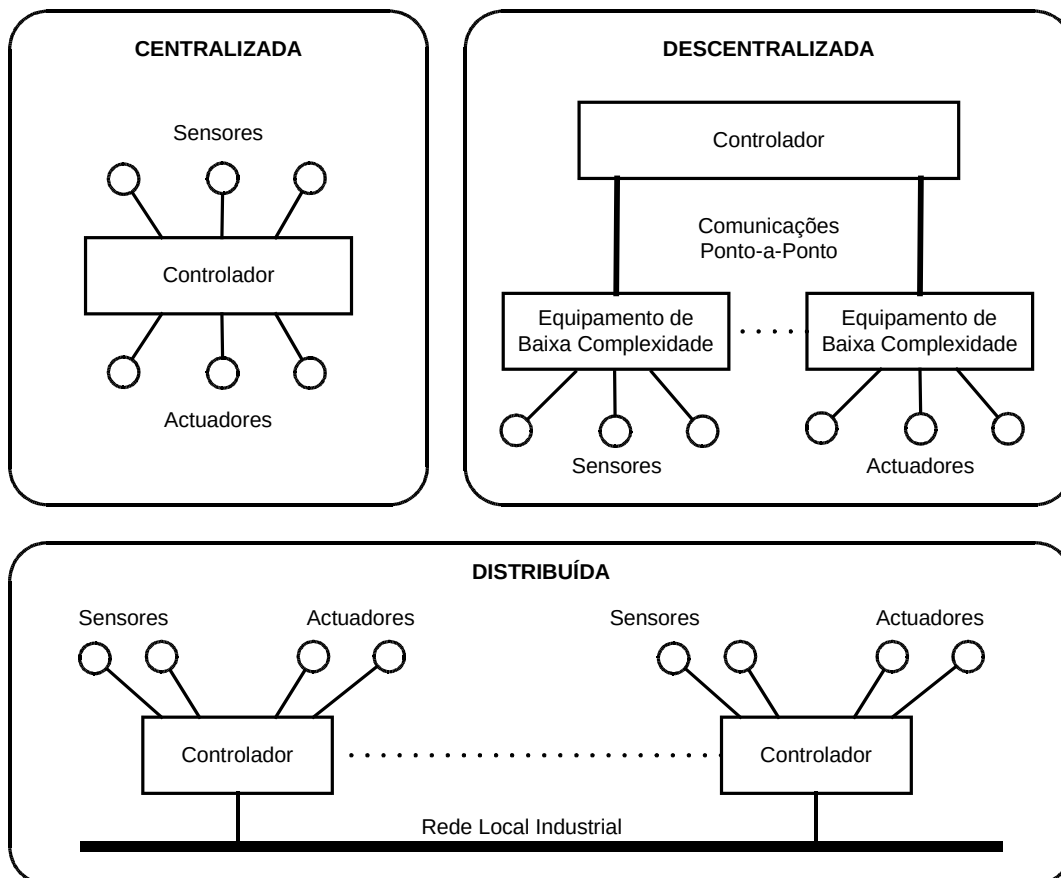


Figura 1.3 - Exemplos de arquitecturas de controlo.

Ao nível da estrutura funcional, estes sistemas podem ser classificados em três tipos de arquitecturas [Tov99] [Hef97] (Fig. 1.3):

- ♦ **Centralizadas.** O algoritmo de controlo é executado por um único equipamento;
- ♦ **Descentralizadas.** O algoritmo é executado num único equipamento, mas algumas tarefas de processamento mais simples (ex. condicionamento e aquisição de sinais) são executadas por outros equipamentos de menor complexidade. Isto implica a existência de uma estrutura de comunicações que permita a interacção e a cooperação entre os vários equipamentos (ex. comunicações série ponto-a-ponto);

- ♦ **Distribuídas.** O algoritmo de controlo encontra-se distribuído por vários equipamentos de complexidade e natureza distintas. Tal como nas arquitecturas descentralizadas, é também necessário dispor de uma estrutura de comunicações adequada, sendo esta, contudo, comparativamente muito mais complexa (ex. rede de campo).

1.3 Evolução das Tecnologias de Controlo

Do desenvolvimento dos primeiros dispositivos mecânicos, cujo emprego remonta aos primórdios da civilização, resultaram os primeiros sistemas de controlo [Gro00]. Estes sistemas eram muito rudimentares e estavam associados a processos muito simples, baseados na utilização de fontes de energia primárias (ex. água, vento, etc.), ou directamente do trabalho animal ou humano.

Este panorama sofreu poucas alterações até aos finais do século XVIII, a partir do qual se dá uma transformação radical com o desenvolvimento da primeira máquina a vapor, marcando assim o início da revolução industrial. Datam desta época os primeiros sistemas de controlo automatizados. Estes sistemas eram utilizados na automatização de algumas tarefas críticas e repetitivas dos processos industriais da época, sendo tipicamente de baixa complexidade e desenvolvidos especificamente para cada tarefa. As funções de controlo eram implementadas através de dispositivos mecânicos que, devido à natureza dos mesmos, impunham muitas limitações ao nível das capacidades de controlo do sistema.

A partir do final do século XIX as formas de energia herdadas da revolução industrial – o carvão, o gás e o vapor – foram sendo progressivamente substituídas pela energia eléctrica. Nas primeiras décadas do século XX, esta transformação tecnológica levou à substituição progressiva dos dispositivos mecânicos por relés eléctricos. Estes dispositivos permitiram o desenvolvimento de funções de controlo mais complexas e sofisticadas, representando desta forma uma alternativa viável face aos dispositivos mecânicos, especialmente para a automação de pequenas máquinas de baixa complexidade. Ao nível dos processos mais complexos, foram preferencialmente utilizadas as tecnologias electromecânicas, tais como a pneumática e a hidráulica, contudo das fortes restrições de utilização impostas pelas mesmas resultou em soluções algo limitadas.

O desenvolvimento do transístor a partir dos finais dos anos 40 e posteriormente do circuito integrado nos anos 60, proporcionou o desenvolvimento de uma nova geração de sistemas de controlo. Em comparação com os relés, esta tecnologia de lógica discreta permitiu desenvolver sistemas mais pequenos, mais rápidos e mais

complexos. No entanto, e tal como na tecnologia de relés, a implementação do algoritmo de controlo era efectuada através da interligação dos componentes eléctricos/electrónicos. Embora sejam de simples implementação, qualquer alteração ao seu funcionamento implica um rearranjo das ligações, com todos os inconvenientes daí resultantes.

O advento do computador no início anos 60, permitiu que estes passassem também a ser utilizados para implementar funções de controlo. O termo *controlo directo digital* (DCC - *Direct Digital Control*) foi utilizado na época para enfatizar o facto do controlo do processo ser realizado directamente pelo computador. O facto de serem programáveis proporcionou-lhes uma esmagadora vantagem em comparação com as tecnologias de lógica discreta utilizadas até ao momento. Um único equipamento (computador), concentra em si quer as tarefas do controlador, quer as interfaces de instrumentação e de homem-máquina. No caso da interface de instrumentação os sensores e actuadores são tipicamente ligados ao controlador através de ligações ponto-a-ponto analógicas (ex. anel de corrente). Em paralelo com este processo, tem também início nos finais do anos 60 o desenvolvimento de máquinas de controlo numérico e de robots industriais.

As arquitecturas de controlo desenvolvidas até a esta época são essencialmente centralizadas. Contudo, as crescentes exigências da indústria, conjugadas com o desenvolvimento do microprocessador no início dos anos 70, permitiram uma evolução para as primeiras arquitecturas descentralizadas. Esta evolução efectuou-se segundo duas perspectivas [Ram03]: uma ao nível das indústrias de processos, com o desenvolvimento dos sistemas de controlo distribuídos (DCS - *Distributed Control Systems*) com o objectivo de interligar hierarquicamente os equipamentos de controlo de menor complexidade (ex. PIDs, instrumentação distribuída) aos de maior complexidade (ex. mini-computadores); a outra ao nível das indústrias de manufactura, onde o autómato programável (PLC – *Programmable Logic Controller*), cujo desenvolvimento se deu no início dos anos 70 [Hug01], foi utilizado como elemento central das arquitecturas de controlo. Em ambas as perspectivas, a interligação entre equipamentos era tipicamente realizada quer através ligações ponto-a-ponto analógicas, quer de ligações digitais, utilizando neste último caso protocolos de comunicação proprietários. Embora esta evolução tenha permitido o desenvolvimento de sistemas de controlo cada vez mais complexos, durante a primeira década da sua utilização, as arquitecturas de controlo continuaram a ser caracterizadas por uma estrutura tipicamente centralizada, e só mais tarde se registou uma evolução para soluções do tipo descentralizado.

Entre meados dos anos 80 e inícios dos 90, assistiu-se a uma alteração da estrutura das arquitecturas de controlo através da crescente utilização de redes de comunicação industriais para interligar os equipamentos de controlo. Esta evolução tecnológica foi o ponto de partida das primeiras arquitecturas de controlo

descentralizadas baseadas numa estrutura de comunicações digital [PD88]. Estas arquitecturas, embora mais complexas, permitiram obter um importante conjunto de vantagens das quais se destacam: menores custos, melhores desempenhos, instalação e manutenção mais simples, modularidade, facilidade na detecção de erros e avarias, etc. Na sequência desta evolução, surge também nesta época o conceito de CIM (*Computer Integrated Manufacturing*) [DSP88].

Embora esta evolução abrisse um conjunto enorme de perspectivas ao nível da integração de equipamentos situados em diferentes níveis de controlo, na prática isto acabou por não se verificar devido ao desenvolvimento de muitas soluções proprietárias. Estas acabaram por limitar as potencialidades das arquitecturas de controlo, nomeadamente ao nível da integração e da interoperabilidade entre equipamentos. Este problema colocou-se de forma especialmente grave ao nível das redes de campo, com os diferentes fabricantes a tentar impor as respectivas soluções como o standard a utilizar (Profibus, WorldFIP, DeviceNet, Interbus-S, AS-I, etc.). Este processo terminou apenas recentemente através da adopção de uma solução de compromisso entre as várias propostas existentes [FS02]. Ao nível das arquitecturas de controlo registou-se uma tendência para adoptar soluções distribuídas, devido quer à possibilidade de dispor de equipamentos com maiores capacidades de processamento, quer através do desenvolvimento de arquitecturas de comunicação que suportam elevados níveis de integração entre equipamentos.

Entre os finais dos anos 90 e início da corrente década verificou-se um fenómeno de migração de tecnologias de uso geral para a área das comunicações industriais. O caso mais paradigmático deu-se com a utilização da rede *Ethernet* em ambientes industriais (*Industrial Ethernet*). Este processo teve um profundo impacto na estrutura das comunicações industriais, afectando todos níveis de controlo desde os superiores [Dec01] [Ida02] [Pro03] [SJO04], até aos inferiores [Eth01] [Mac01] [Vit01] [Mod02] [FFS+02]. Esta migração deveu-se a vários factores, tais como: existência de soluções de *hardware* de baixo custo e de elevado desempenho, protocolos de comunicação abertos e a disponibilização de plataformas de *software* para o desenvolvimento integrado de aplicações distribuídas. No entanto, presentemente este processo parece estar a dar origem aos mesmos problemas que se verificaram inicialmente com as redes de campo, nomeadamente à profusão, e correspondente incompatibilidade, de soluções proprietárias [FS04]. Associado ainda a este processo de migração saliente-se também a crescente utilização de redes de comunicações sem fios (IEEE 802.11 e IEEE 802.15) em ambientes industriais [DDH01] [Hod02] [Alv03] [Wil04].

Embora a introdução das redes industriais viesse resolver o problema da integração horizontal de equipamentos (situados no mesmo nível de controlo), a integração vertical (entre níveis de controlo) foi sempre um problema em aberto. As soluções apontadas inicialmente passavam pela utilização de equipamentos dedicados

(*gateways*) ou pelo desenvolvimento de *software* específico que implementava as tarefas de mapear os serviços de comunicação das diferentes redes. Como estas soluções eram normalmente caras e complexas, surgiu no final dos anos 80 a ideia de desenvolver uma arquitectura de comunicações aberta baseada na estrutura do modelo OSI (*Open Systems Interconnection*) [ISO94]. Exemplos destas soluções foram o MAP (*Manufacturing Automation Protocol*) [MAP85] [DDP88] e o MMS (*Manufacturing Message Specification*) [IS003] [Sch04]. No entanto o seu sucesso acabou por ser limitado devido quer à falta de suporte tecnológico adequado, quer à complexidade das soluções propostas.

Em paralelo, nas indústrias de processos a utilização de SCADAs (*Supervisory Control and Data Aquisition*) foi vista como uma alternativa mais simples e razoavelmente eficaz de realizar a integração vertical [Boy99]. Contudo, e devido quer à ausência de adequação dos SCADAs para o problema em causa, quer à proliferação de equipamentos de controlo com características muito diversas, a utilização desta tecnologia acabou por resultar em soluções bastante limitadas.

Nos finais dos anos 90 e devido às crescentes exigências de integração com as aplicações de nível intermédio e superior, nomeadamente os ERP (*Enterprise Resource Planning*) e os MES (*Manufacturing Execution Systems*) [BPB03] [VW04], foram desenvolvidas um conjunto tecnologias de *software* baseadas em plataformas de objectos distribuídos, que disponibilizam uma infraestrutura, ao nível dos serviços de comunicações, que permitiam desenvolver de forma eficaz os conceitos de integração vertical e horizontal. São exemplos destas tecnologias o CORBA (*Common Object Request Broker Architecture*) [OMG98] com uma gama alargada de domínios de aplicação [PAP04], e o OPC (*OLE for Process Control*) [OPC98] [IL02] especialmente desenvolvido para satisfazer os requisitos no domínio das aplicações industriais. Este processo apresenta actualmente uma grande dinâmica, com destaque para o desenvolvimento de standards com base em arquitecturas de objectos distribuídos especialmente vocacionados para as necessidades dos ambientes industriais: IEC 61499 (*Function Blocks for Industrial-Process Measurement and Control Systems*) [IEC00] [Lew01] e o IEC 61804 (*Function Blocks for Process Control*) [IEC03c] [RD01]. Como resultado deste trabalho as arquitecturas de comunicação mais recentes já incorporam muitas estas funcionalidades, das quais se destacam: o CIP (*Common Industrial Protocol*) [Eth01], a IDA (*Interface for Distributed Automation*) [Ida02], o HSE (*HSE Fieldbus Foundation*) [Mac01] e o PROFINet [Pro03].

A última etapa desta evolução está centrada na utilização das tecnologias desenvolvidas para Web: UDP/TCP/IP, SMTP, HTTP, Proxys, XML, Java, Jini, etc., para o desenvolvimento aplicações industriais [Ban01]. A utilização destas tecnologias, para além de estarem largamente difundidas e serem de custo

reduzido, vai permitir obter níveis de integração superiores, nomeadamente ao nível domínios de aplicação externos ao ambiente industrial [LPS02] [LSD03].

1.4 Arquitectura das Comunicações Industriais

Ao nível dos sistemas de controlo a integração implica a necessidade de cooperação e interacção entre os vários subsistemas do mesmo. Isto significa transferência, armazenamento e processamento de informação em ambientes com características heterogéneas, o que por sua vez obriga à necessidade de dispor de uma infraestrutura de comunicações adequada. As redes locais industriais, não sendo a solução para este problema, são contudo uma parte integrante e essencial dessa solução.

Os fluxos de informação existentes num ambiente industrial possuem características de tal forma distintos que não é possível dispor de uma única rede capaz de satisfazer todas as necessidades de comunicação. Desta forma, a alternativa é dispor de um conjunto de redes que no seu conjunto sejam capazes de satisfazer a totalidades dessas necessidades. Num sistema automatizado as actividades relacionadas com o controlo do processo industrial podem ser estruturadas num modelo hierárquico caracterizado por fluxos de informação verticais, entre entidades de níveis hierárquicos adjacentes, e por fluxos de informação horizontais entre entidades situadas no mesmo nível hierárquico. Como estas actividades estão intimamente associadas à estrutura de comunicações que lhes serve de suporte, surge de forma natural também a adopção de um modelo hierárquico para a arquitectura de comunicações [Pim90] [RNS93] (Fig. 1.4).

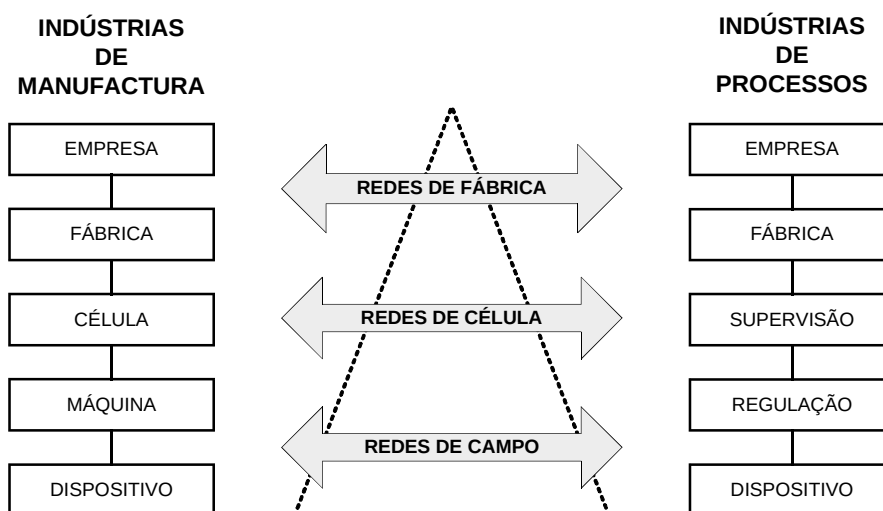


Figura 1.4 - Relação entre os níveis de controlo e arquitectura de comunicações.

Embora os modelos hierárquicos para a estrutura de controlo possam variar em número de níveis, tipicamente entre 4 e 6 [RNS93] [Pim90] [KP95] [Gro00], ao nível da arquitectura de comunicações é usual identificar três níveis distintos: *fábrica*, *célula* e *campo*. Para cada um destes níveis vários tipos de redes foram desenvolvidas, cada uma possuindo diferentes objectivos, protocolos, capacidades e complexidade:

♦ Fábrica

Cobrem as necessidades dos níveis superiores. As principais actividades encontradas a este nível são o planeamento da produção, de processos e de materiais, e as áreas de engenharia, financeira e comercial. O fluxo de informações descendente centra-se essencialmente nas ordens de fabrico, e nas informações associadas ao seu escalonamento. No sentido ascendente, circulam informações relativas ao estado das ordens de fabrico, à qualidade do processo de produtivo e a pedidos de aquisição de materiais/recursos. Este nível é caracterizado por um elevado fluxo de informação horizontal entre e dentro dos vários subsistemas existentes, sem contudo requisitos temporais críticos;

♦ Célula

Cobrem as necessidades dos níveis intermédios. Uma célula agrupa um conjunto de equipamentos que cooperam para a execução de uma determinada tarefa. As principais actividades encontradas a este nível são o escalonamento, o sequenciamento e a execução de tarefas. Outras actividades também executadas têm a ver com a compilação da informação relativa à qualidade da produção e ao desempenho dos equipamentos que constituem a célula. A informação que circula do nível célula para os níveis descendentes inclui ordens de execução de operações ou programas de controlo, no caso de equipamentos programáveis. Em sentido ascendente a informação disponibilizada diz respeito à evolução das operações executadas e aos resultados dessas mesmas operações. Este nível é caracterizado por fluxos de informação de volume intermédio e com requisitos temporais exigentes, que em muitos dos casos podem ser críticos;

♦ Campo

Cobrem as necessidades dos níveis mais baixos. As principais actividades encontradas a este nível estão relacionadas com o controlo directo do processo industrial, nomeadamente a execução de algoritmos de controlo por equipamentos que actuam fisicamente sobre os materiais/produtos a operar. A interface com o processo é realizada por intermédio de sensores e actuadores muitos deles dotados já de capacidades de processamento complexas (*smart sensors*) [Kop01]. Este nível é caracterizado por fluxos de informação de pequeno volume e com requisitos temporais críticos.

1.5 Redes de Campo

As redes de campo (*Fieldbuses*) foram inicialmente desenvolvidas com o objectivo de satisfazer os requisitos de comunicação dos níveis mais baixos das arquitecturas de controlo industriais. Entre estes, destacam-se pela sua importância os seguintes [Tho98] [Tho99] [Dec01]:

- ♦ Capacidade de transferir pequenos volumes de informação de forma eficiente;
- ♦ Suportar quer tráfego periódico (ex. amostragem de dados), quer aperiódico (ex. eventos) com tempos de resposta majorados. Existem assim requisitos de tempo-real associados às comunicações;
- ♦ Capacidade de operar em ambientes industriais típicos sujeitos a interferências electromagnéticas (EMI – *Electromagnetic Interference*), vibrações, corrosão, poeiras, humidade, etc.;
- ♦ Garantir um nível adequado de confiança no funcionamento, nomeadamente no que diz respeito à fiabilidade, disponibilidade e segurança (*safety*);
- ♦ Baixo custo de aquisição, instalação, operação e manutenção.

Estes requisitos traduziram-se na adopção de uma *stack*² de comunicações organizada de acordo com o modelo OSI [ISO94], mas utilizando apenas um modelo compactado de 3 camadas: física, ligação de dados (DLL – *Data Link Layer*) e aplicação (Fig. 1.5). Embora esta última seja denominada de aplicação, acaba por incorporar um conjunto de funcionalidades definidas nas restantes 4 camadas não utilizadas.

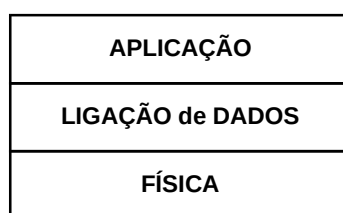


Figura 1.5 - *Stack de comunicações das redes de campo.*

Para cada uma das camadas podem ser definidos múltiplos serviços, e protocolos de comunicação com características muito diversas. A escolha destes elementos resulta essencialmente dos objectivos originais definidos pelos fabricantes das redes de campo, que de forma sucinta podem ser expressos da seguintes forma [Tho99]:

- ♦ A rede de campo é considerada apenas como uma forma de simplificar as ligações físicas entre os vários dispositivos;

2 Optou-se por utilizar a terminologia em Inglês por se julgar mais adequada ao contexto.

Introdução

- ♦ A rede de campo é considerada a *coluna vertebral (backbone)* de um sistema distribuído e de tempo-real.

A diferença entre estas duas perspectivas foi uma das razões que levaram à proliferação de soluções para as redes de campo. Outras razões estão relacionadas com a ausência de um standard internacional único e genérico. Desde o início dos anos 80 que mais de 50 marcas ou standards diferentes surgiram, à medida que as necessidades de soluções baseadas em redes de campo se faziam sentir na área industrial.

IEC 61784 Perfil	IEC 61158 Protocolos - Camadas			Standard CENELEC	Nome Comercial
	Física	Ligação de Dados	Aplicação		
CPF-1/1	Tipo 1	Tipo 1	Tipo 9	EN-50170-A1	Foundation Fieldbus (H1)
CPF-1/2	Ethernet	TCP/UDP/IP	Tipo 5	-	Foundation Fieldbus (HSE)
CPF-1/3	Tipo 1	Tipo 1	Tipo 9	EN-50170-A1	Foundation Fieldbus (H2)
CPF-2/1	Tipo 2	Tipo 2	Tipo 2	EN-50170-A3	ControlNet
CPF-2/2	Ethernet	TCP/UDP/IP	Tipo 2	-	EtherNet/IP
CPF-3/1	Tipo 3	Tipo 3	Tipo 3	EN-50254-3	Profibus-DP
CPF-3/2	Tipo 1	Tipo 3	Tipo 3	EN-50170-A2	Profibus-PA
CPF-3/3	Ethernet	TCP/UDP/IP	Tipo 10	-	PROFINet
CPF-4/1	Tipo 4	Tipo 4	Tipo 4	EN-50170-1	P-Net RS-485
CPF-4/1	Tipo 4	Tipo 4	Tipo 4	EN-50170-1	P-Net RS-232
CPF-5/1	Tipo 1	Tipo 7	Tipo 7	EN-50170-3	WorldFIP (MPS, MCS)
CPF-5/2	Tipo 1	Tipo 7	Tipo 7	EN-50170-3	WorldFIP (MPS, MCS, subMMS)
CPF-5/3	Tipo 1	Tipo 7	Tipo 7	EN-50170-3	WorldFIP (MPS)
CPF-6/1	Tipo 8	Tipo 8	Tipo 8	EN-50170-2	INTERBUS
CPF-6/2	Tipo 8	Tipo 8	Tipo 8	EN-50170-2	INTERBUS TCP/IP
CPF-6/3	Tipo 8	Tipo 8	Tipo 8	EN-50170-2	INTERBUS Subset
CPF-7/1	Tipo 6	Tipo 6	-	-	Swiftnet <i>transport</i>
CPF-7/2	Tipo 6	Tipo 6	Tipo 6	-	Swiftnet <i>full stack</i>

Tabela 1.1 - Perfis e protocolos de acordo com o IEC 61158 e o IEC 61784.

No sentido de encontrar uma solução para este problema, foi desencadeado no início dos anos 90 um processo de especificação de um standard que tinha como objectivo o desenvolvimento de uma solução unificadora. Este processo arrastou-se durante mais de 10 anos [Fel02] [FS02] [Lev01] [Hef97] terminando em 2003

com a adopção de uma solução de compromisso da qual resultaram dois standards internacionais: o IEC 61158 (*Digital Data Communications for Measurement and Control - Fieldbus for use in Industrial Control Systems*) [IEC03a] e o IEC 61784 (*Digital Data Communications for Measurement and Control - Profile Sets for Continuous and Discrete Manufacturing Relative to Fieldbus use in Industrial Control Systems*) [IEC03b], constituídos por um conjunto de perfis de comunicação, aos quais acabaram por corresponder as soluções comerciais mais importantes existentes à data da sua publicação (Tab. 1.1).

À medida que o processo de standardização foi estabilizando, o desenvolvimento focou-se na definição de uma quarta camada, denominada camada de utilizador (*User Layer*). O seu objectivo é disponibilizar ao utilizador uma abordagem integrada no desenvolvimento das aplicações, nomeadamente através da definição de blocos funcionais (*function blocks*) [Eth01] [Mac01] [Pro03], linguagens de descrição dos dispositivos [KM04], interoperabilidade e métricas da qualidade de serviço [BSI00].

Também em relação ao seu posicionamento original referente aos níveis de controlo das aplicações industriais, as redes de campo sofreram uma evolução, passando também a ser utilizadas presentemente como redes de célula. A própria terminologia tem evoluído, através da definição de um conjunto de subcategorias para as redes de campo (Fig. 1.6). Neste sentido, o termo original *Fieldbus* tem sido utilizado para designar as redes (de campo) que estão mais próximas do conceito de rede de célula (ex. Profibus-DP, WorldFIP), e o termo *Sensorbus* para designar as redes mais básicas e mais próximas do conceito original de rede de campo (ex. AS-I, Interbus-S), enquanto que o termo *Devicebus* é utilizado para designar as que estão num plano de aplicação intermédio (ex. DeviceNet, FF-H1). Contudo e por uma questão de simplificação de linguagem, utilizar-se-à nesta dissertação apenas o termo rede de campo para representar todas as subcategorias acima definidas.

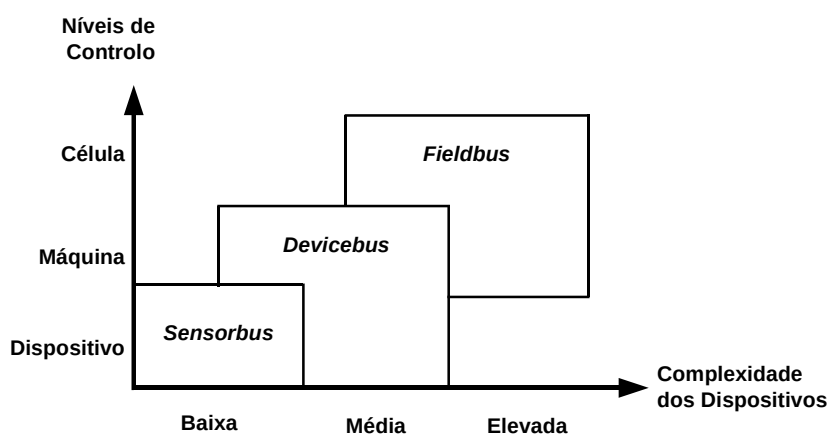


Figura 1.6 - Categorias de redes de campo.

Introdução

Da discussão realizada nestas secções é possível concluir que as redes de campo podem ser utilizadas em vários níveis de controlo, representado actualmente a estrutura sobre qual assentam as arquitecturas de controlo dos níveis intermédios e inferiores dos processos industriais [PD88] [KP95] [CDV95] [Hef97] [SM97] [DV98] [Pat98] [Sch98] [Tho98] [Tho99] [Fel02] [FS02] [IEB04] (Fig. 1.7).

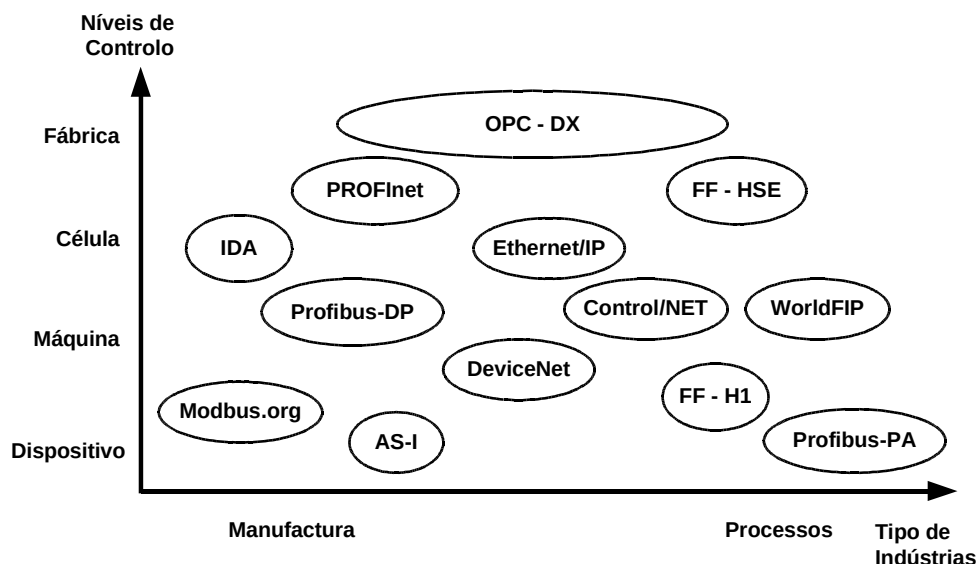


Figura 1.7 - Exemplos de soluções comerciais existentes.

1.6 Motivação

Durante muitos anos os objectivos impostos durante as várias fases do ciclo de vida de um sistema de controlo, concentraram-se quase exclusivamente na minimização dos custos de desenvolvimento e de operação, e na melhoria do desempenho das respectivas funções de controlo. Com a excepção de domínios de aplicação específicos, tais como as indústrias aeronáutica, espacial, ferroviária e nuclear, as questões relacionadas com a confiança no funcionamento, que englobam aspectos tais como a fiabilidade, a disponibilidade e a segurança (*safety*) [Lap95], não eram suficientemente consideradas. Para esta abordagem contribuíram vários factores, dos quais se destacam:

- ♦ O carácter multi-disciplinar do problema obrigava a um esforço na conjugação de conhecimentos provenientes de diferentes domínios científicos;
- ♦ Dificuldade em utilizar uma abordagem integrada que permitisse incluir, durante as várias fases do ciclo de vida do sistema, os diferentes aspectos associados à confiança no funcionamento;

- ♦ Indefinição ao nível da especificação de critérios qualitativos e quantitativos que permitissem definir um processo de validação do sistema na perspectiva da confiança no funcionamento;
- ♦ Baixo nível de complexidade dos sistemas, minimizando assim a ocorrência de avarias;
- ♦ Domínios de aplicação em que as consequências das avarias resultavam em impactos reduzidos, quer ao nível do processo, quer em pessoas ou no ambiente;
- ♦ Ausência de legislação e regulamentação adequadas;
- ♦ Falta de consciencialização para este tipo de questões ao nível do pessoal encarregue do desenvolvimento e operação do sistema.

A evolução dos sistemas de controlo, quer ao nível da complexidade, quer nos domínios de aplicação, levou obrigatoriamente a uma inversão desta atitude. Numa primeira fase foram as questões económicas que estiveram na base desta mudança. As avarias do sistema de controlo resultavam em ausência de produção, que por sua vez se reflectia no custo do produto final. Desta forma, era necessário que estes sistemas fossem dotados de funcionalidades que lhes permitissem garantir níveis elevados de fiabilidade e disponibilidade por forma a minimizar estes custos. Mais tarde, e correspondendo também a uma evolução sociológica, as questões relacionadas com a possibilidade das avarias do sistema de controlo poderem causar danos, com diferentes níveis de gravidade, quer em pessoas, quer ao ambiente, passaram também a ser equacionadas [Sto96] [Gob98].

Hoje em dia é comum de encontrar sistemas de controlo em domínios de aplicação em que a confiança no funcionamento é encarada como um requisito essencial ao funcionamento dos mesmos. Estes vão desde os sistemas embebidos (geralmente imperceptíveis ao comum dos utilizadores) tais como os empregues em aplicações do tipo *drive-by-wire* em automóveis ou *fly-by-wire* em aviões, até sistemas mais perceptíveis tais como: controlo de tráfego, produção de energia eléctrica, extracção e produção de combustíveis, operação de máquinas ferramentas, controlo da qualidade ambiental, sistemas de apoio de vida, etc.

As indústrias de manufactura e de processos, pelas suas características, representam uma das áreas onde a confiança no funcionamento dos sistemas de controlo deve considerada como uma questão importante, quer ao nível económico, quer ao nível da segurança das pessoas e do ambiente. Os primeiros esforços no sentido de incorporar estes requisitos nestas áreas, centraram-se no desenvolvimento de standards regulamentadores, com especial ênfase para as máquinas ferramentas. O objectivo principal destes standards, centrou-se na especificação das características que os equipamentos utilizados deveriam possuir, em função do tipo de ambiente onde tinham que operar [WI00] [SIG04] [WES04] [HON04].

Esta regulamentação aplicava-se de forma geral aos equipamentos utilizados no processo, sem contudo focar as características específicas dos respectivos sistemas de controlo. Durante muitas décadas, as abordagens utilizadas no sentido de garantir a confiança no funcionamento deste tipo de sistemas, foram essencialmente baseadas nas características particulares do processo controlado, ou na experiência prática adquirida pelas equipas encarregues do seu desenvolvimento. Contudo, o aumento da dependência da operacionalidade dos sistemas de controlo, aliada à sua crescente complexidade e combinada com o respectivo emprego em domínios de aplicação de segurança crítica (*safety-critical systems*) obrigou a uma mudança nesta atitude.

Neste sentido, foram desenvolvidos esforços pela comunidade científica e industrial com o objectivo de definir um conjunto de standards que permitissem caracterizar os requisitos necessários, ao nível da confiança no funcionamento, dos sistemas de controlo industriais. Estes focaram a sua atenção particularmente nas funções de segurança do sistemas de controlo (*safety functions*), salientando-se o EN954 e ISO 13849 (*Safety of Machinery – Safety Related Parts of Control Systems*) [ISO99a] para aplicação no domínio das máquinas ferramentas, o IEC 61508 (*Functional Safety of Electrical / Electronic / Programmable Electronic Safety-Related Systems*) [IEC04a], genérico e aplicável em todo o tipo de domínios que incorporem sistemas eléctricos, electrónicos ou programáveis, e o IEC 61511 (*Functional Safety – Safety Instrumented Systems for the Process Sector*) [IEC04b] mais vocacionado para as indústrias de processos.

A aceitação dos standards como elementos regulamentadores obrigou as empresas a certificar os sistemas por si produzidos ou utilizados em domínios de aplicação cobertos pelos mesmos. Este processo de certificação é complexo, envolvendo a utilização de diferentes metodologias ao longo do ciclo de vida do sistema. A abordagem a este problema é geralmente orientada por um objectivo global, que privilegia o resultado final por oposição ao equipamento que é utilizado para o atingir.

De forma geral, a aplicação destes standards passa por uma avaliação prévia dos perigos (*hazards*) envolvidos e da aceitabilidade dos riscos durante a operação do sistema [Sto96]. Da combinação de ambos os aspectos resulta um indicador do nível da integridade de segurança do sistema (SIL - *Safety Integrity Level* [IEC04a]) (Tab. 1.2). Este nível pode ser interpretado como a probabilidade do sistema que assegure a segurança do processo (*safety-related system*) actuar de forma correcta na execução de funções de segurança (*safety functions*) dentro das condições de funcionamento especificadas durante um certo período de tempo.

A definição de um nível de integridade, implica a adopção de um conjunto de procedimentos, metodologias, tecnologias e arquitecturas para a implementação

das funções de segurança do sistema de controlo [Sto96] [IEC04a]. Resulta desta abordagem a necessidade de utilizar técnicas que permitam validar, quer qualitativamente, quer quantitativamente, o comportamento do sistema quando sujeito a situações de falhas ao nível dos seus componentes ou de avarias do próprio.

Níveis de Integridade de Segurança - SIL	Modo de Operação Probabilidade de avaria	
	Baixa utilização (<i>on demand</i>)	Elevada utilização (/hora)
SIL 4	$\geq 10^{-5} \dots < 10^{-4}$	$\geq 10^{-9} \dots < 10^{-8}$
SIL 3	$\geq 10^{-4} \dots < 10^{-3}$	$\geq 10^{-8} \dots < 10^{-7}$
SIL 2	$\geq 10^{-3} \dots < 10^{-2}$	$\geq 10^{-7} \dots < 10^{-6}$
SIL 1	$\geq 10^{-2} \dots < 10^{-1}$	$\geq 10^{-6} \dots < 10^{-5}$

Tabela 1.2 - Níveis de integridade (SIL) de acordo com o IEC 61508.

Representando as redes de campo a estrutura sobre qual assentam as arquitecturas de controlo dos níveis intermédios e inferiores dos processos industriais, implica que a confiança no funcionamento destas mesmas arquitecturas está intimamente ligada à confiança no funcionamento das próprias redes de campo.

Desde cedo que os fabricantes das redes de campo se aperceberam deste aspecto, e procuraram dotá-las de um conjunto de funcionalidades no sentido de melhorar a sua confiança no funcionamento, entre as quais se destacam:

- ♦ Soluções tecnológicas ao nível do *hardware* com objectivo de minimizar os efeitos do ambiente industrial sobre os equipamentos: interferências electromagnéticas, humidade, temperatura, etc;
- ♦ Soluções intrinsecamente seguras (*intrinsically safe*) [Joh99] para suportar ambientes onde existem perigos de explosão e/ou incêndio;
- ♦ Garantir uma elevada capacidade de detecção de erros de transmissão;
- ♦ Detecção e recuperação de avarias permanentes e transitórias ao nível dos equipamentos;
- ♦ Detecção avarias temporais sobre dados transmitidos;
- ♦ Redundância de equipamentos e dos meios de transmissão;
- ♦ Equipamentos que suportem modos de avaria seguros (*fail safe*).

Para além das funções de controlo tradicionais, tem surgido ultimamente interesse em utilizar as redes de campo como sistemas de suporte a funções de segurança (*safety functions*), nomeadamente através da substituição dos sistemas de protecção electromecânicos tradicionais [Ne97] [Pig00] [WI00] [JH02] [CEE03] [Pig04]. A

crescente importância deste tipo de aplicações levou recentemente à criação de um grupo de trabalho específico (WG12) dentro comitê de trabalho IEC/SC65C encarregue da definição de um standard ao nível das redes de campo denominadas *seguras* (*Safety Networks*) [FS04]. Em paralelo, os fabricantes têm vindo a desenvolver, ou adaptar, algumas das soluções comerciais já existentes para este novo domínio de aplicação, das quais se destacam:

- ♦ Soluções especialmente desenvolvidas para este domínio: Safety Bus [Saf04];
- ♦ Demonstração através de um processo de validação baseado em aplicações reais que as soluções existentes já satisfazem os requisitos necessários (*proven-by-use*): WolrdFIP [FNS01];
- ♦ Considerar que as soluções existentes são não-seguras (*unsafe*), e dotá-las de mecanismos adicionais ao nível da arquitectura de comunicações, sem contudo alterar quer o *hardware* quer o *software* existentes, como o objectivo de garantir maiores níveis de integridade [MK03]. As soluções existentes são baseadas no conceito de *black channel* desenvolvido a partir do IEC 62280-1 (*Railway Applications – Communications, Signalling and Processing Systems – Part 1: Safety Related Communication in Closed Transmission Systems*) [IEC02] [Neu03], sendo presentemente as mais relevantes: o *PROFisafe* [Pro02], o *DeviceNet/Safety* [SVV+00], o *Ethernet/IP/Safety* [Odv03], o *AS-Interface* [Ver03] e *Interbus Safety* [Int03].

Da discussão anterior fica patente que as redes de campo já disponibilizam um conjunto de funcionalidades que permitem a sua utilização em domínios de aplicação onde a confiança no funcionamento é um dos requisitos necessários. No entanto, também se verifica que estas funcionalidades foram desenvolvidas sob a forma de componentes específicos ou de subsistemas com características particulares. Embora da perspectiva da confiança no funcionamento estas funcionalidades possam ser avaliadas de forma individual, o utilizador final está obviamente interessado no comportamento global do sistema.

A validação completa de um sistema com estas características é uma tarefa muito complexa. Este processo é naturalmente multi-disciplinar, envolvendo um conjunto alargado de domínios do conhecimento, tais como: *hardware*, *software*, comunicações industriais, sistemas de tempo-real, tolerância a falhas, teoria de controlo, processos estocásticos, etc. Além disso, é necessário dispor de metodologias que permitam analisar de forma integrada e adequada as diferentes funcionalidades, por forma a obter uma perspectiva global do comportamento do sistema. Como exemplo ilustrativo, refira-se o processo de validação da arquitectura tolerante a falhas GUARDS (*Generic Upgradable Architecture for Real-Time Dependable Systems*) (Fig. 1.8) [BFL+01].

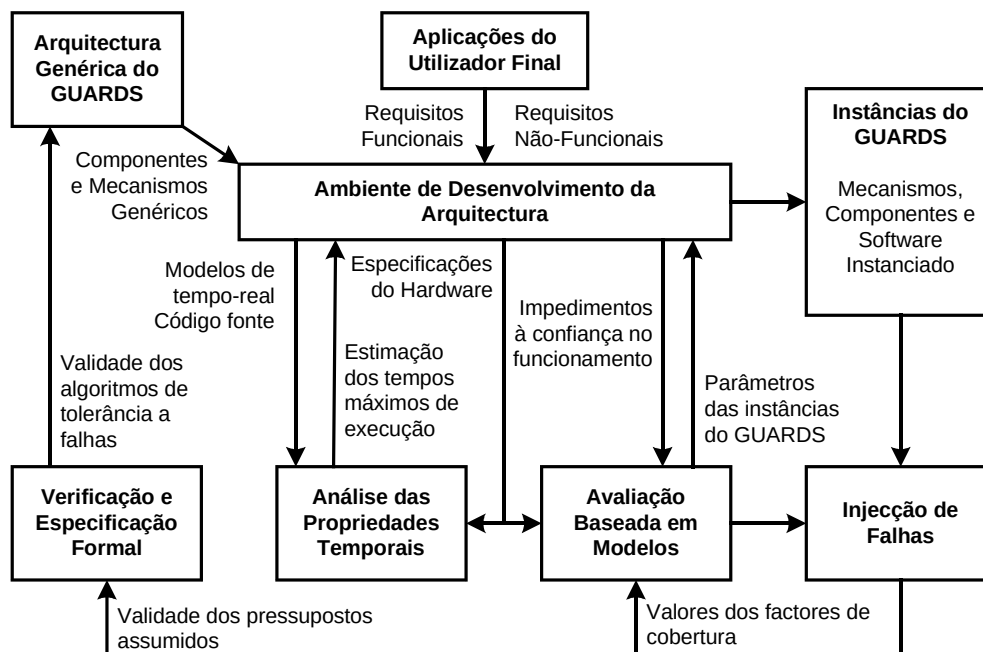


Figura 1.8- Ambiente de validação da arquitectura GUARDS.

Entre as várias metodologias utilizadas neste processo de validação, a utilização de modelos de avaliação estocásticos torna-se especialmente interessante devido à sua capacidade de analisar e prever antecipadamente o comportamento do sistema em cenários que envolvam falhas ao nível dos seus componentes ou de avarias do próprio sistema. Desta avaliação, resultam medidas que permitem caracterizar quantitativamente e qualitativamente este comportamento, e que podem ser utilizadas na tomada de decisões entre propostas de concepção alternativas, ou na definição das condições de operação do sistema. Além disso, a sua flexibilidade, aliada ao seu crescente poder de modelação, tem permitido ultimamente a sua utilização, com cada vez mais sucesso, na avaliação da confiança no funcionamento de sistemas complexos. Neste contexto, o processo de avaliação é geralmente decomposto em duas etapas, executadas sequencialmente [Lap95]:

1. Construção do modelo de avaliação do sistema a partir dos processos estocásticos que modelam o comportamento dos componentes do sistema e respectivas interacções;
2. Processamento do modelo por forma obter as expressões ou os valores das medidas da confiança no funcionamento do sistema.

Nas últimas décadas assistiu-se ao desenvolvimento de várias arquitecturas de controlo tolerantes a falhas (MARS [KDK+93], Delta-4 [Pow94], TTA [SHS+97], GUARDS [PAD+99]) que possibilitaram o desenvolvimento e a aplicação com sucesso de metodologias integradas ao nível da sua validação. Contudo, o sucesso destas abordagens está muito associado ao facto destas arquitecturas possuírem características bem definidas e cujo comportamento é

conhecido em pormenor. Isto por sua vez permitiu a utilização de técnicas e de metodologias de validação muito específicas e adaptadas às características destas arquitecturas.

A crescente importância da utilização das redes de campo em domínios onde é necessário garantir confiança no funcionamento suscita também a utilização de uma abordagem semelhante. Contudo, e ao contrário das arquitecturas anteriores, as redes de campo apresentam características muito diversas entre si, o que dificulta o emprego de uma metodologia integrada e única, capaz de ser utilizada genericamente em qualquer tipo de rede. Para atingir este objectivo é necessário utilizar uma abordagem que inclua diferentes perspectivas, partindo de uma identificação dos aspectos mais importantes ao nível do funcionamento de uma rede de campo em geral e traduzindo-os sob a forma de modelos de avaliação adequados, a partir dos quais é possível obter as medidas de interesse.

Dado o carácter recente da aplicação das redes de campo neste tipo de domínios, são relativamente poucos os trabalhos científicos realizados neste contexto. A sua quase maioria versa apenas sobre aspectos particulares da operação da rede (ex. falhas transitórias durante as comunicações), e são muito reduzidos os que procuram obter uma perspectiva integrada da avaliação da confiança no funcionamento. Mesmo estes, são geralmente baseados em pressupostos bastantes simples, e normalmente focados num tipo de rede em particular.

O objectivo desta dissertação centrou-se assim em preencher parte desta lacuna, através de uma abordagem focada no desenvolvimento de modelos estocásticos, que permitam avaliar a confiança no funcionamento das arquitecturas de controlo baseadas em redes de campo, contribuindo assim para o processo de validação das mesmas e também dos sistemas onde estas se integram.

A abordagem utilizada nesta dissertação para o desenvolvimento dos modelos foi realizada segundo três perspectivas:

1. Desenvolver modelos que permitam, durante as fases iniciais de especificação e desenvolvimento das arquitecturas de controlo, realizar uma avaliação comparativa entre diferentes soluções alternativas. Estes modelos preliminares devem ser simultaneamente genéricos para incorporar o funcionamento de diferentes tipos de redes, e adaptáveis para incluir aspectos particulares das mesmas. Devem também reflectir as limitações existentes nesta fase, nomeadamente ao nível dos dados disponíveis sobre a operação da rede;
2. Definir uma abordagem que permita incorporar de forma integrada diferentes perspectivas sobre o comportamento da rede, disponibilizando assim modelos de avaliação mais abrangentes e rigorosos. Esta abordagem deve realizar-se de acordo com a natureza multi-disciplinar dos vários aspectos envolvidos no

funcionamento da rede e reflectir de forma o mais fiel possível a respectiva operação num ambiente industrial real. Os modelos obtidos devem também ser genéricos e adaptáveis a diferentes tipos de cenários;

3. Desenvolver modelos que permitam avaliar o comportamento da rede em cenários de falhas transitórias que afectem as comunicações. Este tipo de falhas assume um carácter importante nos ambientes industriais, pelo que a sua análise é plenamente justificada. Os modelos devem resultar de uma abordagem rigorosa e pormenorizada, procurando reflectir da forma mais exacta possível o funcionamento da rede neste contexto. Para além deste objectivo principal, os modelos devem permitir incorporar os seus resultados no modelo de avaliação integrado descrito no ponto anterior. Dado que as arquitecturas de comunicações implementadas nas redes de campo possuem características distintas, o desenvolvimento dos modelos foi particularizado para duas redes: o Profibus-DP e o CAN. A opção pelo Profibus-DP baseou-se no facto desta ser presentemente a solução mais utilizada a nível mundial, e o CAN pela crescente importância que tem assumido ao nível dos sistemas de segurança crítica.

1.7 Contribuições da Dissertação

Esta dissertação apresenta as seguintes contribuições de relevo:

- ♦ Desenvolvimento de modelos analíticos baseados em cadeias de Markov Contínuas, que permitem realizar uma avaliação preliminar do comportamento de uma rede de campo em cenários de falhas permanentes. A abordagem centra-se nos efeitos das falhas sobre os componentes mais importantes da rede, incluindo também as avarias em modo-comum e os aspectos relacionados com a reparação dos componentes ou da própria rede. Os modelos propostos são genéricos e facilmente adaptáveis para a análise de diferentes arquitecturas. Para além de uma solução analítica/numérica, é possível a partir dos modelos extrair expressões das medidas de interesse e também identificar os parâmetros mais relevantes, facilitando assim o processo de avaliação;
- ♦ Desenvolvimento de modelos analíticos a partir da conjugação de uma análise probabilística com cadeias de Markov Discretas, que permitem realizar uma avaliação preliminar do comportamento de uma rede de campo em cenários de falhas transitórias que afectem as comunicações. A análise foca-se nos sistemas que possuem requisitos de tempo-real ao nível das comunicações. Os modelos incorporam o comportamento da *stack* de comunicações nas suas

várias camadas, incluindo também a possibilidade de existirem meios de transmissão redundantes e avarias em modo-comum dos mesmos. Os modelos incluem algumas aproximações, com diferentes níveis de rigor, resultantes da escassez de informações existentes durante estas fases. Para além de uma solução analítica/numérica, é possível, a partir de alguns dos modelos, obter expressões das medidas de interesse;

- ♦ Proposta de uma metodologia de avaliação integrada com o objectivo de desenvolver modelos que incorporem de forma adequada e rigorosa diferentes perspectivas sobre o comportamento da rede de campo na presença de falhas com características diversas. Os modelos são baseados em redes de Petri Estocásticas e permitem uma solução analítica. São incluídos os aspectos mais importantes associados ao funcionamento da rede em cenários de falhas permanentes e transitórias e de avarias em modo-comum, bem como as reparações dos componentes ou da própria rede. As falhas podem afectar diferentes tipos de componentes, situados em diferentes níveis funcionais e com com interacções variadas entre si. São considerados vários modos de avaria da rede, com diferentes níveis de gravidade;
- ♦ Desenvolvimento de um modelo de simulação baseado em redes de Petri Estocásticas para avaliação do comportamento de uma rede Profibus-DP, em cenários de falhas transitórias que afectem as comunicações. A análise foca-se nos sistemas que possuem requisitos de tempo-real ao nível das comunicações. O modelo é extremamente rigoroso e detalhado e inclui o funcionamento completo da *stack* de comunicações (física, ligação de dados e aplicação), nomeadamente os diferentes tipos de mensagens trocadas entre mestres e escravos;
- ♦ Desenvolvimento de um modelo de simulação baseado em redes de Petri Estocásticas para análise do comportamento de uma rede CAN, em cenários de falhas transitórias que afectem as comunicações. A análise foca-se nos sistemas que possuem requisitos de tempo-real ao nível das comunicações. O modelo considera cenários mais realistas ao nível da ocorrência das falhas, incluindo os efeitos de *bursts* de falhas, e também ao nível do funcionamento da rede, permitindo assim obter medidas de interesse menos pessimistas.

1.8 Estrutura da Dissertação

No capítulo 2, são apresentados os vários conceitos associados à confiança no funcionamento, bem como uma definição da terminologia associada em Português que será utilizada nos capítulos subsequentes. São também definidos e discutidos os critérios de escolha das técnicas de avaliação da confiança no funcionamento

utilizadas na dissertação, sendo dada particular atenção àquelas que vão ser utilizadas nos capítulos subsequentes, nomeadamente as cadeias de Markov, Discretas e Contínuas, as redes de Petri Estocásticas e a simulação.

No capítulo 3, é discutida uma avaliação preliminar do funcionamento de uma rede de campo em cenários de falhas permanentes e transitórias, incluindo avarias em modo-comum. Em relação às falhas permanentes, são propostos modelos baseados em cadeias de Markov Contínuas que representam duas arquitecturas genéricas. A partir da composição destes modelos é possível analisar um conjunto vasto de outras arquitecturas. É também apresentada uma técnica que a partir dos modelos obtidos permite obter expressões analíticas das medidas de interesse. Em relação às falhas transitórias, a análise foca-se nos seus efeitos durante as comunicações, nomeadamente ao nível da violação das *deadlines* associadas às mensagens de controlo transmitidas. É proposta uma abordagem baseada em argumentos probabilísticos e em cadeias de Markov Discretas para avaliar este comportamento. No final do capítulo é apresentado um caso de estudo.

No capítulo 4, são apresentados os motivos pelos quais é necessário utilizar uma metodologia integrada para avaliação da confiança no funcionamento das rede de campo. São discutidas as várias implicações destas abordagem, nomeadamente em relação à componente multi-disciplinar que lhe está subjacente. São analisados os diferentes trabalhos realizados nesta área, e é proposta uma abordagem analítica através do desenvolvimento de modelos baseados redes de Petri Estocásticas. São identificadas as limitações e os problemas que decorrem da utilização deste formalismo e são propostas técnicas de modelação adequadas para as resolver. Os modelos obtidos resultam de uma análise das características mais importantes que um modelo de avaliação integrado, genérico e rigoroso deve possuir. O capítulo termina com a apresentação de um caso de estudo.

No capítulo 5, são desenvolvidos modelos que permitem avaliar o comportamento de uma rede Profibus-DP em cenários de falhas transitórias que afectam as comunicações. São analisados os diferentes trabalhos realizados nesta área, e justificada a necessidade de uma nova abordagem. Os modelos propostos são baseados em redes de Petri Estocásticas, cuja solução é obtida através de simulação. No final do capítulo é apresentado um caso de estudo.

No capítulo 6, são desenvolvidos modelos que permitem avaliar o comportamento de uma rede CAN em cenários de falhas transitórias que afectam as comunicações. São analisados os diferentes trabalhos realizados nesta área, e justificada a necessidade de uma nova abordagem. Os modelos propostos são baseados em redes de Petri Estocásticas, cuja solução é obtida através de simulação. São também avaliadas as possibilidades de obter um modelo analítico. O capítulo termina com a apresentação de um caso de estudo.

No capítulo 7, são discutidas as conclusões e apresentadas propostas para trabalhos futuros.

1.9 Publicações Resultantes da Dissertação

- ♦ Dependability Evaluation of Profibus-DP Networks by Using a Stochastic Petri Net Framework, in Proceedings of the IEEE International Conference on Industrial Technology, ICIT'04 , 2004;
- ♦ A Framework for Dependability Evaluation of Fieldbus Networks, in Proceedings of the 5th IEEE International Workshop on Factory Communication Systems, WFCS'04, 2004;
- ♦ A Stochastic Petri Net Framework for Dependability Evaluation of Fieldbus Networks – A Controller Area Network (CAN) Example, in Proceedings of International Conference on Mechatronics and Robotics. MECROB'04, 2004;
- ♦ An Approach Based on Stochastic Petri Nets for Dependability Evaluation of Profibus-DP Networks, in Proceedings of 30th Annual Conference of the IEEE Industrial Electronics Society, IECON'04, 2004;
- ♦ A Framework for Dependability Evaluation of PROFIBUS Networks, in Proceedings of the International Symposium on Industrial Electronics, ISIE'03, 2003;
- ♦ Fieldbus Networks: a Dependability Evaluation, in Proceedings of ISA Instrumentation Society of America Industrial In-Plant Communications Conference, 2002;
- ♦ On Dependability Evaluation of Fieldbus Networks: a Permanent Fault Analysis, in Proceedings of 27th Annual Conference of the IEEE Industrial Electronics Society, IECON'01, 2001;
- ♦ On Dependability Evaluation of Fieldbus Networks: a Transient Fault Analysis, in IFAC International Conference on Fieldbus Systems and their Applications, FET'01, 2001.

Capítulo 2

Conceitos e Técnicas de Avaliação da Confiança no Funcionamento

2.1 Introdução

Neste capítulo pretende-se apresentar os vários conceitos associados à confiança no funcionamento, bem como as respectivas técnicas de avaliação.

Partindo do conceito unificador de confiança no funcionamento, são apresentados e discutidos um conjunto de outros conceitos associados. Pretende-se também sistematizar a respectiva terminologia em Português, por forma a dispor de uma linguagem consistente que sirva de suporte às discussões realizadas nos capítulos subsequentes.

A abordagem proposta nesta dissertação implica que avaliação do comportamento do sistema se realize através da utilização de modelos e de técnicas de avaliação estocásticas, com o objectivo de obter as medidas de interesse. Dado o grande número de técnicas existentes, é necessário definir critérios, com base nos

requisitos do sistema a avaliar, que permitam realizar uma escolha fundamentada sobre as mais adequadas a utilizar. Será dada particular atenção àquelas que vão ser empregues nos próximos capítulos, nomeadamente as cadeias de Markov, Discretas e Contínuas, as redes de Petri Estocásticas e a simulação.

A abordagem seguida neste capítulo não pretende ser nem exaustiva nem totalmente completa, mas apenas tornar esta dissertação o mais auto-contida possível.

2.2 Conceitos e Terminologia Associados à Confiança no Funcionamento

O processo de evolução de um qualquer domínio de conhecimento impõe à respectiva comunidade científica a tarefa de definir, preferencialmente de forma consensual e de acordo com critérios adequados, os conceitos e a terminologia necessários ao estabelecimento de uma semântica consistente e não-ambígua que permita a discussão e apresentação dos trabalhos de investigação realizados.

A área científica ligada à confiança no funcionamento não é excepção, tendo-se verificado ao longo dos últimos anos um esforço por parte de vários investigadores no sentido da definição, de forma clara e aceite pela comunidade, de uma proposta para alguns conceitos fundamentais. Destaca-se, nesta área, a formação de vários grupos de trabalho: *IEEE Computer Society Technical Committee on Fault Tolerant Computing* nos anos 70 e o *IFIP Working Group 10.4 Dependable Computing and Fault Tolerance* nos anos 80, que contribuíram fortemente para uma definição consistente de um conjunto de conceitos e da respectiva terminologia associada. Durante os anos 80 assistiu-se nesta área a uma progressiva sistematização de conceitos, nomeadamente através da publicação de vários artigos de posição em conferências de referência na área [FTCS82] [Lap85] e em trabalhos realizados no âmbito do IFIP WG 10.4. O culminar deste trabalho resultou na publicação em 1992 por J.-C. Laprie de um livro considerado como o documento de referência nesta área: *Dependability: Basic Concepts and Terminology* [Lap92]. Posteriormente a este, foram publicados outros artigos [Lap93] [Lap95] [ALR00] com o objectivo de refinar alguns dos conceitos apresentados no documento original.

Nas secções seguintes são apresentados um conjunto de termos e definições básicas relativas à área da confiança no funcionamento. Um dos problemas que se colocou nesta apresentação, foi o relacionado com a adopção de uma terminologia em Português adequada. Das várias propostas existentes [VL89] [Mag95] [AV98], considerou-se que a última representa a abordagem mais consistente e adequada da

tradução para Português da terminologia associada. Para além desse documento ser uma tradução devidamente comentada do texto original de J.-C Laprie ([Lap92]), incorpora também as mais recentes evoluções na área, quer ao nível científico, quer ao nível da terminologia.

2.2.1 Definição de Confiança no Funcionamento

A introdução do conceito de confiança no funcionamento tornou-se necessária devido à proliferação de várias perspectivas distintas sobre a análise do sistema (ex. a fiabilidade, a disponibilidade, etc.), que permitiam apenas obter uma visão parcial das questões. Assim, vários aspectos que anteriormente eram tratados separadamente, são agora tratados de forma unificada, permitindo obter uma perspectiva global.

Por definição, a **confiança no funcionamento** (*dependability*) de um sistema *é a propriedade desse sistema que permite que um seu utilizador possa depositar uma confiança justificada no serviço que este presta* [Lap92].

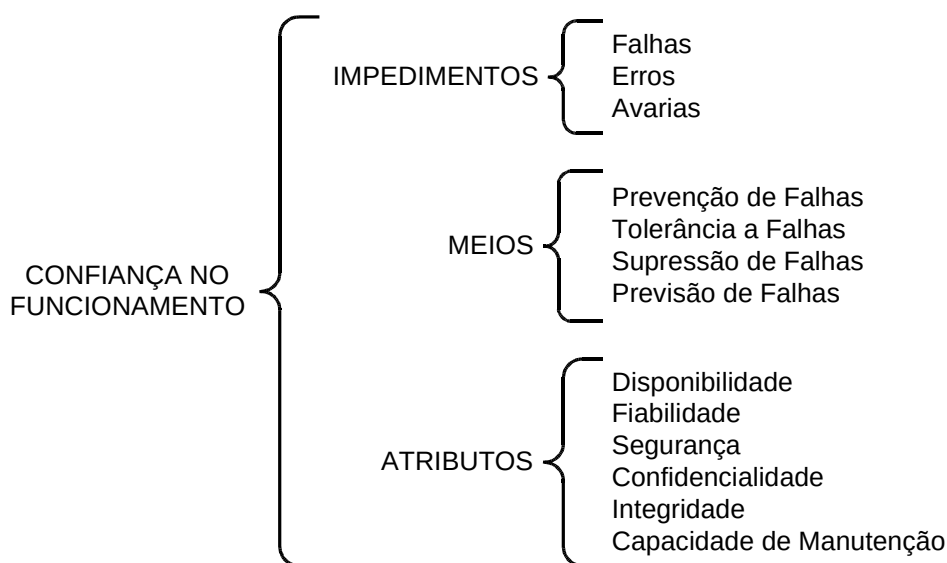


Figura 2.1 - Conceito de confiança no funcionamento.

Este conceito pode ser analisado a partir de três perspectivas (Fig. 2.1):

- ♦ Os **impedimentos** à obtenção da confiança no funcionamento: as falhas, os erros e as avarias. São indesejados, mas em princípio não inesperados, sendo resultantes da não-confiança no funcionamento do sistema;
- ♦ Os **meios** para garantir a confiança no funcionamento: prevenção de falhas, tolerância a falhas, supressão de falhas e previsão de falhas;

Estes são métodos e técnicas que permitem:

- Garantir a capacidade de fornecer um serviço em que possa ser justificada a confiança;
- Obter a confiança nesta capacidade.
- ♦ Os **atributos** da confiança no funcionamento: disponibilidade, fiabilidade, segurança, confidencialidade, integridade e a capacidade de manutenção. Estes permitem:
 - Expressar um conjunto de propriedades que se esperam do sistema;
 - Avaliar a qualidade do sistema que resulta dos impedimentos à confiança no funcionamento e dos meios para os ultrapassar.

A definição do conceito de **sistema** assume neste contexto um papel importante, dado que grande parte dos conceitos associadas à confiança no funcionamento recorrem a este. Os vários autores nesta área assumem posições muito próximas. Assim, [Lap92] define-o a partir de dois pontos de vista: um externo em que é considerado como *uma entidade que interage ou interfere com outras entidades*, isto é, com outros sistemas, e um outro interno, ou estrutural, como um *conjunto de componentes interligados de forma a interagirem*. [LA90] define-o como um *conjunto de componentes interligados juntamente com uma arquitectura que prescreve e controla um padrão de interacção*. De ambas as definições, resulta que os componentes de um sistema podem ser considerados também como sistemas de direito próprio (esta é uma definição recursiva que pode terminar com a consideração de componentes atómicos, ou seja, que não passíveis de mais decomposições) e portanto de serem analisados como tais.

O conjunto dos outros sistemas com os quais o sistema interage é considerado o seu **ambiente**. A interacção do sistema com este último é realizada através de uma **interface**. O **utilizador** do sistema é um outro sistema (físico ou humano, fazendo parte do ambiente) que utiliza o serviço disponibilizado pelo mesmo. O **serviço** prestado pelo sistema é definido como uma abstracção do seu comportamento externo, ou seja a forma como este comportamento é percebido pelos seus utilizadores.

A **função** de um sistema é aquilo a que se destina o sistema, cuja descrição é realizada através de uma **especificação**. Esta representa uma descrição acordada da função e/ou do serviço esperado.

2.2.2 Impedimentos à Confiança no Funcionamento

O serviço é considerado **correcto** quando implementa a função do sistema, ou seja se está conforme com a sua especificação. Caso contrário considera-se **incorrecto**.

Uma **avaria** (*failure*) é um evento que ocorre quando o serviço prestado pelo sistema se desvia do serviço correcto. Isto acontece quando o serviço prestado deixa de estar conforme a especificação, ou quando a especificação não descreve de forma adequada a função do sistema.

Uma forma simples de analisar o comportamento externo e interno do sistema, é através da caracterização da sua evolução a partir de um conjunto de estados [LA90], uns considerados correctos e outros incorrectos. Assim, uma avaria pode ser vista como a transição (evento) entre dois estados externos caracterizados respectivamente pela prestação de um serviço **correcto** (estado correcto) e **incorrecto** (estado incorrecto). A transição no sentido oposto é denominada **restauração** do serviço.

Um **erro** no sistema é definido como parte de um estado interno (incorrecto) capaz de levar a uma avaria subsequente. Isto é, a existência de um erro não é imediatamente sinónimo de avaria, mas evolução do sistema a partir deste estado interno pode levar à avaria do mesmo. A avaria ocorre quando o erro atinge a interface do serviço e altera o respectivo serviço.

Sendo o estado interno o conjunto dos estados externos dos componentes do sistema, então um erro no sistema implica que o estado externo de algum dos seus componentes esteja errado. Note-se, que embora um componente possa apresentar um estado externo errado isto não implica que este último esteja avariado. Pode acontecer que este estado não seja compatível com o estado dos outros componentes, devido por exemplo a uma avaria na arquitectura da sistema, ou à propagação de um erro a partir de um outro componente avariado.

Uma **falha** (*fault*) é a causa hipotética ou julgada de um erro. Assim, uma falha manifesta-se como erro num componente e/ou na arquitectura do sistema.

Concluindo, uma avaria num componente do sistema introduz uma falha no sistema. Esta manifesta-se através de um erro, que por sua vez pode levar à avaria do sistema, e portando do serviço que este fornece. Assim, uma avaria de um sistema manifesta-se como uma falha no sistema de nível hierárquico superior onde este se enquadra. Esta definição de falha é propositadamente arbitrária, por forma a parar a definição recursiva entre a relação causal de falhas, erros e avarias. De acordo com [Lap95], uma falha é nada mais do que a consequência de uma avaria num outro sistema que forneceu ou fornece um serviço ao sistema em causa. Um

erro é assim a manifestação de uma falha ao nível do sistema, enquanto que uma avaria é a manifestação de um erro no serviço.

2.2.2.1 Falhas

As falhas e as suas fontes são extremamente diversas, podendo ser classificadas de acordo com os seguintes critérios (Fig. 2.2) [ALR00]:

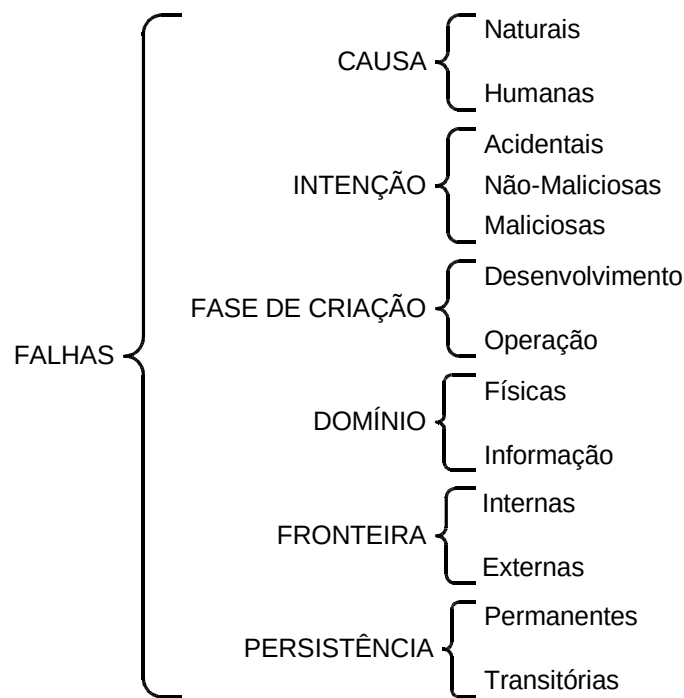


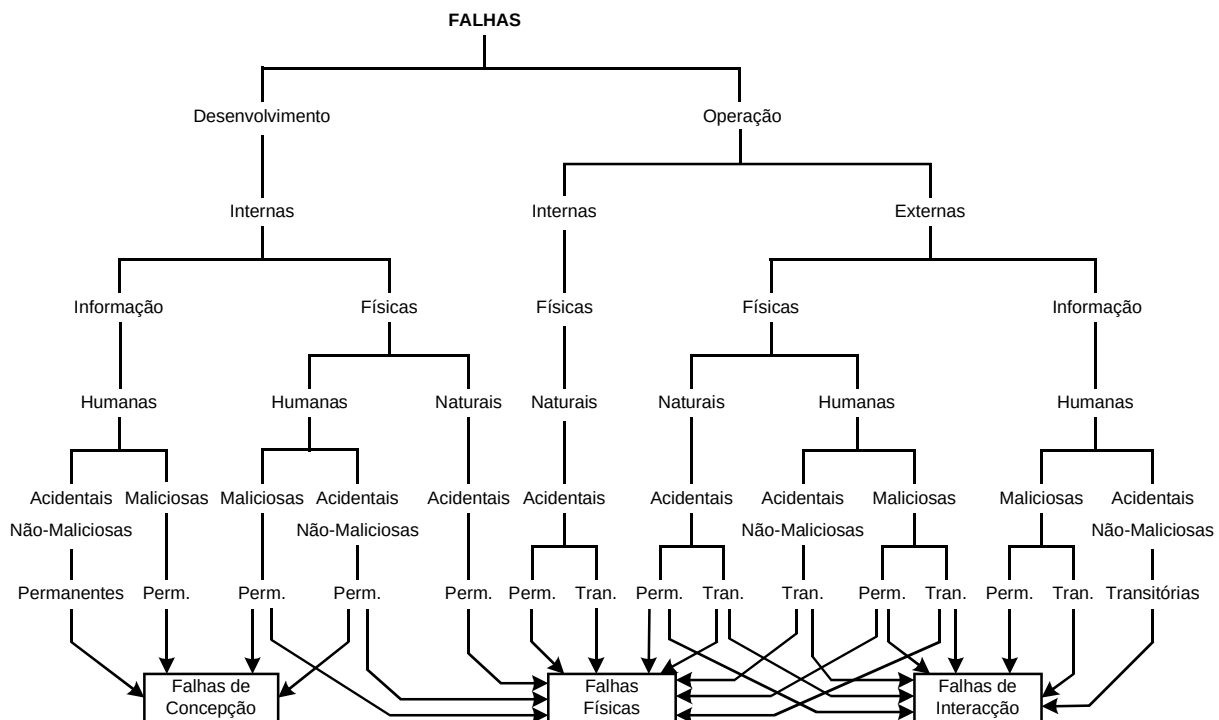
Figura 2.2 - Classes elementares de falhas.

- ♦ Causa fenomenológica
 - **Naturais**: quando resultam de fenómenos físicos;
 - **Humanas**: quando resultam de erros humanos.
- ♦ Intencionalidade
 - **Acidentais**: quando aparecem ou são criadas fortuitamente;
 - **Não-maliciosas**: quando são criadas deliberadamente, mas sem intenção de causar danos ao sistema;
 - **Maliciosas**: quando são criadas deliberadamente com intenção de causar danos ao sistema.

- ♦ Fase de criação ou ocorrência
 - **Desenvolvimento:** que ocorrem durante a fase de desenvolvimento do sistema;
 - **Operação:** que ocorrem durante a fase de operação do sistema.
- ♦ Domínio:
 - **Físicas:** quando têm origem em fenómenos físicos;
 - **Informação:** quando têm origem no processamento de informação.
- ♦ Fronteira do sistema
 - **Internas:** quando tem origem interna ao sistema;
 - **Externas:** quando tem origem externa ao sistema através da interacção ou interferência com o ambiente.
- ♦ Persistência temporal
 - **Permanentes:** quando não resultam de condições pontuais, externas ou internas;
 - **Transitórias:** quando resultam de situações pontuais, e portanto de duração limitada.

É importante ter em conta que esta classificação não significa que uma falha seja vista como pertencente a uma única classe. Pelo contrário, a maioria das falhas podem-se classificar-se em várias classes. Segundo [Lap95] não é necessário considerar a totalidade das combinações possíveis, sendo comum o agrupamento em classes combinadas (Fig. 2.3):

- ♦ **Físicas:** relacionam as que partilham as falhas físicas como falhas elementares;
- ♦ **Concepção:** resultantes de erros (humanos) realizados durante: (i) a concepção do sistema ou subsequentes modificações; (ii) a especificação de procedimentos para a operação e manutenção dos sistema;
- ♦ **Interacção:** resultantes da violação de procedimentos de operação ou manutenção.



2.2.2.2 Erros

Os erros podem ser classificados da seguinte forma:

- ◆ **Permanentes:** quando persistem permanentemente até uma acção explícita de reparação os remover;
- ◆ **Transitórios:** quanto existem por um curto intervalo de tempo (comparado com a referência do sistema) e desaparecem sem se ter efectuado qualquer acção de reparação.

Um erro pode estar num de dois estados: **latente** ou **detectado**. Um erro é dito latente quando ainda não foi reconhecido como tal. A sua detecção pode ser realizada por intermédio de mecanismos ou algoritmos específicos. Um erro pode desaparecer antes de ser detectado, podendo, além disso, propagar-se e originar outros erros.

2.2.2.3 Avarias

Na prática não existem sistemas que nunca avariaram, sendo portanto necessário caracterizar as maneiras pelas quais este pode avariar. Estas maneiras denominam-se **modos de avarias**, que podem ser classificados de acordo a sua **gravidade**. Os modos de avaria caracterizam o serviço incorrecto de acordo com os seguintes pontos de vista (Fig. 2.4):

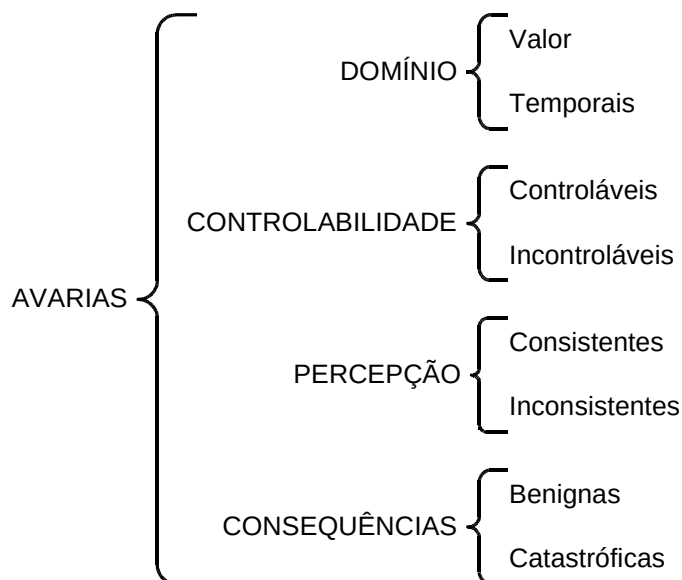


Figura 2.4 - Modos de avaria.

- ♦ Domínio
 - **Valor:** quando o valor lógico do serviço é incorrecto;
 - **Temporais:** quando a prestação do serviço é efectuada fora de um intervalo temporal especificado.
- ♦ Controlabilidade
 - **Controláveis:** quando o modo de avaria foi especificado e o sistema comporta-se de acordo com a especificação;
 - **Incontroláveis:** quando o modo de avaria não está de acordo com a especificação ou não foi especificado.
- ♦ Percepção pelo utilizador
 - **Consistentes:** quando todos os utilizadores do sistema têm a mesma percepção da avaria;
 - **Inconsistentes:** quando utilizadores diferentes, têm percepções diferentes da mesma avaria.
- ♦ Consequências sobre o ambiente
 - **Benignas:** quando as consequências da avaria são da mesma ordem de grandeza dos benefícios proporcionados pelo funcionamento correcto do sistema;

- **Catastróficas:** quando as consequências da avaria são várias ordens de grandeza superiores aos benefícios proporcionados pelo funcionamento correcto do sistema.

2.2.2.4 Patologia das Avarias

Os mecanismos de criação e manifestação de falhas, erros e avarias podem ser descritos pela seguinte sequência (Fig. 2.5) [ALR00]:

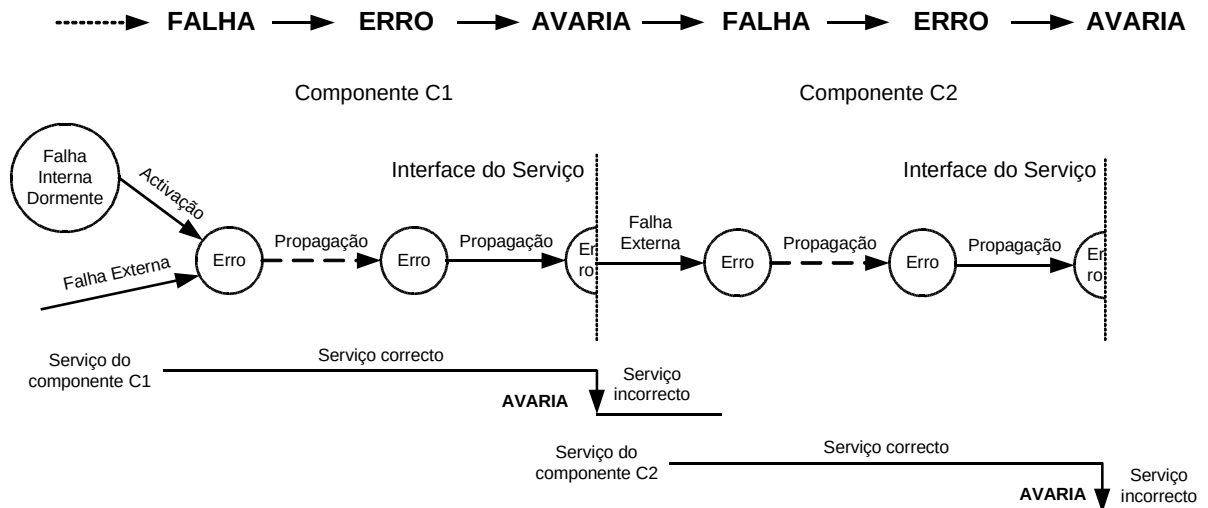


Figura 2.5 - Relação causal entre falhas, erros e avarias.

1. A falha torna-se **activa** quando produz um erro, caso contrário é dita **dormente**. Uma falha activa poderá ser: (i) uma falha interna que estava previamente dormente e que foi posteriormente activada pelo processo computacional ou por condições ambientais; (ii) uma falha externa. A **activação** de uma falha resulta da aplicação de uma entrada a um componente que obriga a falha dormente a tornar-se activa;
2. A propagação do erro dentro de um componente é causada pelo processo computacional. O erro é sucessivamente transformado em outros erros.
3. A avaria ocorre quando o erro se propaga à interface do serviço e altera de forma inaceitável o serviço prestado pelo sistema. A avaria de um componente causa uma falha no sistema que contém o componente. Os modos de avaria de um componente tornam-se assim os tipos de falhas para os componentes que com ele interagem.

2.2.3 Meios Para Obter Confiança no Funcionamento

A concepção de um sistema em cujo funcionamento se pode confiar, implica necessariamente a utilização combinada de um conjunto de meios, que podem ser classificados da seguinte forma:

- ♦ **Obtenção**
 - **Prevenção de falhas:** como prevenir durante a fase de desenvolvimento a ocorrência ou a introdução de falhas;
 - **Tolerância a falhas:** como disponibilizar um serviço capaz de garantir a função do sistema mesmo na presença de falhas.
- ♦ **Validação**
 - **Supressão de falhas:** como reduzir a presença, através de uma verificação, quer em número quer em gravidade, das falhas;
 - **Previsão de falhas:** como estimar, através de uma avaliação, o número presente de falhas, a futura incidência e a as suas consequências.

2.2.3.1 Prevenção de Falhas

O objectivo da prevenção de falhas é impedir uma avaria do sistema, assegurando que todas as causas possíveis das mesmas são removidas durante a fase de desenvolvimento, isto é, antes que o sistema seja colocado em funcionamento.

Para atingir este objectivo é necessário utilizar um conjunto de técnicas e tecnologias, ao nível do *hardware* e do *software*, que impeçam/evitem a introdução de falhas no sistema. A utilização de componentes com elevada fiabilidade e o uso de metodologias estruturadas para o desenvolvimento de *software*, são alguns exemplos destas técnicas. Embora se tenha verificado ao longo destes anos uma grande evolução nesta área, especialmente nas metodologias de desenvolvimento de *software*, a complexidade dos sistemas também tem aumentado, o que torna praticamente impossível garantir que o sistema desenvolvido esteja ausente de falhas.

2.2.3.2 Tolerância a Falhas

O objectivo da tolerância a falhas é o de melhorar a confiança no funcionamento no sistema, permitindo que este último forneça o seu serviço na presença de falhas activas. Esta pode ser decomposta nas seguintes fases, executadas sequencialmente:

1. **Processamento de erros:** cujo objectivo é a detecção e a remoção dos erros, que possam subsequentemente dar origem a falhas. É importante salientar que a identificação de uma falha activa só pode ser determinada a partir da detecção de um erro;
2. **Tratamento de falhas:** cujo objectivo é impedir que as falhas sejam reactivadas.

Quando se pretende que um sistema tolere um conjunto de falhas é necessário que durante a fase de especificação do mesmo se indiquem quais os tipos e qual a frequência das falhas que se pretendem tolerar, isto é as **hipóteses de falhas** (*fault hypotheses*).

Directamente ligadas a estas hipóteses estão as escolhas das várias técnicas associadas à implementação da tolerância a falhas. A situação mais comum é recorrer à utilização de técnicas **redundância**, que podem ser classificadas como:

- ♦ **Estrutural**: relacionada com a replicação da estrutura física ou lógica do sistema;
- ♦ **Temporal**: com o repetido uso de um componente ou de uma sequência de acções;
- ♦ **Informação**: relacionada com replicação de estruturas de informação (dados).

2.2.3.2.1 Processamento de Erros

O processamento de erros pode ser decomposto nas seguintes fases, executadas sequencialmente:

1. **Deteção de erros**: utilização de mecanismos/técnicas que permitem a identificação de um estado errado no sistema;
2. **Diagnóstico de erros**: avaliação dos danos causados pelos erros, ou pela propagação de um erro antes da sua detecção;
3. **Recuperação de erros**: substituição de um estado errado do sistema por um correcto. Esta actividade pode ser realizada de três formas:
 - **Recuperação para trás** (*backward recovery*): consiste em levar o sistema para um estado que anteriormente já tinha ocupado, antes da ocorrência do erro. Isto envolve o estabelecimento de pontos de recuperação (*recovery points*), que são pontos, no tempo, durante a execução de um processo a partir dos quais um estado, livre de erros, pode ser subsequentemente restaurado;
 - **Recuperação para a frente** (*forward recovery*): consiste em encontrar um novo estado, livre de erros, a partir do qual o sistema possa continuar a operar;
 - **Compensação de erros**: o estado errado do sistema possui redundância suficiente para permitir a sua transformação num estado sem erros.

2.2.3.2 Tratamento de Falhas

O tratamento das falhas pode ser decomposto nas seguintes fases, executadas sequencialmente:

1. **Diagnóstico das falhas:** identificação e registo das causas dos erros em termos da sua localização e tipo;
2. **Isolamento das falhas:** exclusão física ou lógica dos componentes avariados, impedindo-os de participarem em futuras prestações do serviço;
3. **Reconfiguração do sistema:** modificação da estrutura do sistema de forma a que os componentes não avariados permitam prestar um serviço, com eventuais diferentes níveis de prestação.

2.2.3.3 Supressão de Falhas

A supressão de falhas é realizada durante as fases de concepção e de operação do sistema. Durante a fase de concepção, as actividades realizadas podem ser decompostas nas seguintes fases, executadas sequencialmente:

1. **Verificação:** determinar se o sistema está de acordo, ou não, com um conjunto de propriedades denominadas *condições de verificação*. Esta análise pode ser realizada recorrendo a um conjunto variado de técnicas, das quais se salientam os testes de conformidade, funcionais, estruturais, etc.;
2. **Diagnóstico de falhas:** diagnosticar os motivos pelos quais as condições de verificação não foram respeitadas;
3. **Correcção de falhas:** através da correcção das falhas encontradas. Depois desta fase o sistema deve ser reinicializado para se verificar que o processo de supressão de falhas não introduziu consequências indesejáveis.

As técnicas de verificação podem ser classificadas dependendo de envolverem ou não a activação do sistema. A verificação de um sistema sem o activar é denominada **estática**, caso contrário é **dinâmica**. Neste último caso, as entradas fornecidas ao sistema podem ser simbólicas, no caso da **execução simbólica**, ou estimadas no caso de **testes de verificação**.

A supressão de falhas durante a fase operacional do sistema é realizada por intermédio de manutenção correctiva, que por sua vez pode ser decomposta em manutenção curativa e preventiva.

2.2.3.4 Previsão de Falhas

A previsão das falhas que podem vir a afectar o sistema é baseada na avaliação do comportamento do sistema em relação à ocorrência e à activação das falhas. Esta avaliação pode ser realizada de duas formas:

- ♦ **Qualitativamente:** (i) identificar, classificar e ordenar os modos de avarias do sistema; (ii) identificar as combinações de eventos (avarias de componentes ou condições ambientais) que podem levar a eventos indesejados;
- ♦ **Quantitativamente:** avaliar, em termos probabilísticos, alguns dos atributos da confiança no funcionamento, que por sua vez podem ser vistos como **medidas** da confiança no funcionamento.

As duas principais abordagens à previsão de falhas probabilística, tendo como objectivo a obtenção de estimativas (probabilísticas) da confiança no funcionamento, são a **modelação** e os **testes de avaliação** (ex. injeção de falhas). Estas abordagens são complementares dado que o processo de modelação necessita de dados sobre o sistema modelado, que por sua vez podem ser obtidos através de testes, ou através do processamento de dados sobre as avarias.

A existência de uma forte interacção entre a supressão e a previsão de falhas, permite que ambos os meios possam ser vistos de uma única perspectiva, denominada **validação** do sistema.

2.2.4 Atributos da Confiança no Funcionamento

A confiança no funcionamento pode ser analisada a partir de diferentes, mas complementares, propriedades, que por sua vez permitem a definição dos respectivos atributos:

- ♦ **Disponibilidade:** prontidão para prestar um serviço correcto;
- ♦ **Fiabilidade:** continuidade da prestação de um serviço correcto;
- ♦ **Segurança** (*safety*): ausência de consequências catastróficas para os utilizadores e o ambiente;
- ♦ **Confidencialidade:** ausência de divulgação não autorizada de informação;
- ♦ **Integridade:** ausência de alterações impróprias do estado do sistema;
- ♦ **Capacidade de manutenção** (*maintainability*): aptidão para suportar reparações e modificações.

Alguns destes atributos podem quantificados e vistos como **medidas** da confiança no funcionamento. Na secção §2.4 realizar-se-à uma descrição formal das mesmas.

A importância destes atributos é muito dependente das aplicações respectivas. Assim, a fiabilidade e a disponibilidade são sempre requeridas (o sistema tem que funcionar), embora num grau variável entre aplicações. A segurança e a confidencialidade podem ou não ser requeridas em função da aplicação. A integridade é um pré-requisito para a disponibilidade, a fiabilidade e a segurança, mas pode não o ser para a confidencialidade. A **inviolabilidade** (*security*), no sentido de falhas intencionais, pode ser vista como uma combinação da confidencialidade, da integridade e da disponibilidade.

A definição de capacidade de manutenção pode ser vista de várias formas:

- ♦ **Manutenção curativa:** remoção de falhas que produziram erros ou avarias;
- ♦ **Manutenção preventiva:** remoção de falhas antes que estas produzam erros ou avarias;
- ♦ **Manutenção correctiva:** preservar ou melhorar a capacidade do sistema para fornecer o respectivo serviço. Esta pode ser vista como a conjugação da manutenção curativa e preventiva;
- ♦ **Manutenção adaptativa:** ajustar a sistema em função das mudanças ambientais;
- ♦ **Manutenção perfectiva:** melhorar a função do sistema em função de mudanças registadas ao nível quer do cliente, quer de quem concebe o sistema.

2.3 Técnicas de Avaliação da Confiança no Funcionamento

A avaliação do comportamento de um sistema, com o objectivo de obter de forma probabilística estimativas quantitativas das medidas da confiança no funcionamento, pode ser realizada através de duas abordagens distintas, embora complementares:

- ♦ Utilizando o sistema real, através da observação e da caracterização do comportamento das suas saídas em função de valores que são aplicados às respectivas entradas;

- ♦ Utilizando um modelo cujo comportamento representa uma abstracção do funcionamento do sistema real.

Embora fosse possível e interessante discutir as vantagens e as desvantagens inerentes a cada uma das abordagens no sentido de permitir uma escolha entre ambas, foi definido à partida como um requisito prévio a esta dissertação a utilização de modelos de estocásticos para realizar a avaliação da confiança no funcionamento dos sistemas em causa. As razões que levaram à tomada desta opção foram de três tipos. A primeira e a mais importante, foi a possibilidade de prever antecipadamente qual o comportamento do sistema em cenários de falhas, e quais as consequências das respectivas avarias. Esta informação pode ser utilizada a vários níveis, nomeadamente na tomada de decisões entre propostas de concepção alternativas, na identificação dos componentes mais críticos ao nível do funcionamento do sistema e na definição das condições de operação do mesmo. A segunda razão, está relacionada com a rapidez e a flexibilidade com que o processo de avaliação pode ser realizado. A utilização de formalismos de modelação suportados por ferramentas adequadas permite desenvolver modelos de avaliação forma muito flexível, com a possibilidade de analisar de forma rápida e eficaz diferentes configurações para o funcionamento e a operação do sistema, ou avaliar qual o grau de influência nas medidas de interesse dos vários pressupostos assumidos durante o desenvolvimento do modelo. Por fim, todo o processo de avaliação pode ser realizado sem a necessidade de ter que se dispor fisicamente do sistema em causa. Este aspecto é particularmente importante quando o sistema é complexo de avaliar fisicamente por motivos que vão desde a sua dimensão geográfica, até às dificuldades intrínsecas de conceber e de implementar os testes de avaliação (ex. avaliação de sistemas ultra-fiáveis).

Contudo, o emprego desta abordagem não comporta apenas vantagens, existindo também várias limitações [BFL+01]. Entre estas destacam-se:

- ♦ Necessidade de dispor de dados fiáveis relacionados com a ocorrência das falhas nos componentes do sistema. Como as medidas de interesse obtidas a partir do modelo resultam directamente dos dados utilizados no mesmo, a credibilidade das primeiras depende fortemente destes últimos;
- ♦ Os modelos muitas das vezes dependem de parâmetros cujas variações são extremamente críticas ao nível do seu impacto nos resultados obtidos (ex. factores de cobertura). Neste caso, a credibilidade dos resultados implica que estes parâmetros necessitem de ser determinados com o maior rigor possível;
- ♦ O modelo, consequência do formalismo utilizado, pode não ter a capacidade de representar de forma correcta as interacções e as dependências funcionais e estruturais que existem na prática entre os diferentes componentes do sistema;

- ♦ O tipo de solução pretendida para o modelo (ex. analítica) obriga à utilização de várias simplificações ao nível da modelação, com o objectivo de garantir a exequibilidade da respectiva solução.

Embora as limitações da abordagem proposta possam parecer de certa forma problemáticas, é importante notar, tal como discutido no capítulo 1, que a realização com sucesso do processo de validação de um sistema implica a utilização combinada e integrada de um conjunto de metodologias de natureza distintas. Nesta perspectiva, a avaliação utilizando modelos estocásticos deve ser encarada um dos meios necessários (mas não o único) para realizar esta avaliação. O conjunto de vantagens disponibilizadas por uma abordagem com estas características, permite mesmo concluir que a sua utilização se traduz por um importante valor acrescentado ao nível do processo de validação de um sistema.

2.3.1 Formalismos de Modelação Estocásticos

O modelo de avaliação deve garantir dois requisitos essenciais: representar com o maior rigor possível o comportamento do sistema, e permitir a avaliação da respectiva confiança no funcionamento. O seu desenvolvimento deve ser realizado a partir das características do funcionamento do sistema, mas numa perspectiva relacionada com a ocorrência das falhas e respectivas consequências. Ou seja, o modelo representa o comportamento do sistema durante a ocorrência de um conjunto de falhas que afectam os seus componentes e cujas as consequências se repercutem no funcionamento global do sistema. A partir do modelo desenvolvido são aplicadas técnicas (de avaliação) apropriadas que permitem obter as medidas (de confiança no funcionamento) de interesse.

As características do modelo obtido resultam naturalmente do formalismo de modelação utilizado durante o seu desenvolvimento. Cada formalismo possui uma determinada capacidade de modelação, que por sua vez restringe o tipo de comportamentos que podem ser modelados, bem como o tipo de medidas que podem ser obtidas. No entanto, comum a todas eles está o facto de assumirem, de forma directa ou indirecta, a existência de um comportamento estocástico por parte do sistema. Isto resulta directamente dos vários pressupostos assumidos neste tipo de avaliação, nomeadamente os associados aos processos de ocorrência das falhas. Tendo em conta que estes processos têm tipicamente um comportamento estocástico, é lógico assumir que este se estende ao funcionamento do próprio sistema, induzindo desta forma um comportamento análogo. Além disso, o próprio sistema pode depender de outros processos cuja natureza seja também estocástica. Estes podem ser externos ao sistema, como é o caso das reparações, ou internos, relacionados com aspectos específicos do funcionamento do mesmo.

São muitas as técnicas³ que permitem avaliar a confiança no funcionamento de sistemas, das quais se destacam as seguintes:

- ♦ Análise de Modos e de Efeitos de Avarias (FMEA - *Failure Modes and Effects Analysis*) [MIL80] [BAD+04];
- ♦ Análise de Modos e de Efeitos de Avarias e da sua Criticalidade (FMECA - *Failure Modes, Effects and Criticality Analysis*) [Wei91];
- ♦ Blocos de Fiabilidade (RB - *Reliability Blocks*) [Sho70] [BA91];
- ♦ Gráficos de Fiabilidade (RG - *Reliability Graphs*) [Col87];
- ♦ Árvore de Falhas (FT - *Fault Trees*) [LGT+85] [PD96];
- ♦ Árvore de Falhas com Eventos Repetidos (FTR - *Fault Trees with Repeated Events*) [DBB92];
- ♦ Árvore de Eventos (ET - *Event Trees*) [Pap98] [AD00];
- ♦ Cadeias de Markov [MMT96];
- ♦ Redes de Petri Estocásticas [MT95];
- ♦ Simulação [LK00].

Para além destas técnicas existem também outras, mas cuja gama de aplicação é mais reduzida, destacando-se as seguintes:

- ♦ Estudo de Perigos e Operacionalidade (HAZOP - *HAZard and OPerability study*) [TK99];
- ♦ Álgebras de Processos Estocásticos (SPA - *Stochastic Process Algebras*) [Lin98];
- ♦ Fiabilidade Difusa (*Fuzzy Reliability*) [BP95].

Não é objectivo desta dissertação realizar uma apresentação detalhada das características de cada uma das técnicas indicadas, existindo para isso uma vasta literatura especializada [DS81] [BA91] [Cass93] [TFM+93] [MT94] [MBC+95] [MMT96] [Lin98] [Gob98] [Tys99] [PT00] [MFT00] [HMR+01] [Triv02] [Her02] [VV04]. No entanto e face ao elevado número de técnicas disponíveis, coloca-se a questão de escolher as mais adequadas para a avaliação dos sistemas discutidos nesta dissertação. Tendo em conta a complexidade desta questão, optou-se por definir um conjunto de critérios a partir dos quais, e após uma análise e ponderação cuidadas, se possa realizar uma escolha fundamentada das técnicas mais adequadas.

³ Na verdade a denominação correcta seria formalismo de modelação, contudo na literatura especializada, estes são usualmente denominados como técnicas de avaliação, pois está subjacente o seu emprego na avaliação da confiança no funcionamento. Refira-se contudo, que alguns dos formalismos apresentados podem ser utilizados também em outros contextos (ex. medidas de desempenho).

A definição desses critérios deve ser realizada tendo em conta quer as características do sistema que se pretende avaliar, quer as capacidades disponibilizadas por cada técnica. Assim, e de acordo com esta abordagem, foram definidos os seguintes critérios:

- ◆ **Natureza da avaliação**

Os resultados da avaliação podem ser expressos de forma qualitativa ou quantitativa. Neste último caso é também necessário definir quais os tipos de medidas que se pretendem obter;

- ◆ **Estrutura dos modelos**

Os modelos obtidos podem ser classificados em função das propriedades que decorrem da sua estrutura. Nesta perspectiva, é comum utilizar as seguintes denominações [PT00] [MFT00]: (i) **textuais**, quando o modelo é descrito através de uma estrutura textual (ex. tabelas); (ii) **combinacionais**, quando o modelo não resulta da enumeração dos estados do sistema; (iii) **espaço de estados**, quando o modelo implica e enumeração dos vários estados associados à evolução do sistema. Fortemente ligado a este aspecto, estão as formas de obter a solução do modelo;

- ◆ **Dependências funcionais e/ou estruturais**

Num sistema de controlo é natural existirem dependências funcionais ou estruturais entre os vários componentes que compõem o sistema (ex. a informação produzida por um componente é utilizada por outro; factores de cobertura das falhas; forma e tipo das reparações; redundância entre componentes, etc.). Estas dependências podem ser de vários tipos, e têm necessariamente de ser formalizadas de forma adequada. Assim, é necessário que a técnica utilizada tenha capacidade de representar de forma adequada o tipo de relações funcionais existentes entre componentes;

- ◆ **Dependências estocásticas**

Este tipo de dependência define uma relação funcional ao nível dos vários processos estocásticos subjacentes. Esta dependência pode ser complexa, e dar origem a interacções entre processos de diferentes tipos. Exemplos destes tipos de relações, ocorrem quando se consideram os processos de reparação de componentes, quando as taxas de avaria são função do estado actual do sistema, ou mais simplesmente quando o comportamento dos componentes (na perspectiva da ocorrências das falhas) não podem ser considerados independentes entre si;

- ◆ **Processos estocásticos**

A existência de um comportamento estocástico ao nível do sistema, implica que o tipo e as características do processo estocástico associado tenham que

ser convenientemente modelados. É assim necessário que a técnica utilizada seja capaz de os suportar de forma adequada. Exemplos deste tipo de processos, são os que resultam dos pressupostos assumidos ao nível da ocorrência de falhas e dos processos de reparação dos componentes;

◆ **Custo da solução**

No caso de uma avaliação quantitativa é necessário ter em conta quais os recursos (computacionais, tempo, etc.) necessários para obter, a partir do modelo, uma solução para as medidas de interesse. Relacionado com este aspecto está também qual o tipo solução pretendida (ex. numérica, analítica, simulação);

◆ **Ferramentas de avaliação**

Tendo em conta a complexidade dos sistemas actuais torna-se praticamente inviável realizar a respectiva avaliação através de técnicas manuais. Desta forma, a existência de ferramentas que suportem a técnica pretendida permite que a avaliação do sistema se faça de forma muito mais rápida e menos sujeita a erros de análise [JM88] [TM93] [HN96].

O mapeamento entre critérios definidos e as técnicas de avaliação da confiança no funcionamento previamente indicadas é apresentado na Tabela 2.1.

Os sistemas que se pretendem avaliar nesta dissertação possuem ao nível da sua estrutura as seguintes características⁴:

- ◆ Relações de dependência funcional e estrutural entre componentes;
- ◆ Relações de dependência estocástica entre processos;
- ◆ Processos estocásticos complexos (ex. Não-Markovianos);

Ao nível dos modelos que se pretendem obter, é possível estabelecer os seguintes requisitos:

- ◆ Avaliação quantitativa com uma solução preferencialmente analítica;
- ◆ Soluções de custo baixo/médio;
- ◆ Necessidade de ferramentas para proceder à avaliação.

A partir da informação apresentada (Tab. 2.1) torna-se óbvio que não existe uma técnica ideal (ou óptima) para realizar a avaliação pretendida, e que a sua escolha resulta necessariamente de um compromisso entre os vários aspectos envolvidos.

⁴ A antecipação das características dos sistemas a avaliar justifica-se pela necessidade de realizar uma escolha das técnicas de avaliação. Pretende-se assim previamente à apresentação de qualquer modelo, disponibilizar uma descrição detalhada das características das técnicas utilizadas no desenvolvimento dos mesmos.

Conceitos e Técnicas de Avaliação da Confiança no Funcionamento

Assim e de acordo com características e com os requisitos acima descritos, e em função das propriedades das técnicas apresentadas, conclui-se que as técnicas mais adequadas para realizar a avaliação pretendida dos sistemas em causa são as Cadeias de Markov e as Redes de Petri Estocásticas.

Técnica de Avaliação	Natureza da Avaliação	Estrutura dos Modelos	Dependências Funcionais e/ou Estruturais	Dependências Estocásticas	Processos Estocásticos	Custo da Solução	Ferramentas de Avaliação
FMEA	Qualitativa	Textuais	Relações básicas	Não	Não	Baixa	Várias
FMECA	Qualitativa e quantitativa	Textuais	Relações básicas	Não	Muito básicos	Baixa (Numérica)	Várias
Blocos de Fiabilidade	Quantitativa (limitações no tipo de medidas)	Combinacionais	Relações do tipo série, paralelo ou combinação de ambas	Não	Abrangente	Média (Analítica/Numérica)	Várias
Gráficos de Fiabilidade	Quantitativa (limitações no tipo de medidas)	Combinacionais	Relações do tipo série, paralelo ou combinação de ambas	Não	Abrangente	Média (Analítica/Numérica)	Várias
Árvore de Falhas	Quantitativa (limitações no tipo de medidas)	Combinacionais	Relações definidas a partir de condições lógicas	Não	Abrangente	Média (Analítica/Numérica)	Várias
Árvore de Falhas com Eventos Repetidos	Quantitativa (limitações no tipo de medidas)	Combinacionais	Relações definidas a partir de condições lógicas	Muito limitadas	Abrangente	Média (Analítica/Numérica)	Várias
Árvore de Eventos	Quantitativa (limitações no tipo de medidas)	Combinacionais	Relações definidas a partir de condições lógicas	Não	Abrangente	Média (Analítica/Numérica)	Várias
Cadeias de Markov	Quantitativa	Espaço de estados	Relações definidas a partir do estado do sistema	Sim (com algumas limitações)	Limitado (apenas Markovianos)	Média/Alta (Analítica/Numérica)	Muitas
Redes de Petri Estocásticas	Quantitativa	Espaço de estados	Relações definidas a partir do estado do sistema	Sim (com algumas limitações)	Ligeiramente abrangente (Markovianos e alguns Não-Markovianos)	Média/Alta (Analítica/Numérica)	Várias
Simulação	Quantitativa	Eventos discretos	Qualquer tipo de relação	Sem limitações	Sem limitações	Alta/Muito Alta (Simulação)	Muitas

Tabela 2.1 - Comparação entre técnicas de avaliação da confiança no funcionamento.

As técnicas escolhidas para o desenvolvimento dos modelos, correspondem àquelas que possuem maior poder de modelação (Fig. 2.6) [MFT00] [MT94]. Esta métrica é definida pelo tipo de dependências (funcionais e estocásticas) existentes no sistema que podem ser modeladas, bem como pelo tipo de medidas que podem ser obtidas.

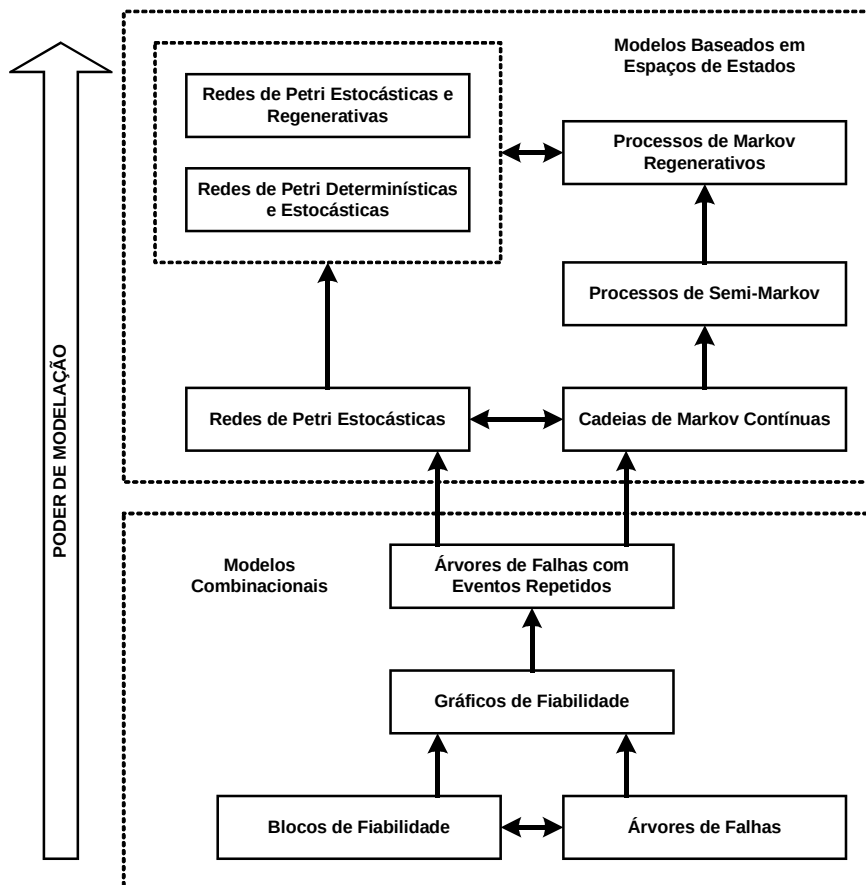


Figura 2.6 - Poder de modelação das várias técnicas de avaliação da confiança no funcionamento.

As técnicas escolhidas são baseadas em modelos denominados de **espaço de estados** (*state space models*) [MFT00], porque inerente à sua construção existe a necessidade de enumerar o espaço de estados do sistema modelado. Estes modelos sofrem do problema intrínseco associado à dimensão do respectivo espaço de estados. Quanto mais complexo for o sistema, ou as relações a modelar, maior será a dimensão do espaço de estados. Isto implica que o modelo seja capaz de armazenar toda esta informação, surgindo tipicamente dificuldades computacionais ao nível da obtenção da sua solução. É assim necessário utilizar abordagens de modelação adequadas para evitar estes problemas. No capítulo 5 este assunto será tratado com detalhe e serão apresentadas várias soluções para o mesmo. Embora não decorra directamente das características dos sistemas a avaliar, a simulação foi também empregue como técnica para a obtenção das soluções de alguns modelos. A necessidade de se recorrer a soluções de simulação foi devida às restrições impostas pela capacidade de modelação das técnicas analíticas.

Nas secções seguintes pretende-se apresentar dois aspectos centrais à avaliação da confiança no funcionamento. Em primeiro lugar, re-apresentar as medidas da confiança no funcionamento mas utilizando uma perspectiva mais formal e centrada na forma como vão ser obtidas. Em segundo lugar, discutir de forma concisa as características mais importantes de alguns processos estocásticos, nomeadamente aqueles que constituem a base dos vários modelos de confiança no funcionamento propostos nesta dissertação. Em secções distintas são apresentadas e discutidas as rede de Petri Estocásticas e as técnicas de simulação. No Anexo 1 são apresentadas sucintamente as propriedades de algumas funções distribuição mencionadas durante esta dissertação.

2.4 Medidas da Confiança no Funcionamento

As medidas da confiança no funcionamento podem ser de vários tipos, podendo contudo ser classificadas com pertencendo a duas grandes categorias: fiabilidade e disponibilidade do sistema⁵ [MFT00], a partir dais quais as restantes podem ser derivadas (Tab. 2.2). Podem ser avaliadas pontualmente, num instante de particular do tempo, ou assintoticamente quando $t \rightarrow \infty$.

Medidas Pontuais	Medidas Assimptóticas	Valores Esperados
Fiabilidade		MTTF
Segurança (<i>Safety</i>)		MST
Disponibilidade	Disponibilidade	MUT
Indisponibilidade	Indisponibilidade	MTBF
Capacidade de manutenção		MDT

Tabela 2.2 - Medidas da confiança no funcionamento.

As medidas de fiabilidade são tipicamente empregues para avaliar sistemas que são fortemente sensíveis à ocorrência de avarias. Estes sistemas são usualmente denominados como sistemas orientados à missão (*mission oriented systems*) [MB99]. Exemplos destes sistemas podem ser encontrados no controlo de voo de aviões, nas missões espaciais, bem como nos sistemas de protecção dos reactores nucleares.

As medidas de disponibilidade são tipicamente empregues para avaliar sistemas onde pequenas interrupções de serviço podem ser toleradas. Exemplos destes sistemas podem ser encontrados no comercio electrónico, processos industriais e telecomunicações.

⁵ Estes termos são aqui utilizados no seu sentido mais lato.

2.4.1 Relações Fundamentais

A forma mais simples de definir as medidas de confiança no funcionamento é assumir, sem perda de generalidade, que o comportamento do sistema evolui ao longo de um conjunto de estados que podem ser agrupados em dois conjuntos complementares: Operacional (**OP**), correspondendo ao serviço correcto, e avaria (**AV**) correspondendo ao serviço incorrecto. A alternância entre estes dois estados é governada respectivamente pelas avarias e reparações do sistema (Fig. 2.7).

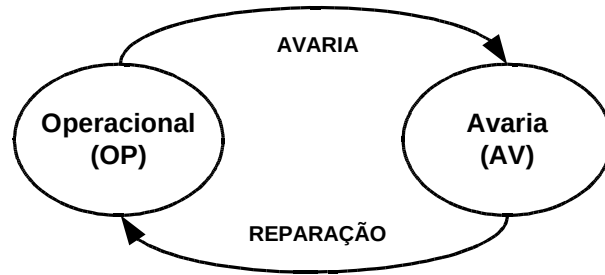


Figura 2.7 - Alternância entre estados do sistema.

2.4.1.1 Probabilidade de Avaria

A probabilidade do sistema avariar no intervalo de tempo $[0, t[$ corresponde ao complementar da probabilidade de ocupar o estado **OP** durante esse intervalo, sendo definida como

$$Prob\{T \leq t\} = F_T(t), \quad t \geq 0 \quad (2.1)$$

onde T é a variável aleatória que representa o instante da ocorrência da avaria do sistema e $F_T(t)$ a respectiva função distribuição de probabilidade.

2.4.1.2 Taxa de Avaria

A taxa à qual ocorrem as avarias do sistema $\lambda(t)$ (*hazard rate*), corresponde à taxa de transição entre o estado **OP** e o estado **AV**, sendo definida a partir de uma probabilidade condicionada

$$\lambda(t) = \lim_{\Delta t \rightarrow 0} \frac{1}{\Delta t} Prob[\text{Avariar em } [t, t + \Delta t] \mid \text{não avariou em } [0, t]] \quad (2.2)$$

ou de outra forma

$$\lambda(t) = \frac{f_T(t)}{1 - F_T(t)} \quad (2.3)$$

⁶ A abreviatura de *Probabilidade* (*Prob*) será utilizada em todas as equações nesta dissertação.

em que $f_T(t)$ a função densidade de probabilidade das avarias, sendo definida como

$$f_T(t) = \frac{\partial}{\partial t} F_T(t) \quad (2.4)$$

Dada a importância que $\lambda(t)$ assume no comportamento do sistema, a sua caracterização têm sido alvo de um elevado número de estudos [Dhi79] [SS94] [Won95] [SMN03] [Eco04]. De facto, sem um conhecimento rigoroso de $\lambda(t)$ qualquer avaliação da confiança no funcionamento perde qualquer credibilidade por estar baseada em pressupostos incorrectos.

Historicamente o comportamento de $\lambda(t)$ referente a um equipamento genérico tem sido modelada utilizando a tradicional curva em forma de banheira (*bathtub*) [Dhi79] [SS94] [KKW03], cujo comportamento é decalcado do conceito de taxa de mortalidade humana (Fig. 2.8).

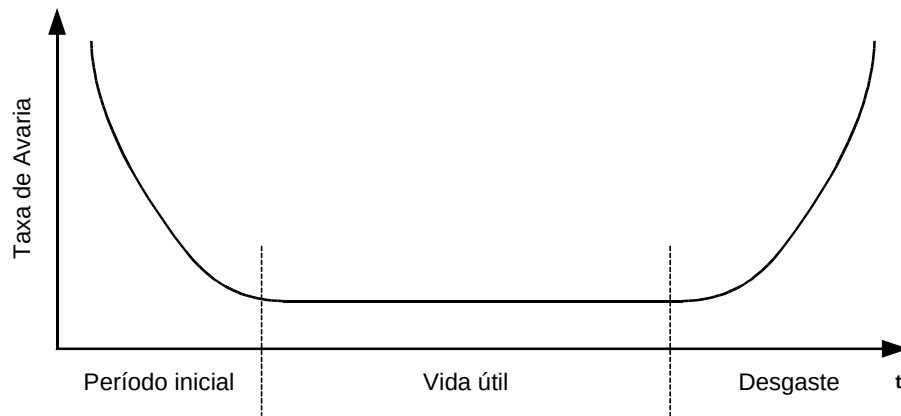


Figura 2.8 - Evolução típica da taxa de avarias.

Durante o período inicial de vida (também denominado como mortalidade infantil ou *burn in*) as avarias são do tipo endógeno e resultam de defeitos no sistema atribuídos a falhas de concepção, fabrico e montagem. Durante este período é esperado que a taxa de avarias decresça com o tempo. Posteriormente à remoção dos defeitos iniciais, as avarias são do tipo exógeno e geralmente associadas às condições ambientais em que o sistema opera. Este período representa a fase de vida útil do sistema, e a taxa de avaria pode considerar-se aproximadamente constante. Quando o sistema se aproxima do fim da sua vida útil, a taxa de avaria aumenta e é dito que o sistema entrou na sua fase de desgaste.

Dada a natureza dos fenómenos em causa, a caracterização de $\lambda(t)$ nem sempre é simples. A estratégia normalmente utilizada passa por estabelecer um modelo analítico, baseado numa função distribuição de probabilidade, que caracteriza a ocorrência das avarias, e posteriormente pela obtenção dos valores dos parâmetros deste modelo. Estes modelos são tipicamente desenvolvidos a partir do

conhecimento real do comportamento do sistema, ou dos processos físicos subjacentes ao mesmo. São assim muito dependentes das características do sistema em causa. Por exemplo, comportamento de um sistema com componentes eléctricos ou electrónicos será distinto de um sistema com componentes mecânicos. O mesmo se aplica a sistemas que incluam *software*.

Embora existam um grande número de modelos no contexto dos sistemas abordados nesta dissertação, são comuns utilizar as seguintes distribuições [Dhi99] [Whi02]:

- ♦ **Distribuição Exponencial.** Representa o comportamento durante a fase de vida útil do sistema, caracterizando-se por assumir uma taxa de avaria constante. É especialmente utilizada em sistemas com componentes eléctricos ou electrónicos, verificando-se experimentalmente que é uma boa aproximação;
- ♦ **Distribuição de Weibull.** Permite representar qualquer das três áreas da curva. É geralmente utilizada na área da micro-electrónica para representar a mortalidade infantil;
- ♦ **Distribuição Log-Normal.** Permite representar qualquer das três áreas da curva. É geralmente utilizada na área da micro-electrónica para representar o desgaste;
- ♦ Modelos específicos utilizados ao nível do *software* (*Software Reliability Growth Models*) [Goe85] [Yam00].

Os parâmetros para os modelos podem ser obtidos de duas formas. A mais exacta, mas também a mais complexa, passa por observar as características das avarias em sistemas semelhantes e posteriormente proceder a um tratamento estatístico que permite obter os parâmetros da distribuição. A alternativa, passa por utilizar dados recolhidos e compilados por várias organizações, que permitam estimar os parâmetros do modelo. Esta é opção mais simples e a mais utilizada. No entanto é importante perceber que a informação obtida desta forma não deve ser encarada como totalmente exacta. O seu objectivo enquadra-se mais na definição de uma base de trabalho, que permita estabelecer uma análise comparativa entre várias alternativas.

Dentro desta última abordagem a documentação existente é muito grande e variada [Dhi99] e não cabe nos objectivos desta dissertação discuti-la. Contudo, não é possível deixar de referir o documento mais utilizado: o MIL-HDBK-217 *Reliability Prediction of Electronic Equipment* [MIL91]. Este documento disponibiliza modelos para calcular as taxas de avarias de um grande número de componentes eléctricos e electrónicos. Estes modelos assumem uma taxa de avarias constante (distribuição exponencial), e cujo valor é função das características dimensionais do

componente e do ambiente em que opera. Por exemplo, para o caso de um microprocessador a taxa de avarias λ é dada por

$$\lambda = (C_1 \pi_T + C_2 \pi_E) \cdot \pi_Q \pi_L \quad (2.5)$$

em que C_1 representa taxa de avarias associada à complexidade do circuito (número de transístores ou *gates*), π_T o factor de temperatura, C_2 a taxa de avarias associada ao tipo de encapsulamento do circuito (*package*), π_E o factor ambiental (tipo de ambiente a que está exposto), π_Q o factor de qualidade de fabrico e π_L o factor de aprendizagem (*learning*), associado ao número de anos em que o circuito está em produção.

2.4.1.3 Fiabilidade

A *fiabilidade* $R(t)$ ⁷ (*Reliability*) corresponde à probabilidade de não ter ocorrido qualquer avaria do sistema no intervalo $[0, t[$, isto é a uma presença contínua no estado **OP**.

$$R(t) = \text{Prob} \{ \text{Sistema não avariar em } [0, t[\} \quad (2.6)$$

ou seja

$$R(t) = \text{Prob}(T > t) \quad (2.7)$$

Em alternativa, pode também ser definida a partir da função densidade de probabilidade de avarias $f_T(t)$ como

$$R(t) = (1 - F_T(t)) = \int_t^\infty f_T(t) dt \quad (2.8)$$

ou após algumas manipulações

$$R(t) = \exp \left[- \int_0^t \lambda(t) dt \right] \quad (2.9)$$

2.4.1.4 Segurança

A *segurança* $S(t)$ (*Safety*) é expressa de forma semelhante à fiabilidade. A diferença principal está unicamente semântica relacionada com a definição dos estados **OP** e **AV**. Neste caso considera-se que o estado **OP** (**AV**) representa o estado seguro (não-seguro) do sistema.

⁷ Optou-se por utilizar a simbologia tradicional da literatura.

2.4.1.5 Disponibilidade

A disponibilidade $A(t)$ (*Availability*) é definida como a probabilidade do sistema estar a operar correctamente no instante t , ou seja em estar presente no estado **OP** nesse instante.

$$A(t) = \text{Prob}\{\text{Sistema estar operacional em } t\} \quad (2.10)$$

A indisponibilidade $\bar{A}(t)$ (*Unavailability*) é definida como o complemento da disponibilidade

$$U(t) = 1 - \bar{A}(t) \quad (2.11)$$

2.4.1.6 Taxa de Reparação

A taxa a que o sistema é reparado $\mu(t)$ (*repair rate*), corresponde à taxa de transição entre o estado **AV** e o estado **OP**, sendo definida a partir de uma probabilidade condicionada

$$\mu(t) = \lim_{\Delta t \rightarrow 0} \frac{1}{\Delta t} \text{Prob}\{\text{Sistema reparado em } [t, t + \Delta t] \mid \text{está avariado em } [0, t]\} \quad (2.12)$$

2.4.1.7 Capacidade de Manutenção

A capacidade de manutenção $M(t)$ (*Maintainability*) é definida como o complementar da probabilidade do sistema não ser reparado no intervalo $[0, t]$, dado que estava avariado no instante $t=0$, isto é a uma presença em **OP** quando inicialmente estava em **AV**.

$$M(t) = 1 - \text{Prob}\{\text{Sistema não ser reparado em } [0, t]\} \quad (2.13)$$

De forma análoga a (2.9)

$$M(t) = 1 - \exp\left[-\int_0^t \mu(t) dt\right] \quad (2.14)$$

2.4.1.8 Valores Esperados

Uma outra forma de caracterizar as medidas de confiança no funcionamento é através dos respectivos valores esperados (valores médios):

- ♦ MTTF^8 (*Mean Time to Failure*): tempo médio de funcionamento até à primeira avaria

⁸ Optou-se por utilizar a terminologia em Inglês por se julgar mais adequada ao contexto.

$$MTTF = \int_0^{\infty} R(t) dt \quad (2.15)$$

- ♦ MST (*Mean Safe Time*): pode ser definido de forma equivalente ao MTTF, mas considerando a avaria como um estado seguro do sistema;
- ♦ MTTR (*Mean Time to Repair*): tempo médio da reparação completa do sistema

$$MTTR = \int_0^{\infty} 1 - M(t) dt \quad (2.16)$$

- ♦ MUT (*Mean Up Time*): tempo médio de funcionamento após a reparação. Embora muito semelhante ao MTTF, contempla as situações em que após uma reparação é possível identificar estados operacionais em que existem alguns componentes do sistema do que ainda não foram reparados;
- ♦ MDT (*Mean Down Time*): tempo médio de duração de uma avaria. Embora muito semelhante ao MTTR, contempla as situações em que é possível identificar diferentes estados de avaria do sistema com diferentes tempos de reparação;
- ♦ MTBF (*Mean Time Between Failures*): tempo médio entre duas avarias.

$$MTBF = MUT + MDT \quad (2.17)$$

Em termos assintóticos é possível verificar a seguinte relação [PG80]

$$A_{\infty} = \lim_{t \rightarrow \infty} A(t) = \frac{MTTF}{MTTF + MTTR} \quad (2.18)$$

2.5 Processos Estocásticos

A forma mais adequada de analisar o comportamento estocástico de um sistema é através da definição de uma variável aleatória $X(t)$ que descreve o estado do sistema no instante de tempo t .

A família de variáveis aleatórias $X(t)$ constitui um processo estocástico

$$\mathbf{X} = \{X(t) \mid t \in \mathbf{T}\} \quad (2.19)$$

que está definido num espaço de probabilidades $X(t) \in \mathbf{S}$ indexado pelo parâmetro t , onde t toma valores num conjunto ordenado $\mathbf{T}$ [Kul95] [Kao97] [Triv02]. Neste contexto o espaço de probabilidades $\mathbf{S}$ é interpretado como o espaço de

estados do sistema (conjunto de valores que o estado do sistema pode assumir), e o parâmetro t como o tempo.

Os processos estocásticos podem ser classificados de várias formas em função das suas propriedades. Uma primeira classificação pode ser realizada com base nas características de $\mathbf{S}$ e de $\mathbf{T}$. O espaço de estados $\mathbf{S}$ pode ser discreto (o caso mais comum), contínuo ou uma mistura de ambos. O conjunto $\mathbf{T}$ pode ser discreto ou contínuo, conduzindo a processos estocásticos de tempo discreto e de contínuo, respectivamente.

O gráfico da evolução de $X(t)$ em função de t é denominado *realização amostral* (*sample path*) do processo estocástico. Esta evolução é caracterizada por transições que correspondem à mudança de estado do sistema (Fig. 2.9). Estas transições efectuam-se quando ocorrem *eventos* (aleatórios) que afectam o comportamento do sistema.

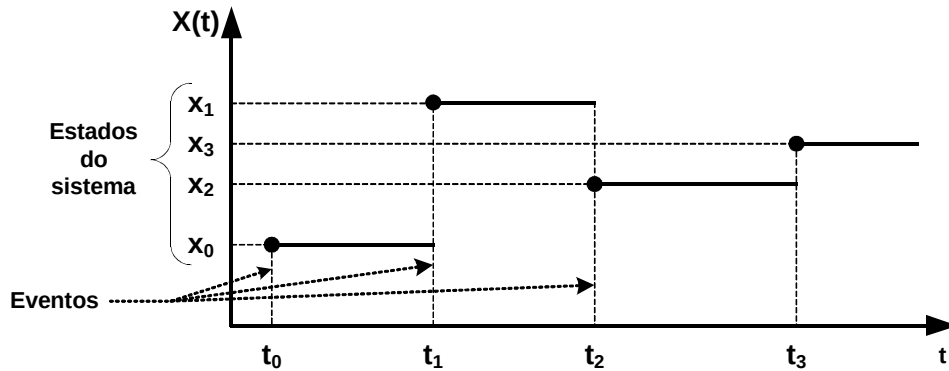


Figura 2.9 - Exemplo da realização de um processo estocástico.

Uma segunda classificação dos processos estocásticos pode ser estabelecida a partir de conjunto de propriedades adicionais. A propriedade de Markov indica que a evolução futura do processo depende apenas do seu presente e não do seu passado, isto é que o processo não possui memória (*memoryless*)

$$\begin{aligned} & \text{Prob}\{X(t)=x \mid X(t_n)=x_n, X(t_{n-1})=x_{n-1}, \dots, X(t_0)=x_0\} \\ &= \text{Prob}\{X(t)=x \mid X(t_n)=x_n\} \end{aligned} \quad (2.20)$$

$$t_0 < t_1 < \dots < t_n < t, \quad t, t_n, t_{n-1}, \dots, t_0 \in \mathbf{T}, \quad x, x_n, x_{n-1}, \dots, x_0 \in \mathbf{S}$$

Os processos estocásticos que respeitam esta propriedade para todos os instantes de tempo (discreto ou contínuo) denominam-se **Markovianos**, sendo os restantes naturalmente denominados **Não-Markovianos**.

A invariância em relação à origem dos tempos é denominada **homogeneidade temporal**, e indica que o comportamento do processo é independente do início da observação do mesmo

$$\begin{aligned} Prob\{X(t)=x \mid X(t_n)=x_n\} &= Prob\{X(t-t_n)=x \mid X(0)=x_n\} \\ \forall t, t_n \in T, \forall x, x_n \in S \end{aligned} \quad (2.21)$$

Dado que a definição de um processo estocástico é baseada em variáveis aleatórias, é possível aplicar o conceito de funções distribuição e densidade de probabilidade.

Desta forma $\pi_i(t)$

$$\pi_i(t) = Prob\{X(t)=i\}, \quad i \in S \quad (2.22)$$

define a probabilidade do processo estocástico estar no estado $i \in S$ no instante t , sendo referida como a probabilidade transitória (*transient*) do estado i . Se o limite

$$\pi_i = \lim_{t \rightarrow \infty} \pi_i(t) \quad (2.23)$$

existir, π_i é referida como a probabilidade estacionária do estado i . No caso do limite anterior não existir, π_i pode também ser definida como o valor médio temporal do limite

$$\pi_i = \lim_{t \rightarrow \infty} \frac{1}{t} \int_0^t \pi_i(x) dx \quad (2.24)$$

Na maioria das situações o valor médio temporal do limite existe. Se ambos os limites existem, então são iguais. Ambos podem ser intuitivamente interpretados como a fracção do tempo que o processo passa num determinado estado quando é observado por um período de tempo muito longo.

Dada a importância dos processos que respeitam a propriedade de Markov (em todos os instantes do tempo, ou em apenas em pontos particulares) assumem nesta dissertação, proceder-se-á nas secções seguintes a uma discussão dos vários processos deste tipo e das respectivas propriedades⁹ (Fig. 2.10 [Lin98]).

São vários os objectivos que se pretendem atingir nas próximas secções. Em primeiro lugar, apresentar as principais características dos processos que estão subjacentes aos modelos de avaliação da confiança no funcionamento propostos nesta dissertação. Em segundo lugar, discutir o tipo de processos estocásticos

⁹ Na discussão realizada nas secções seguintes assume-se, até ressalva em contrário, que os processos são homogéneos no tempo e que possuem um espaço de estados discreto e finito.

suportados pelos formalismos de modelação escolhidos: Cadeias de Markov e redes de Petri Estocásticas. Por fim, formalizar o tratamento analítico destes mesmos processos.

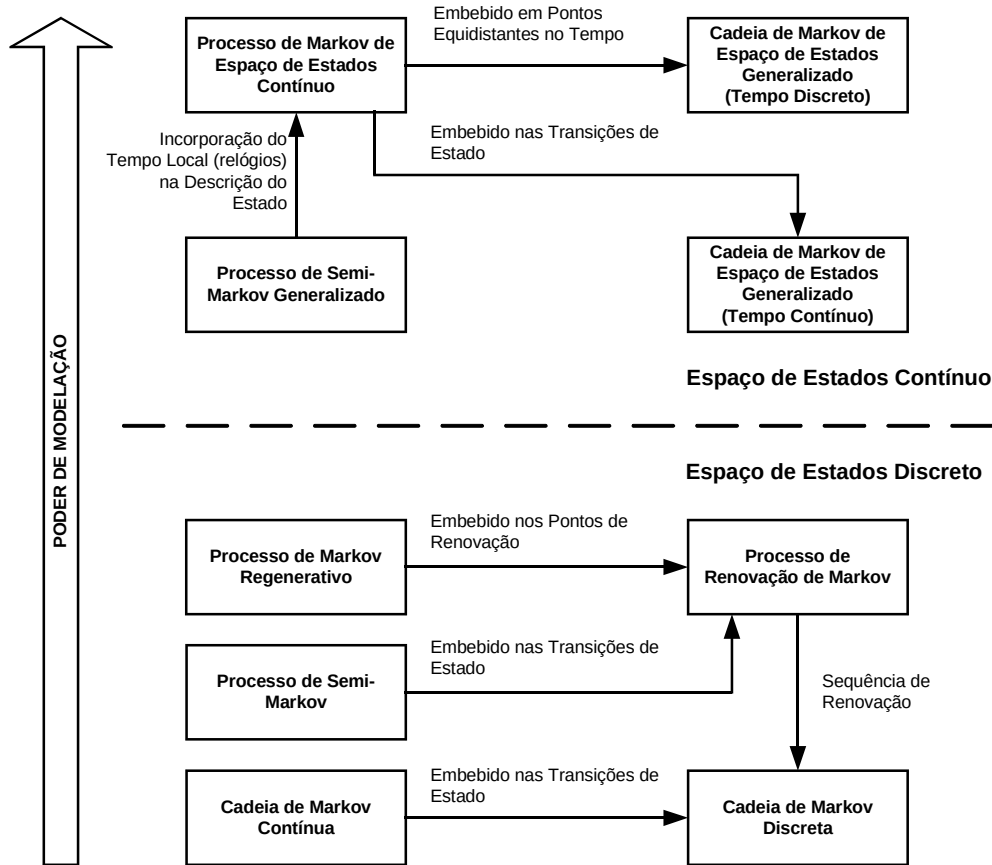


Figura 2.10 - Relações e hierarquia entre as várias classes de processos de Markov.

2.5.1 Processos de Renovação

Considere-se um processo estocástico $X = \{X(t) | t \in T\}$ do qual interessa analisar a ocorrência de um evento particular do sistema (ex. um estado específico).

Um *processo de renovação* (*renewal process*) é um processo de tempo discreto definido por uma sequência de variáveis aleatórias

$$\{H_n\} \quad n \in \mathbb{N}_0 \quad (2.25)$$

em que os intervalos de tempo entre ocorrências sucessivas $H_n = T_{n+1} - T_n$ são variáveis aleatórias independentes e identicamente distribuídas (Fig. 2.11). Os instantes do tempo $T_0, T_1, \dots, T_n$ onde se verifica a ocorrência do evento que interessa analisar são denominados *instantes de renovação*. Um instante de tempo é denominado *instante de regeneração* se, para além de ser um instante de renovação, se

o tempo médio (valor esperado) da primeira visita ao estado em que o processo reside no instante de renovação é finito (o processo regenera-se).

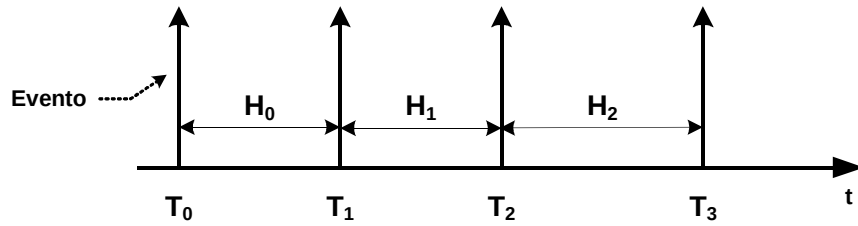


Figura 2.11 - Realização de um processo de renovação.

Se a observação do sistema não se inicia em $T_0=0$, o processo estocástico é denominado processo de renovação *atrasado* (*delayed renewal process*).

2.5.1.1 Processo de Poisson

Um dos processos de renovação mais comuns é o *processo de Poisson* [Triv02]. De acordo com este processo, os intervalos de tempo entre ocorrências sucessivas são definidos por uma função distribuição exponencial

$$F(x) = 1 - \exp(-\lambda x) \quad (2.26)$$

Esta distribuição caracteriza-se por possuir um valor esperado igual a $1/\lambda$, com uma variância de $1/\lambda^2$. Desta forma, o intervalo de tempo entre duas renovações sucessivas é $1/\lambda$. Pode demonstrar-se [Kul95] [Triv02] que esta é a única distribuição de tempo contínuo que não tem memória (a sua “equivalente” em tempo discreto é a distribuição geométrica).

A probabilidade $Prob(n, t)$ de ocorrerem n renovações (eventos) num intervalo de tempo $]0...t]$ é

$$Prob(n, t) = e^{-\lambda t} \cdot \frac{(\lambda t)^n}{n!} \quad n > 0 \quad (2.27)$$

em que λ pode também ser interpretado como a taxa à qual ocorrem as renovações.

Este processo desempenha um papel muito importante na avaliação da confiança no funcionamento, nomeadamente na modelação de falhas com origem física. Uma das razões para a sua importância, está relacionada com a facilidade do seu tratamento analítico. Uma outra razão, é o resultado da demonstração que em condições muito gerais a soma de um número elevado de processos de renovação independentes se comporta como um processo de Poisson [Triv02].

Neste sentido, é importante apresentar as propriedades mais importantes deste processo (Fig. 2.12):

- ♦ A sobreposição de vários processos de Poisson independentes resulta num processo de Poisson cuja taxa é a soma das taxas dos processos anteriores;
- ♦ Um processo de Poisson pode ser decomposto em vários processos de Poisson independentes através da definição de uma escolha probabilística associada à ocorrência de uma renovação. Cada um destes novos processos, possui uma taxa a que é igual à probabilidade da ocorrência multiplicada pela taxa original.

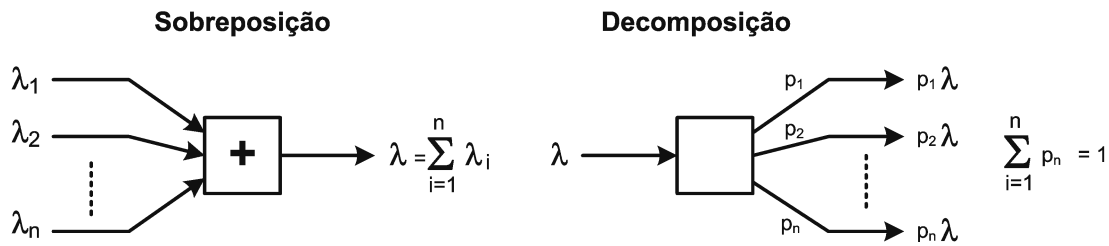


Figura 2.12 - Propriedades do processo de Poisson.

2.5.2 Processos de Renovação de Markov

Seja $\mathbf{T} = \{T_n | n \geq 0\}$ uma sequência de variáveis aleatórias não decrescentes com $0 = T_0 \leq T_1 \leq T_2 \dots$, e $\mathbf{X} = \{X_n | n \geq 0\}$ um processo estocástico com espaço de estados discreto $\mathbf{S}$.

Um *processo de renovação de Markov* [Kul95] [Triv02], é um processo estocástico de tempo discreto $(\mathbf{X}, \mathbf{T}) = \{(X_n, T_n) | n \geq 0\}$ em que

$$\begin{aligned} & \text{Prob}\{X_{n+1}=j, T_{n+1}-T_n \leq t | X_n=i, \dots, X_0=0; T_n=t_n, \dots, T_0=0\} \\ & = \text{Prob}\{X_{n+1}=j, T_{n+1}-T_n \leq t | X_n=i\} \end{aligned} \tag{2.28}$$

$$X_n = X(t = T_n^+)$$

$$n \in \mathbb{N}_0, \quad t \geq 0, \quad j, i \in \mathbf{S}$$

A variável aleatória X_n representa o estado do processo imediatamente a seguir à n -ésima transição, que por sua vez ocorre no instante T_n . Este processo satisfaz a propriedade de Markov apenas nos instantes T_n , e não para $\forall t \geq 0$. Isto significa que o tempo de estadia H_n num estado possui uma função distribuição arbitrária (ou generalizada), que é função dos estados de partida e chegada (Fig 2.13).

O processo $T = \{T_n | n \geq 0\}$ não é um processo de renovação em si, porque as variáveis $T_{n+1} - T_n$ não são independentes e identicamente distribuídas. No entanto, este pode ser visto como uma sobreposição de vários processos de renovação resultantes das visitas aos vários estados do processo.

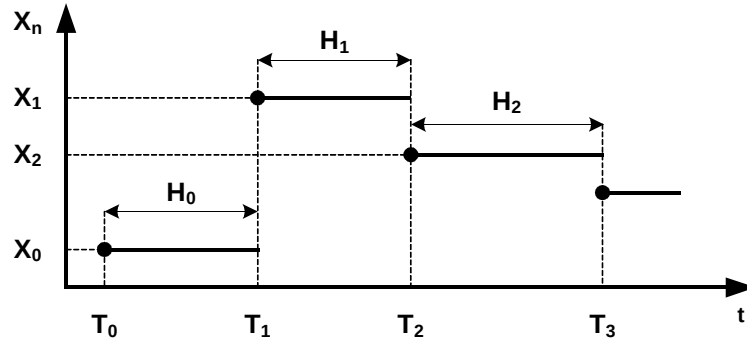


Figura 2.13 - Realização de um processo de Renovação de Markov.

O comportamento deste processo é definido por uma família de matrizes estocásticas denominadas *núcleo (kernel)* e definidas como

$$\begin{aligned}
 k_{ij}(t) &= \text{Prob}\{X_{n+1}=j, T_{n+1}-T_n \leq t \mid X_n=i\} \\
 0 &\leq k_{ij}(t) \leq 1 \quad \forall i, j \in S \\
 \sum_{j \in S} k_{ij}(t) &= 1 \quad \forall i \in S
 \end{aligned}
 \tag{2.29}$$

O núcleo especifica a função distribuição do tempo até à próxima renovação, que é definido como

$$\text{Prob}\{T_{n+1}-T_n \leq t \mid X_n=i\} = \sum_{j \in S} k_{ij}(t)
 \tag{2.30}$$

Embebido neste processo encontra-se uma cadeia de Markov Discreta $X = \{X_n | n \geq 0\}$ (§2.5.5.1) que governa as transições entre estados e cuja matriz transição de probabilidades $P = [p_{ij}]$ é definida como

$$p_{ij} = \lim_{t \rightarrow \infty} k_{ij}(t)
 \tag{2.31}$$

A matriz $K(t)$ e o vector de probabilidades iniciais $p_i = \text{Prob}\{X_0=i\}$ definem completamente o processo estocástico.

Os processos de renovação de Markov são muito importantes pois encontram-se directamente embebidos na definição de vários processos estocásticos Markovianos e Não-Markovianos, definindo assim muitas das propriedades e características

destes últimos. Nas próximas secções estes aspectos serão discutidos com maior nível de detalhe.

2.5.3 Processos de Markov Discretos

Um *processo de Markov de tempo discreto* [Kul95] [Triv02], é definido por uma sequência de variáveis aleatórias $\{X_n | n \geq 0\}$ que tomam valores em S e que satisfazem a propriedade de Markov da seguinte forma

$$\begin{aligned} \text{Prob}\{X_{n+1}=j | X_n=i, X_{n-1}=i_{n-1}, \dots, X_0=i_0\} &= \text{Prob}\{X_{n+1}=j | X_n=i\} \\ n \in \mathbb{N}_0, \quad \forall i, j \in S \end{aligned} \quad (2.32)$$

Este é um processo Markoviano, sendo usualmente denominado por *cadeia de Markov Discreta*. Como resultado da propriedade de Markov (válida para $\forall T_n \ n \geq 0$), demonstra-se que o tempo de ocupação H_i de um estado i , isto é o tempo que o processo passa no estado i durante uma visita particular a esse estado, tem uma função distribuição do tipo geométrica.

Um processo de Markov discreto pode ser visto como uma sobreposição de vários processos de renovação resultantes das visitas aos vários estados do processo.

2.5.4 Processos de Markov Contínuos

Um *processo de Markov de tempo contínuo* [Kul95] [Triv02], é definido por uma sequência de variáveis aleatórias $\{X(t) | t \geq 0\}$ que tomam valores em S e que satisfazem a propriedade de Markov da seguinte forma

$$\begin{aligned} \text{Prob}\{X(t+u)=j | X(v): 0 \leq v < t, X(t)=i\} &= \text{Prob}\{X(t+u)=j | X(t)=i\} \\ t, u \in \mathbb{R}_0^+, \quad \forall i, j \in S \end{aligned} \quad (2.33)$$

Este é um processo Markoviano, sendo usualmente denominado por *cadeia de Markov Contínua*. Como resultado da propriedade de Markov (válida para $\forall t \geq 0$), demonstra-se que o tempo de ocupação H_i de um estado i tem uma função distribuição do tipo exponencial.

Embebida numa cadeia de Markov Contínua encontra-se uma cadeia de Markov Discreta, definida nos instantes em que ocorrem as transições entre estados.

2.5.5 Análise de Processos Markovianos

A evolução dos processos Markovianos pode obter-se através de argumentos probabilísticos simples, nomeadamente utilizando o equilíbrio dinâmico do fluxo de probabilidades entre estados, dos quais resultam as equações de Chapman-Kolmogorov [Kul95] [Triv02], aplicáveis quer a processos de tempo discreto, quer de tempo contínuo¹⁰

$$\begin{aligned}
 p_{ij}(t) &= \text{Prob} \{ X(t+u) = j \mid X(u) = i \} \\
 \sum_{j \in S} p_{ij}(t) &= 1 \\
 p_{ij}(t+u) &= \sum_{r \in S} p_{ir}(t) \cdot p_{rj}(u) \\
 t, u &\in \mathbb{R}_0^+, \quad \forall i, j \in S
 \end{aligned}
 \tag{2.34}$$

2.5.5.1 Cadeias de Markov Discretas

Numa cadeia de Markov Discreta a evolução do processo é definida a partir de uma matriz estocástica de transição de probabilidades $\mathbf{P} = [p_{ij}]$ em que

$$\begin{aligned}
 p_{ij} &= \text{Prob} [X_{n+1} = j \mid X_n = i] \\
 0 &\leq p_{ij} \leq 1 \quad \forall i, j \in S \\
 \sum_{j \in S} p_{ij} &= 1 \quad \forall i, j \in S
 \end{aligned}
 \tag{2.35}$$

Como resultado da propriedade de Markov, as probabilidades p_{ij} caracterizam um processo sem memória, em que a evolução futura só depende do estado presente. Resulta desta forma, que o tempo de estadia em cada estado é definido por uma distribuição geométrica com parâmetro $(1 - p_{ii})$.

É comum representar graficamente uma cadeia de Markov Discreta através de um diagrama de transição de estados (Fig. 2.14). Um estado $i \in S$ é representado por um círculo com a respectiva denominação. As transições entre estados são representadas por arcos, aos quais estão associados as respectivas probabilidades de transição p_{ij} .

¹⁰ Aqui são apenas apresentadas as equações de tempo contínuo.

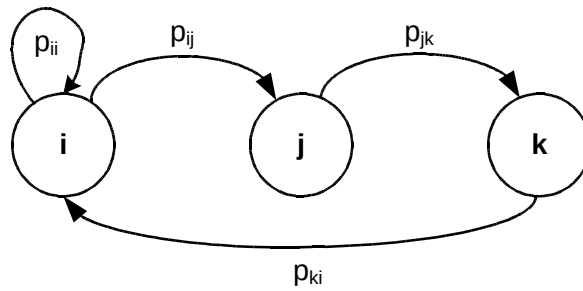


Figura 2.14 - Exemplo de uma cadeia de Markov Discreta.

O comportamento transitório de uma cadeia de Markov Discreta é dado por

$$\boldsymbol{\pi}(n+1) = \boldsymbol{\pi}(n) \mathbf{P}$$

ou

(2.36)

$$\boldsymbol{\pi}(n) = \boldsymbol{\pi}(0) \mathbf{P}^n$$

em que $\mathbf{P} = [p_{ij}]$ é a matriz de transição de probabilidades, $\boldsymbol{\pi}(0) = [\pi_i(0)]$ o vector da distribuição inicial e $\boldsymbol{\pi}(n) = [\pi_i(n)]$ o vector de probabilidades dos estados ao fim de n transições.

O comportamento estacionário do sistema ($n \rightarrow \infty$) é função de um conjunto de propriedades associadas aos estados da cadeia de Markov Discreta, das quais se destacam as seguintes:

- ♦ A cadeia é dita *irredutível* se qualquer estado pode ser alcançado de qualquer outro estado com probabilidade maior de que zero. Caso contrário, é dita *reduzível*. Uma cadeia é irredutível se o seu gráfico de transição de estados for fortemente conexo;
- ♦ Os estados de uma cadeia reduzível podem ser agrupados em classes denominadas *transitórias* e *recorrentes*. Um estado é dito recorrente se, partindo desse estado, o processo retorna a esse estado¹¹ com probabilidade 1. Caso contrário, é dito transitório;
- ♦ Um subconjunto de $\mathbf{A}$ é dito fechado se não é possível qualquer transição para fora desse conjunto. Uma classe recorrente é um subconjunto de $\mathbf{A}$ que não contém quaisquer outros subconjuntos fechados. Assim, isoladamente, uma classe recorrente é irredutível. Todos os estados de $\mathbf{A}$ que não pertencem a classes recorrentes estão agrupados em classes transitórias. Uma cadeia de Markov Discreta consiste assim numa, ou mais, classes recorrentes e numa, eventualmente vazia, classe transitória;

¹¹ O tempo de médio de recorrência pode ser finito (*positive recurrent*) ou infinito (*null recurrent*).

- ♦ As classes recorrentes constituídas por único estado são denominados estados *absorventes*;
- ♦ O período de um estado recorrente i é definido como o maior divisor comum de um conjunto de inteiros n de forma que $p_{ii}(n) > 0$. Um estado cujo período é $n=1$ é dito aperiódico. A cadeia é *aperiódica* se todos os seus estados são aperiódicos, caso contrário é periódica;
- ♦ Numa cadeia irredutível todos os estados são periódicos ou aperiódicos. Para se verificar o último caso basta que $\exists i \in S : p_{ii} > 0$;
- ♦ Uma cadeia é dita *ergódica* se for irredutível, aperiódica e recorrente (finita). Nos problemas de análise de confiança no funcionamento, este tipo de cadeia corresponde ao caso mais frequentemente encontrado.

O comportamento estacionário ($n \rightarrow \infty$) nem sempre está definido, isto é a probabilidade limite pode não existir

$$\pi_i = \lim_{n \rightarrow \infty} \pi_i(n) \quad (2.37)$$

Contudo, e centrando-nos no caso mais frequente, para uma cadeia ergódica o limite existe e é dado por

$$\begin{aligned} \pi &= \pi P \\ \sum_{i \in S} \pi_i &= 1 \end{aligned} \quad (2.38)$$

em que $\pi = [\pi_i]$ é o vector das probabilidades estacionárias dos estados. Neste caso o comportamento estacionário é independente de $\pi(0)$.

No caso de uma cadeia periódica é apenas possível definir o valor médio das probabilidade limite, que por sua vez são iguais às probabilidades estacionárias, sendo contudo dependentes de $\pi(0)$.

No caso de uma cadeia com estados absorventes, a probabilidade estacionária não é normalmente uma medida de muito interesse, dado que nos estados transitórios será 0 e no absorvente¹² 1. Neste caso dá-se mais relevância ao tempo de absorção, que é definido como o tempo, partindo de um estado inicial, que o processo demora até atingir um estado absorvente, tratando-se, também de uma variável aleatória. É comum caracteriza-lo através do valor esperado e da variância, embora seja possível também obter, com maior complexidade, a sua distribuição.

¹² Se existir mais do que um estado absorvente, as probabilidades associadas a cada estado podem ter interesse em determinar.

A solução de uma cadeia de Markov Discreta, pode ser obtida através de métodos simbólicos, deduzindo as respectivas expressões analíticas, ou numéricos, através do cálculo numérico das equações associadas ao comportamento da cadeia. Os métodos existentes podem ser classificados em função do tipo de análise pretendida [MMT94] [MMT96] [MS97] [Kao97] [Ger00] [Triv02] [Kri04]:

♦ **Análise transitória**

- **Simbólicos.** Utilização da transformada Z e da respectiva transformada inversa para obter a solução no domínio do tempo discreto. Esta técnica é apenas possível de aplicar eficazmente em cadeias com um reduzido espaço de estados;
- **Numéricos.** Método da potência (*power method*), através da multiplicação sucessiva de $\pi(n)$ por P .

♦ **Análise estacionária**

- **Simbólicos.** Utilização da transformada Z (Teorema do Valor Final), e análise das condições de equilíbrio do fluxo probabilístico entre estados;
- **Numéricos.** Através da utilização de métodos para a resolução de sistemas de equações lineares, tais como: eliminação Gaussiana, potência (*power method*), Gauss-Seidel e sobre-relaxação sucessiva.

2.5.5.2 Cadeias de Markov Contínuas

Numa cadeia de Markov Contínua a evolução do processo também é definida através de uma matriz estocástica de transição de probabilidades $P(t)=[p_{ij}(t)]$ em que

$$p_{ij}(t)=\text{Prob}\{X(t+u)=j \mid X(u)=i\} \quad (2.39)$$

Como a definição de $P(t)$ não facilita a obtenção de expressões analíticas, recorre-se normalmente à definição de uma matriz *geradora infinitesimal* $Q=[q_{ij}]$, relacionada com $P(t)$ da seguinte forma

$$\begin{aligned} q_{ij} &= \lim_{\Delta t \rightarrow 0} \frac{p_{ij}(\Delta t) - \delta_{ij}}{\Delta t}, \quad i \neq j & q_{ii} &= \lim_{\Delta t \rightarrow 0} \frac{p_{ii}(\Delta t) - 1}{\Delta t} \\ q_{ij} &\geq 0 \quad \forall i, j \in S, i \neq j & \sum_{j \in S} q_{ij} &= 0 \quad \forall i \in S \end{aligned} \quad (2.40)$$

em que os seus elementos podem ser interpretados como as taxas de transição entre estados. Assim, q_{ij} corresponde à taxa de transição entre os estados i e j , e q_{ii} ao simétrico da soma de todas as taxas das transições que saem do estado i .

Como resultado da propriedade de Markov, o tempo de ocupação de cada estado é definido por uma distribuição exponencial de parâmetro q_{ii} .

Tal como no caso de uma cadeia de Markov Discreta, uma cadeia de Markov Contínua pode ser representado através de um diagrama de transição de estados (Fig. 2.15). A principal diferença está no facto que associadas às transições entre estados estão as respectivas taxas. Além disso também não existem transições que tenham como ponto de partida e chegada o mesmo estado.

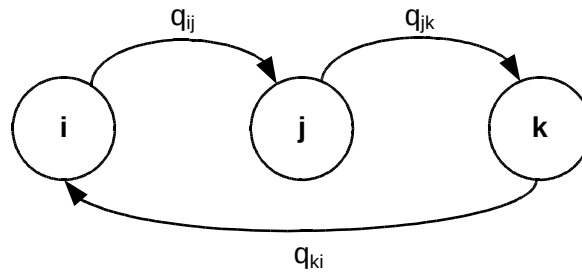


Figura 2.15 - Exemplo de uma cadeia de Markov Contínua.

O comportamento transitório da cadeia de Markov Contínua é dado por

$$\dot{\mathbf{P}}(t) = \mathbf{Q} \cdot \mathbf{P}(t)$$

ou

$$\mathbf{P}(t) = \exp(\mathbf{Q}t)$$

$$\boldsymbol{\pi}(t) = \boldsymbol{\pi}(0) \cdot \mathbf{P}(t)$$

(2.41)

em que $\mathbf{P}(t) = [p_{ij}(t)]$ é a matriz de transição de probabilidades, $\boldsymbol{\pi}(0)$ o vector da distribuição inicial e $\boldsymbol{\pi}(t)$ o vector de probabilidades dos estados no instante t .

Dado que numa cadeia de Markov Contínua existe embebida uma cadeia de Markov Discreta definida nos instantes de transição entre estados, resulta que o conjunto de definições e propriedades desta última sejam também aplicáveis às cadeias de Markov Contínuas. Dito de outra forma, as propriedades da cadeia de Markov Discreta embebida implicam as mesmas propriedades na cadeia de Markov Contínua.

Assim, e centrando-nos no caso mais frequente, para uma cadeia ergódica o comportamento estacionário ($t \rightarrow \infty$) é dado por

$$\boldsymbol{\pi} \cdot \mathbf{Q} = \mathbf{0}$$

$$\sum_{i \in S} \pi_i = 1$$

(2.42)

Tal como no caso discreto, também $\boldsymbol{\pi}=[\pi_i]$ é independente de $\boldsymbol{\pi}(0)$.

Os métodos existentes para obter a solução de uma cadeia de Markov Contínua podem ser classificados de forma análoga ao caso discreto [MMT94] [MMT96] [MS97] [Kao97] [Ger00] [Triv02] [Kri04]:

- ♦ **Análise transitória**

- **Simbólicos.** Utilização da transformada de Laplace e da respectiva transformada inversa para obter a solução no domínio dos tempos. Esta técnica é apenas possível de aplicar eficazmente em cadeias com um reduzido espaço de estados. Uma alternativa, consiste na utilização de métodos semi-simbólicos baseados na determinação dos valores próprios da matriz $\mathbf{Q}$ e na expressão das probabilidades $\boldsymbol{\pi}=[\pi_i]$ como um polinómio de funções exponenciais;
- **Numéricos**
 - Resolução do sistema de equações diferenciais ordinárias, através de métodos como Runge-Kutta e Jansen (uniformização);
 - Utilização das relações existentes entre a cadeia de Markov Discreta embebida na cadeia de Markov Contínua. Assim, resolvendo a cadeia de Markov Discreta, utilizando os métodos previamente descritos, obtêm-se um conjunto de resultados que são utilizados para obter a solução da cadeia de Markov Contínua. São exemplos destes métodos: a equivalência entre primeiros momentos (método aproximado), cadeia de Markov Discreta embebida em instantes equidistantes e nos instantes de de transição e a uniformização da cadeia de Markov Contínua;

- ♦ **Análise estacionária**

- **Simbólicos.** Utilização da transformada de Laplace (Teorema do Valor Final), e análise das condições de equilíbrio do fluxo probabilístico entre estados;
- **Numéricos**
 - Resolução do sistema de equações de equações lineares através de métodos tais como: eliminação Gaussiana, potência (*power method*), Gauss-Seidel e sobre-relaxação sucessiva;
 - Utilização das relações existentes entre a cadeia de Markov Discreta embebida na cadeia de Markov Contínua.

2.5.6 Processos de Semi-Markov

Um *processo de Semi-Markov* [Kul95] [Triv02], é um processo de tempo contínuo $Y(t)$, $t \geq 0$ definido a partir de um processo de Renovação de Markov (X, T) com espaço de estados S , como

$$Y(t) = X_n \quad T_n \leq t < T_{n+1} \quad (2.43)$$

$$n \in \mathbb{N}_0, \quad t \geq 0$$

As visitas aos estados são governadas por uma cadeia de Markov Discreta, $X = \{X_n | n \geq 0\}$, e o tempo de estadia nos mesmos possui uma função distribuição que depende dos estados entre os quais a transição é realizada (Fig. 2.16). Como a propriedade de Markov só se verifica nos instantes T_n (instantes de renovação), este é um processo Não-Markoviano.

Este processo pode ser visto como uma generalização de uma cadeia de Markov Contínua, em que é permitido que o tempo de estadia num estado possua uma função distribuição generalizada, e em que existem transições para o próprio estado.

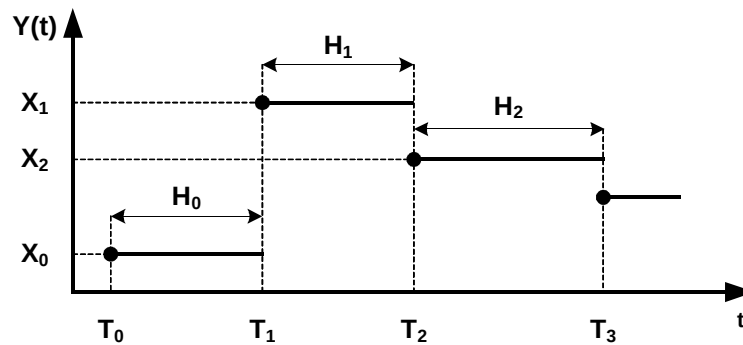


Figura 2.16 - Realização de um processo de Semi-Markov.

Dado que nas transições entre estados, está embebido um processo de Renovação de Markov, a sequência dessas mesmas transições constituem pontos de renovação para o processo de Semi-Markov. Isto implica que ambos os processos possuam o mesmo espaço de estados S .

O comportamento transitório do processo pode ser determinado através da equação do núcleo do processo de Renovação de Markov, enquanto o comportamento estacionário pode ser determinado a partir resolução do sistema de equações lineares da cadeia de Markov Discreta embebida.

2.5.7 Processos de Markov Regenerativos

Um *processo de Markov Regenerativo* [Kul95] [Triv02], é um processo de tempo contínuo $Z(t)$, $t \geq 0$ de espaço de estados $\mathcal{S}$ definido a partir de um processo de Renovação de Markov (X, T) com espaço de estados $\bar{\mathcal{S}}$ como

$$\text{Prob}\{Z(T_n+t)=j \mid Z(u):0 \leq u \leq T_n, X_n=i\} = \text{Prob}\{Z(t)=j \mid X_0=i\} \quad (2.44)$$

$$n \in \mathbb{N}_0, \quad \forall t, u \geq 0 \quad i, j \in \bar{\mathcal{S}}, \quad \bar{\mathcal{S}} \subset \mathcal{S}$$

Este processo não satisfaz a propriedade de Markov em geral, mas existe uma sequência embebida de instantes de tempo T_n em que o estado do processo X_n nesses instantes (de renovação, mas também denominados *de regeneração*) satisfaz a propriedade de Markov (Fig. 2.17). Isto significa que a evolução do processo depende apenas do estado no instante de regeneração, e não da evolução antes da regeneração (torna-se assim uma réplica probabilística a partir deste ponto). Está-se assim também perante um processo Não-Markoviano.

O comportamento do processo entre dois pontos de renovação sucessivos, T_n e T_{n+1} , é definido por um processo estocástico dito *subordinado*. Este processo pode ser qualquer processo estocástico contínuo com espaço de estados $\mathcal{S}$, como por exemplo uma cadeia de Markov Discreta ou até um outro processo Regenerativo de Markov.

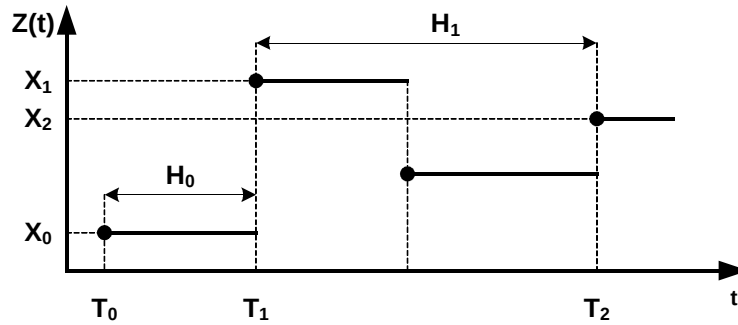


Figura 2.17 - Realização de um processo Regenerativo de Markov.

O processo de Renovação de Markov está embebido em apenas algumas das transições de estados do correspondente processo Regenerativo de Markov. Isto implica que o espaço de estados do processo de Renovação de Markov $\bar{\mathcal{S}}$ seja um subconjunto do espaço de estados do processo Regenerativo de Markov $\mathcal{S}$. Esta propriedade constitui uma importante extensão em relação aos processos de Semi-Markov, onde se verifica que $\mathcal{S} = \bar{\mathcal{S}}$.

O comportamento transitório e estacionário deste processo é obtido de forma semelhante aos processos de Semi-Markov, embora a definição da matriz núcleo seja realizada de forma diferente e baseada nos instantes de regeneração [Kul95].

2.5.8 Processos Generalizados de Semi-Markov

A definição de um processo Generalizado de Semi-Markov assenta na definição prévia de uma cadeia de Markov de Espaço de Estados Generalizado.

Uma *cadeia de Markov de Espaço de Estados Generalizado* (X, C) é uma sequência de variáveis aleatórias $U = \{U_n | n \geq 0\}$ que tomam valores num espaço de estados generalizado definido como

$$U_n \in \{(X_n, C_n)\} \quad n \in \mathbb{N}_0, X_n \in S, C_n \geq 0 \quad (2.45)$$

em que a componente discreta X_n representa o estado do processo na definição clássica, e a componente contínua C_n é um vector que representa o tempo (relativo) associado a um conjunto de eventos (relógios) que ocorrem no estado X_n e que satisfazem a propriedade de Markov da seguinte forma

$$Prob\{U_{n+1} \in \wp | U_n, U_{n-1}, \dots, U_0\} = Prob\{U_{n+1} \in \wp | U_n\} \quad (2.46)$$

em que $\wp$ é um subconjunto do espaço de estados $\{(X_n, C_n)\}$.

Esta definição consiste essencialmente numa redefinição do espaço de estados, mantendo contudo a propriedade de Markov em instantes particulares (processo Não-Markoviano). Trata-se assim de uma generalização do conceito de Cadeia de Markov. O comportamento desta cadeia é completamente definido pela distribuição inicial e por uma matriz (*kernel*) de transição de estados [Lin98].

Um *processo Generalizado de Semi-Markov* [PG89] [Lin98] é um processo estocástico de tempo contínuo $W(t)$ definido a partir de um processo de contagem $N(t)$ sobre uma Cadeia de Markov de Espaço de Estados Generalizado (X, C) como

$$W(t) = X_{N(t)} \quad N(t) = \max_{n \geq 0} \{\zeta_n \leq t\} \quad (2.47)$$

em que ζ_n representa o instante da n -ésima transição de estado, definida por

$$\zeta_n = \sum_{k=0}^{n-1} t(X_k, C_k) \quad (2.48)$$

$$n \in \mathbb{N}_0, \xi_0 = 0$$

em que $t(X_k, C_k)$ representa o tempo (relativo) até à próxima transição de estado.

A Figura 2.18 representa um exemplo da evolução deste tipo de processo no caso de existirem dois relógios.

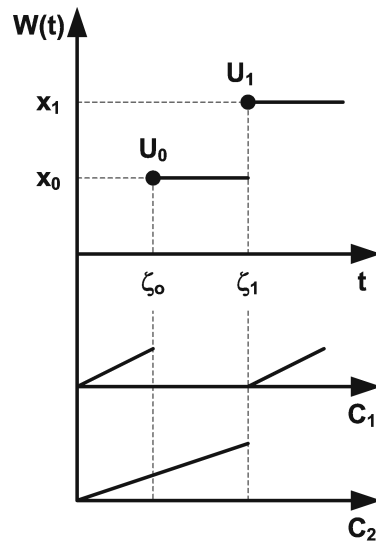


Figura 2.18 - Realização de um processo Generalizado de Semi-Markov.

Neste processo uma transição de estado ocorre quando um, ou mais eventos associados ao estado actual (ocupado) ocorrem. Esses eventos competem entre si para definir o instante de próxima transição de estado. Cada vez que ocorre uma transição de estado, um novo conjunto de eventos é escalonado. Para cada um desses eventos é definido um relógio que indica o tempo até à sua ocorrência, que por sua vez é definido de acordo com um mecanismo independente e estocástico (a partir da respectiva função distribuição). Se o evento escalonado não pertencer ao conjunto de eventos que definem uma transição desse estado mas que está associado ao próximo estado, o seu relógio continua a correr, caso contrário é cancelado e o seu conteúdo é ignorado. Este funcionamento é análogo ao dos Sistemas de Eventos Discretos (*Discrete Events Systems*) [PG89] [Cass93].

2.5.9 Análise de Processos Não-Markovianos

A evolução dos processos não-Markovianos pode analisar-se a partir de um conjunto de técnicas que procuram explorar a existência de instantes nesse processo onde a propriedade de Markov é respeitada. As técnicas existentes são:

- ♦ Expansão por Fases;
- ♦ Variáveis Suplementares;
- ♦ Teoria da Renovação.

2.5.9.1 Expansão por Fases

De acordo com esta técnica, um estado $i \in S$ para o qual a função distribuição do tempo de ocupação não seja do tipo exponencial, é substituída por um conjunto de estados (expansão por fases, de acordo com uma distribuição do tipo *phase-type*) com as seguintes propriedades [MR93] [BT97] (Fig. 2.19):

- ♦ O tempo de ocupação de cada um dos estados (fase) é exponencialmente distribuído (λ_i);
- ♦ A função distribuição correspondente ao tempo total de permanência no conjunto desses estados é uma aproximação da função distribuição original do tempo de ocupação do estado (distribuição *phase-type* correspondendo ao tempo de absorção de uma cadeia de Markov com estados absorventes).

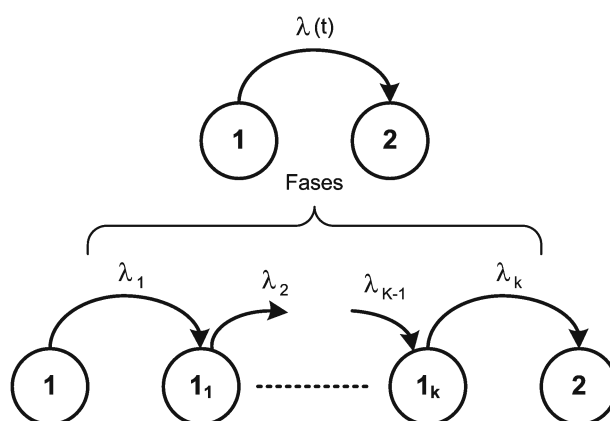


Figura 2.19 - Expansão por fases.

Trata-se assim de um método aproximado. Em princípio qualquer distribuição que possa ser expressa sob a forma um quociente em termos de transformada de Laplace é susceptível de ser aproximada por este método.

Os principais factores que influenciam a escolha da aproximação são [MR93]:

- ♦ Qualidade da aproximação obtida;
- ♦ Número de estados da aproximação;
- ♦ Facilidade de obtenção do modelo Markoviano (cadeia de Markov Discreta ou Contínua);
- ♦ Facilidade de ajuste dos parâmetros da aproximação.

A sua principal vantagem reside no facto de que após ser obtida a aproximação desejada, a solução poder ser obtida pelos métodos clássicos de resolução de cadeias de Markov. A sua principal desvantagem está relacionada o eventual

aumento incontrollável do espaço de estados (*state space explosion*) devido à introdução de novos estados.

A expansão pode ser realizada ligando em série ou paralelo (ou combinação de ambos) os vários estados intermédios. Uma ligação em série dá origem a uma distribuição hipo-exponencial ou de Erlang [Triv02], dependendo das fases possuírem ou não parâmetros idênticos. No caso de uma ligação em paralelo, obtêm-se uma distribuição hiper-exponencial. A selecção das distribuições que vão ser utilizadas para aproximar a distribuição não-exponencial é tipicamente realizada através de um parâmetro denominado *coeficiente de variação* C_x , sendo definido como

$$C_x = \frac{\sigma_x}{E[X]} \quad (2.49)$$

em que σ_x é o desvio padrão da variável aleatória X e $E[X]$ o seu valor esperado. A Tabela 2.3 apresenta a variação deste coeficiente em função do tipo de distribuição.

Importantes generalizações deste método são as distribuições de Cox, *Phase-Type* e Hiper-exponencial Generalizada. Uma alternativa para a representação das distribuições de Cox com vantagens computacionais, são as distribuições polinomial ou expo-polinomial. Saliente-se por fim, também a existência de ferramentas [BHT02] que permitem obter de forma automática os parâmetros associados a estas distribuições.

C_x	Distribuição
> 1	Hiper-exponencial
1	Exponencial
< 1	Hipo-exponencial
0	Determinística

Tabela 2.3 - Coeficiente de variação.

2.5.9.2 Variáveis Suplementares

Esta técnica baseia-se na re-definição do espaço de estados do processo através da introdução no mesmo de um conjunto de variáveis suplementares. Cada variável tem como objectivo representar o tempo de ocupação de um estado $i \in \mathcal{S}$ do sistema funcionando assim como uma espécie de relógio local [BT97] [Ger00].

Esta redefinição do espaço de estados num tuplo com uma componente discreta (o estado $i \in \mathcal{S}$) e outra contínua (o tempo T), desde que realizada apropriadamente,

resulta num processo Generalizado de Semi-Markov, o qual respeita a propriedade de Markov na nova definição do espaço de estados, resultando numa cadeia de Markov de Espaço de Estados Generalizado.

A partir de uma abordagem semelhante à formulada para as cadeias de Markov Contínuas, através da utilização de argumentos probabilísticos simples, nomeadamente no equilíbrio do fluxo de probabilidades entre estados, dos quais resultam as equações de Kolmogorov, é possível obter as equações que descrevem o comportamento do sistema. A solução consiste num sistema de equações diferenciais ordinárias ou parciais com condições fronteira [Lin98] [GT99] [Ger99] [Ger00].

A principal vantagem está no facto de qualquer distribuição do tempo de ocupação de um estado poder ser analiticamente tratada. A principal desvantagem está na complexidade da solução. Quando existem estados com tempos de ocupação não-exponenciais (ex. n variáveis suplementares), a solução é um sistema de equações diferenciais ordinárias ou parciais n -dimensional, o que resulta numa elevada complexidade.

Embora soluções simbólicas sejam possíveis para sistemas muito simples ou com uma estrutura particular, como por exemplo para funções distribuição em degrau (tempo de ocupação determinístico), a situação normal consiste na utilização de métodos numéricos, em tudo análogos aos definidos para as cadeias de Markov, para obter a solução. Embora existam soluções analíticas para sistemas n -dimensionais [Ger99], em termos de aplicações práticas está-se normalmente limitado a uma ou duas dimensões [Lin98] [Ger00].

2.5.9.3 Teoria da Renovação

Esta técnica baseia-se na aplicação da Teoria da Renovação (*Renewal Theory*) [Kul95] [Kao97] [Triv02], e mais concretamente nas propriedades dos processos de Renovação de Markov e nos processos Regenerativos. Neste caso, o estado do processo é apenas observado em certos instantes onde a propriedade de Markov se aplica. Entre estes instantes de regeneração está subordinado um outro processo estocástico, como por exemplo Semi-Markov ou Markov Regenerativo, a partir do qual é possível obter a solução do comportamento do sistema [LTP95] [FTP+98] [Ger00].

O comportamento do sistema é descrito a partir da estrutura de um processo Regenerativo (§2.5.7) através de duas matrizes, denominadas *núcleo local* $E(t)=[E_{ij}]$ e *núcleo global* $K(t)=[K_{ij}]$ definidas como

$$\begin{aligned} K_{ij}(t) &= \text{Prob}\{X_1=j, T_1 \leq t \mid X_0=i\} \\ E_{ij}(t) &= \text{Prob}\{Z(t)=j, T_1 > t \mid Z_0=i\} \end{aligned} \quad (2.50)$$

A matriz $\mathbf{K}$ representa a distribuição condicional conjunta do tempo até à próxima regeneração (renovação) T_1 e do estado depois da próxima regeneração X_1 , dado o estado presente no instante de regeneração X_0 , ou seja, o comportamento do processo nos instantes de regeneração. A matriz $\mathbf{E}$ descreve o comportamento do processo entre dois instantes consecutivos de regeneração. Como se trata de um processo regenerativo, basta considerar o comportamento até ao primeiro instante de regeneração T_1 , dado que desse ponto em diante o comportamento do sistema, em termos estocásticos, repete-se.

Definindo a matriz da probabilidade condicionada dos estados $\mathbf{V}(t)=[v_{ij}]$ como

$$v_{ij}(t) = \text{Prob}\{Y(t)=j \mid Y(0)=i\} \quad (2.51)$$

O comportamento do processo é dado por

$$\mathbf{V}(t) = \mathbf{E}(t) + \int_0^t \partial \mathbf{K}(u) \mathbf{V}(t-u) = \mathbf{E}(t) + \mathbf{K}(t) * \mathbf{V}(t) \quad (2.52)$$

com o integral da convolução de Stieltjes definido como

$$\int_0^t \partial \mathbf{K}(u) \mathbf{V}(t-u) = \int_0^t k(u) \mathbf{V}(t-u) du \quad (2.53)$$

onde $\mathbf{K}(t)$ possui uma função densidade de probabilidade $k(t) = \frac{\partial}{\partial t} \mathbf{K}(t)$.

A solução consiste assim num sistema de equações integrais, denominado sistema de *Equações de Volterra do Segundo Tipo* [GLT95]. Dado o vector da distribuição inicial $\boldsymbol{\pi}(0)$, o vector de probabilidades dos estados no instante t , $\boldsymbol{\pi}(t)$ é definido como

$$\boldsymbol{\pi}(t) = \boldsymbol{\pi}(0) \cdot \mathbf{V}(t) \quad (2.54)$$

A solução pode ser obtida quer no domínio dos tempos por métodos numéricos, quer no domínio de Laplace, embora neste último caso a transformação inversa para o domínio dos tempos seja apenas exequível para processos pouco complexos. É possível demonstrar a equivalência entre este método com o das variáveis suplementares [GT99].

A principal vantagem está no facto de qualquer distribuição do tempo de ocupação de um estado poder ser analiticamente tratada. A principal desvantagem está na dificuldade do estabelecimento dos pontos de regeneração, que nem sempre podem existir, e na complexidade da solução.

2.6 Modelos de Recompensa

Os modelos de recompensa (*Reward Models*) são extensões aos processos estocásticos obtidos através da definição de funções de transformação que mapeiam elementos da estrutura do processo no conjunto $\mathbb{R}$. O valor obtido dessas funções é denominado recompensa (*reward*) [STR88] [CMS+90] [SM91b] [MFT00] [PT00].

Estas funções são tipicamente definidas por uma estrutura composta por um vector $\mathbf{rr}=[rr_i]$ e por uma matriz $\mathbf{IR}=[ir_{ij}]$ da seguinte forma (Fig. 2.20):

- ♦ **Taxa de recompensa** (*Rate reward*). Durante a permanência no estado i do processo a recompensa é acumulada a uma taxa rr_i ;
- ♦ **Impulso de recompensa** (*Impulse reward*). Quando ocorre uma transição do estado i para o estado j , é adicionado instantaneamente à recompensa um valor ir_{ij} .

As medidas associadas às taxas de recompensa são calculadas a partir do tempo de ocupação dos estados, enquanto as associadas com os impulsos de recompensa necessitam das frequências de transição entre estados.

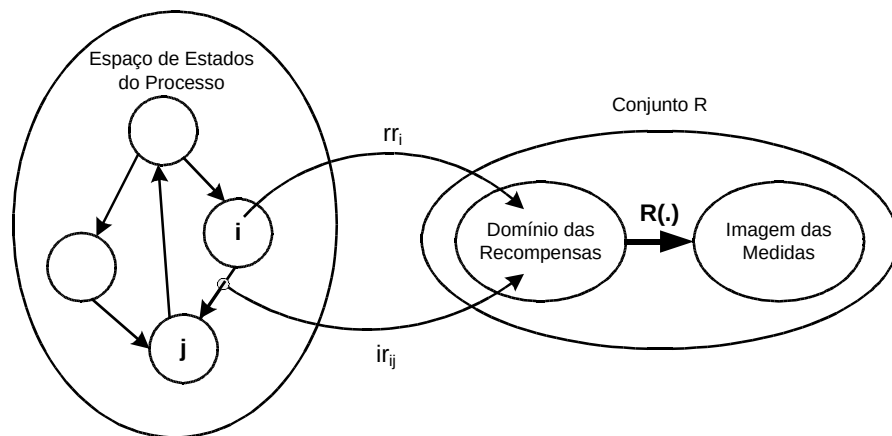


Figura 2.20 - Conceito de função de recompensa.

Definindo as seguintes variáveis indicadoras:

$$I_i(t) = \begin{cases} 1 & X(t) = i, \quad i \in S \\ 0 & \text{caso contrário} \end{cases} \quad (2.55)$$

$$I_{ij}(t) = \begin{cases} 1 & \text{se ocorrer uma transição de } i \rightarrow j \text{ no instante } t, \\ 0 & \text{caso contrário} \end{cases} \quad i, j \in S \quad (2.56)$$

é possível definir as seguintes variáveis de recompensa:

- ♦ Valor instantâneo

$$R_I(t) = \sum_{i \in S} rr_i \cdot I_i(t) + \sum_{i, j \in S} ir_{ij} \cdot I_{ij}(t) \quad (2.57)$$

- ♦ Valor acumulado

$$R_A(t) = \int_0^t R_I(x) dx \quad (2.58)$$

- ♦ Valor médio temporal (*time averaged*)

$$R_M(t) = \frac{1}{t} R_A(t) \quad (2.59)$$

Como se tratam de processos estocásticos, as variáveis de recompensa são também variáveis aleatórias sendo mais usual definir outras medidas associadas, tais como:

- ♦ Valor esperado: $E[R_x(t)]$, $x \in \{I, A, M\}$;
- ♦ Função distribuição de probabilidade: $Prob[R_x(t) \leq u]$, $x \in \{I, A, M\}$

As medidas de valor esperado são as mais simples de obter e são suficientes para a maioria dos problemas. As funções distribuição são mais complexas de obter, existindo técnicas analíticas para processos de Markov e de Semi-Markov [STR88] [CMS+90].

A grande vantagem da utilização das variáveis de recompensa reside no facto de constituírem uma estrutura ortogonal ao próprio processo, a partir das quais é possível definir muitas outras medidas, tais como as relacionadas com a confiança no funcionamento.

2.6.1 Relação com as Medidas da Confiança no Funcionamento

Dada a importância que os modelos de recompensa assumem na definição das medidas da confiança no funcionamento nesta secção pretende-se apresentar alguns exemplos das relações existentes.

Assim, é possível demonstrar [Ger00] que

$$E[R_I(t)] = \sum_{i \in S} rr_i \pi_i(t) + \sum_{i,j \in S} ir_{ij} \phi_{ij}(t) \quad (2.60)$$

em que $\pi_i(t)$ representa a probabilidade do estado i no instante t , e $\phi_{ij}(t)$ o valor esperado instantâneo da frequência de transição entre os estados i e j . O valor estacionário $t \rightarrow \infty$ é dado por

$$E[R_I(\infty)] = \sum_{i \in S} rr_i \pi_i + \sum_{i,j \in S} ir_{ij} \Phi_{ij} \quad (2.61)$$

em que Φ_{ij} é o valor esperado estacionário da frequência de transição entre os estados i e j .

O valor esperado acumulado, é obtido de forma análoga

$$E[R_A(t)] = \sum_{i \in S} rr_i \int_0^t \pi_i(x) dx + \sum_{i,j \in S} ir_{ij} \int_0^t \phi_{ij}(x) dx \quad (2.62)$$

Em termos estacionários, a medida anterior só tem sentido se for finita.

Recuperando a terminologia apresentada na secção §2.4, a disponibilidade pode ser obtida atribuindo uma taxa $rr_i=1$ aos estados operacionais. A disponibilidade instantânea $A(t)$ pode ser obtida de $E[R_I(t)]$, e a disponibilidade A_∞ através de $E[R_I(\infty)]$.

A fiabilidade $R(t)$ pode obter-se de forma análoga. No entanto, só tem sentido obter $E[R_I(t)]$ dado que os estados de avaria são absorventes. Um raciocínio semelhante pode ser realizado para a segurança (*safety*) e para a capacidade de manutenção. Utilizando a mesma definição de recompensa, o tempo médio até falhar MTTF, pode ser obtido através de $E[R_A(\infty)]$.

Para terminar, não é possível deixar de referir a importância deste formalismo na obtenção de medidas do desempenho na presença de falhas (*performability*) [Mey80] [SM91b] [PT00].

2.7 Redes de Petri

As redes de Petri são uma ferramenta de modelação gráfica e matemática, com a capacidade de descrever e analisar a dinâmica de sistemas caracterizados pela existência de comportamentos concorrentes, síncronos, assíncronos, paralelos, sequenciais, com conflitos, de partilha de recursos, determinísticos e estocásticos [Mur89]. Como ferramenta gráfica, disponibiliza um poderoso meio de troca de informações entre utilizadores, onde requisitos e especificações complexas podem ser eficientemente representadas de forma simples e não ambígua. Como ferramenta matemática, permite obter um modelo do comportamento do sistema, a partir do qual é possível realizar uma análise formal e extrair um conjunto relevante de propriedades que permitem a sua caracterização.

Em termos de representação gráfica, uma rede de Petri é constituída por vários elementos gráficos cuja interpretação pode ser realizada segundo diferentes perspectivas [Mur89]. No sentido de facilitar a apresentação das redes de Petri, e tendo em conta o contexto desta dissertação, optou-se por uma interpretação fortemente ligada ao conceito de sistema de eventos discretos (DES – *Discrete Event Systems*) [Cass93].

Desta forma, o estado do sistema é representado por posições (círculos) que podem conter diferentes números testemunhos (*tokens*, pequenos círculos preenchidos) (Fig. 2.21). Cada posição pode ser considerada como um estado local do sistema. Actividades ou eventos do sistema são representados por meio de transições (barras) que estão interligadas às posições por arcos (ligações orientadas). De acordo com certas regras bem definidas as transições disparam e movimentam os testemunhos entre as posições, reflectindo assim a mudança do estado do sistema. A rede de Petri na sua globalidade pode ser encarada como o modelo do comportamento do sistema.

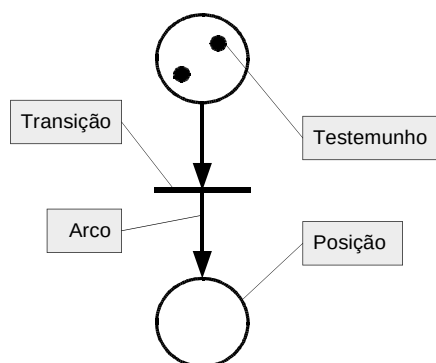


Figura 2.21 - Exemplo de uma rede de Petri.

Nesta secção pretende-se apresentar as redes de Petri, e em particular as redes de Petri Estocásticas. Sem ser exaustivo, o objectivo é introduzir conceitos com um nível suficiente de detalhe que permitam abordar de forma simples os modelos de confiança no funcionamento apresentados em capítulos posteriores. [Mur89] [DA92] [Cass93] [MBC+95] [Wang98] [Lin98] [Ger00] [Haa02] são referências para um tratamento mais detalhado do assunto.

2.7.1 Redes de Petri Ordinárias

Formalmente uma rede de Petri Ordinária (RP) é definida como [Ger00]:

$$RP=(P,T,I,O,M_0) \quad (2.63)$$

em que:

- ♦ $P=\{p_1, p_2, \dots, p_n\}$ é o conjunto das posições, com cardinalidade n ;
- ♦ $T=\{t_1, t_2, \dots, t_m\}$ é o conjunto das transições, com cardinalidade m , em que: $P \cup T \neq \emptyset$ e $P \cap T = \emptyset$;
- ♦ $I \subseteq P \times T$ é o conjunto dos arcos de entrada, definidos das posições para as transições;
- ♦ $O \subseteq T \times P$ é o conjunto dos arcos de saída, definidos das transições para as posições;
- ♦ $M_0 \in \mathbb{N}_0^n$ é o vector da marcação inicial. Cada entrada m_i do vector representa o número de testemunhos existentes inicialmente na posição p_i .

Por forma de simplificar a descrição do funcionamento de uma rede de Petri, é comum definir adicionalmente a seguinte notação:

- ♦ $\#p$ representa o número de testemunhos numa posição p ;
- ♦ A marcação da rede é definida pelo vector $M=\{\#p_1, \#p_2, \dots, \#p_n\} \in \mathbb{N}_0^n$. A marcação representa o estado global da rede de Petri. Cada entrada m_i do vector representa o número de testemunhos existentes na posição p_i na marcação M , usualmente definida como $\#(p, M)$;
- ♦ A função $I(p, t)$ é definida como igual a 1 se $(p, t) \in I$, e igual a 0 caso contrário. $O(t, p)$ é definida de forma análoga;
- ♦ As posições ligadas por um arco de entrada a uma transição são definidas como posições de entrada. As posições de saída, são definidas de forma análoga.

A dinâmica de uma rede de Petri é descrita por um conjunto de regras que definem a sua evolução:

- ♦ Um transição t é dita *habilitada* (*enabled*) numa marcação M , se pelo menos um (1) testemunho está presente em todas as suas posições de entrada

$$I(p, t) \leq \#(p, M), \quad p \in P \quad (2.64)$$

caso contrário é dita *desabilitada* (*disabled*);

- ♦ Uma transição habilitada t pode disparar, removendo um testemunho de cada posição de entrada e colocando um testemunho em cada posição de saída. Esta operação é considerada atômica. O disparo de uma transição t numa marcação M origina uma nova marcação M' definida como:

$$\#(p, M') = \#(p, M) - I(p, t) + O(p, t), \quad p \in P \quad (2.65)$$

- ♦ Se existirem várias transições habilitadas simultaneamente, estas não podem disparar simultaneamente. Isto implica a adopção de uma semântica onde as transições são disparadas de forma intercalada (*interleaving semantics*), sem nenhuma ordem particular pré-estabelecida [MBC+95];
- ♦ O disparo de uma transição pode desabilitar outras transições;
- ♦ Uma marcação é dita acessível, se pode ser acedida a partir da marcação inicial através do disparo de um conjunto de transições. Cada marcação corresponde a um estado do sistema;
- ♦ O conjunto das marcações acessíveis S (*reachability set*) é constituído por todas as marcações que podem ser acessíveis, sendo usualmente identificado como o espaço de estados da rede de Petri, e consequentemente também do sistema;
- ♦ O grafo das marcações acessíveis (S, A) (*reachability graph*) consiste num grafo constituído por vértices S e por ligações $A \subseteq S \times T \times S$ entre estes. Uma ligação $a = (M, t, M')$ pertence a A se a transição t estiver habilitada na marcação M e o seu disparo levar a uma nova marcação M' .

Para além da descrição formal, é comum caracterizar uma rede de Petri através de um conjunto de propriedades adicionais definidas quer ao nível da estrutura da rede, quer ao nível do comportamento dinâmico da mesma. Exemplos destas propriedades são: limitada (*boundedness*), viva (*liveness*), segura (*safeness*), reversível, P-invariantes, T-invariantes, etc. [DA92] [MBC+95]. A partir destas propriedades podem ser extraídas informações importantes sobre o comportamento do sistema, bem como sobre a validação do próprio modelo (rede de Petri) desenvolvido.

2.7.2 Redes de Petri Extendidas

No sentido de aumentar quer a capacidade de modelação, definida como a classe de problemas que podem ser modelizados através de uma rede Petri, quer a conveniência da modelação, definida como o grau de facilidade e a capacidade de representar o comportamento de forma simples, compacta e de natural, foram definidas várias extensões estruturais às redes de Petri [Cia87], das quais se destacam as seguintes:

- ♦ **Multiplicidades dos arcos.** Associando a cada arco uma multiplicidade $\in \mathbb{N}^+$ é possível remover ou adicionar mais do que um testemunho simultaneamente às posições quando uma transição dispara;
- ♦ **Arcos inibidores.** Um arco inibidor liga uma posição a uma transição e permite desabilitar uma transição quando a respectiva posição possui um ou mais testemunhos. Por analogia, um arco inibidor pode ter uma multiplicidade superior a um. Graficamente são representados por um arco cuja extremidade orientada é substituída por um pequeno círculo;
- ♦ **Prioridades.** A introdução de prioridades nas transições permite definir a ordem pela qual as transições são disparadas;
- ♦ **Guardas.** Para além das condições já definidas para a habilitação de uma transição, é possível acrescentar uma condição adicional através de um predicado definido sobre a marcação M ;
- ♦ **Multiplicidades dos arcos dependentes da marcação.** A multiplicidade dos arcos pode depender da marcação M .

Para além destas, outras extensões foram também definidas para as redes de Petri, tais como: coloridas, contínuas, híbridas, sincronizadas, interpretadas, orientadas ao objecto e de lógica difusa [DA92] [Wang98]. Contudo, dado não serem relevantes para o trabalho desenvolvido, a sua descrição é omitida.

A partir das extensões estruturais apresentadas define-se uma rede de Petri Extendida (RPX) como [Ger00]:

$$RPX = (P, T, I, O, H, \Pi, G, M_0) \quad (2.66)$$

onde:

- ♦ O conjunto das posições P , o conjunto das transições T e a marcação inicial M_0 , são definidas como no caso de uma rede de Petri Ordinária;
- ♦ $I \in (\mathbb{N}_0^n \rightarrow \mathbb{N}_0)^{n \times m}$ é a matriz da multiplicidade dos arcos de entrada dependente da marcação. A entrada (i, j) da matriz indica a multiplicidade dependente

da marcação do arco de entrada da posição p_i para a transição t_j . Cada entrada é um conjunto de funções definidas de $\mathbb{N}_0^n$ para $\mathbb{N}_0$. Se o arco possui uma multiplicidade que não é dependente da marcação, a entrada é definida por uma função constante. Um arco com multiplicidade igual a zero, corresponde a um arco que não existe;

- ♦ $O \in (\mathbb{N}_0^n \rightarrow \mathbb{N}_0)^{m \times n}$ é a matriz da multiplicidade dos arcos de saída dependente da marcação. As definições são análogas ao caso da matriz dos arcos de entrada;
- ♦ $H \in (\mathbb{N}_0^n \rightarrow \mathbb{N}_0)^{n \times m}$ é a matriz da multiplicidade dos arcos inibidores dependente da marcação. As definições são análogas ao caso ao caso da matriz dos arcos de entrada;
- ♦ $\Pi \in \mathbb{N}_0^m$ é um vector que associa a cada transição uma prioridade;
- ♦ $G \in (\mathbb{N}_0^n \rightarrow \{\text{verdadeiro}, \text{falso}\})^m$ é o vector que associa a cada transição uma guarda. $\mathbb{N}_0^n \rightarrow \{\text{verdadeiro}, \text{falso}\}$ é o conjunto de predicados definidos sobre as marcações.

A introdução deste tipo de extensões estruturais obrigou também a uma redefinição da dinâmica da rede, passando esta a ser definida como [Ger00]:

- ♦ Uma transição t possui *concessão* (*concession*) numa marcação $\mathbf{M}$ se e só se:
 1. O número de testemunhos em cada posição de entrada é maior ou igual à multiplicidade do arco de entrada

$$\#(p, \mathbf{M}) \geq I(p, t, \mathbf{M}), \quad p \in P^{13} \quad (2.67)$$

2. Se existir um arco inibidor, o número de testemunhos na posição inibidora (posição de onde parte o arco) é menor do que a multiplicidade do arco

$$\#(p, \mathbf{M}) < H(p, t, \mathbf{M}) \vee H(p, t, \mathbf{M}) = 0, \quad p \in P \quad (2.68)$$

3. O predicado da guarda respectiva é verdadeiro

$$G(t, \mathbf{M}) = \text{verdadeiro} \quad (2.69)$$

- ♦ A transição t está habilitada na marcação $\mathbf{M}$ se e só se :

1. t tiver concessão em $\mathbf{M}$;

¹³ A existência de definições dependentes das marcação $\mathbf{M}$, implicou também a extensão da notação apresentada anteriormente para passar a incluir a mesma sob a forma $(\dots, \mathbf{M})$.

2. Todas as transições t' com concessão em M tem prioridade menor ou igual à prioridade de t

$$\Pi(t') \leq \Pi(t) \quad (2.70)$$

- ♦ Uma transição habilitada t pode disparar e remover testemunhos de cada uma das suas posições de entrada e adicionar testemunhos a cada uma das suas posições de saída de acordo com a multiplicidade dos arcos de entrada e saída respectivamente

$$\#(p, M') = \#(p, M) - I(p, t, M) + O(p, t, M), \quad p \in P \quad (2.71)$$

- ♦ Tal como numa rede de Petri Ordinária se existirem várias transições habilitadas simultaneamente, estas não podem disparar simultaneamente;
- ♦ As definições de marcação acessível, de conjunto de marcações acessíveis e grafo das marcações acessíveis mantêm-se inalteradas.

2.7.3 Redes de Petri Temporizadas e Estocásticas

As extensões apresentadas anteriormente permitem apenas aumentar a conveniência da modelação, sem contudo aumentar o poder da mesma. Dito de outra forma, as redes de Petri obtidas permitem apenas realizar uma avaliação qualitativa do sistema, sem contudo permitirem capturar as suas dependências temporais.

Uma das possibilidades para simultaneamente resolver esta limitação e aumentar o poder de modelação de uma rede de Petri, consiste na introdução do factor tempo. Das várias propostas para introduzir o tempo na semântica das redes de Petri, a mais comum, e também adoptada nesta dissertação, consiste em associar o tempo às transições [Wang98]. A vantagem desta abordagem está no facto desta não alterar a semântica da rede de Petri não-temporizada subjacente.

Associar o tempo às transições é equivalente a identificar a habilitação de uma transição com o início de uma actividade do sistema, enquanto o seu disparo corresponde ao fim dessa mesma actividade. Quando o disparo de uma transição desabilita outra transição, anteriormente habilitada, isto corresponde à interrupção da respectiva actividade. Associada à transição existem assim uma função distribuição que define o tempo de disparo (inclui-se assim tempos puramente determinísticos ou estocásticos ou até a combinação de ambos).

No contexto da modelação de sistemas cujo comportamento é estocástico, o tempo de disparo é expresso através de uma variável aleatória, especificada a partir da sua função distribuição. Por forma também a permitir a modelação de dependências

lógicas, as transições podem também disparar sem consumir qualquer tempo (operação atómica). Estas transições são denominadas *imediatas*, por oposição às restantes denominadas *temporizadas* [CMB+93]. Em termos gráficos uma transição imediata é representada por uma barra fina, enquanto uma transição temporizada é representada por um rectângulo, eventualmente preenchido de forma diferente em função do tipo de distribuição associada à transição.

Em termos formais o tempo é associado a uma transição t_k sob a forma de um relógio individual [MBB+89] (Fig. 2.22).

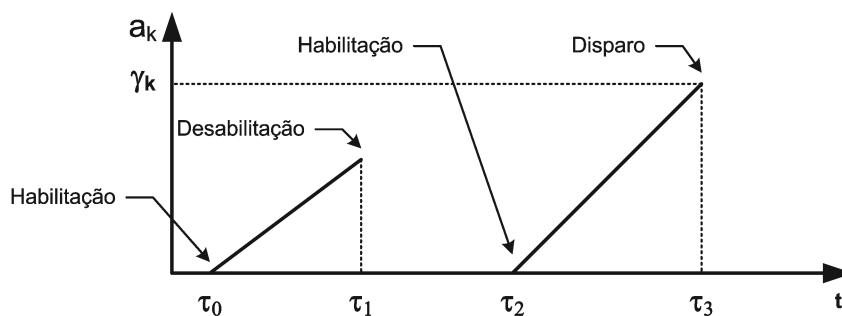


Figura 2.22 - Exemplo do disparo de uma transição temporizada.

O valor do relógio é representado por intermédio de uma variável de memória a_k . Quando a transição fica habilitada pela primeira vez (τ_0 e τ_2) a variável a_k é inicializada com zero. Enquanto a transição se mantiver habilitada, o relógio é incrementado a uma velocidade constante. A transição dispara (τ_3) quando o relógio atinge o tempo de disparo γ_k da transição. O tempo de disparo corresponde assim ao tempo requerido pela actividade associada à transição. Neste último caso a_k é de novo reinicializada com zero. Se a transição for desabilitada antes de disparar (τ_1), a variável de memória a_k pode ou não ser reinicializada, em função da política de execução definida. O tempo de disparo γ_k é amostrado da função distribuição associada à transição.

A introdução do factor tempo juntamente com a existência de conflitos entre transições simultaneamente habilitadas, colocam em destaque duas questões importantes:

1. Entre várias transições habilitadas numa marcação M , qual é aquela que vai disparar primeiro?
2. Quando uma transição é desabilitada pelo disparo de uma outra transição, o que é que acontece ao seu relógio? E de que forma o seu relógio memoriza quanto tempo esteve a transição habilitada?

A resposta a ambas as questões é definida através da respectiva **política de execução** (*execution policy*) [MBB+89]. Esta consiste num conjunto de

especificações que univocamente definem o processo estocástico da rede de Petri em função quer da sua estrutura, quer do conjunto de funções distribuição associadas a cada transição temporizada. A política de execução é normalmente decomposta em várias: disparo, memória e concorrência [BT98].

2.7.3.1 Política de Disparo

A política de disparo define, entre as transições que estão habilitadas, a forma como é escolhida a próxima transição que vai disparar [MBB+89]. As propostas existentes são:

- ♦ **Política de corrida** (*race policy*). Quando uma marcação é atingida, o tempo restante de disparo (RFT - *Remaining Firing Time*), é calculado para cada transição habilitada em função do seu tempo de disparo e da sua variável de memória ($RFT = \gamma_k - a_k$). A próxima transição a disparar é aquela que tem o menor tempo restante de disparo;
- ♦ **Política de pré-selecção** (*pre-selection policy*). A próxima transição a disparar é escolhida de acordo com uma função distribuição de probabilidade (discreta) definida externamente e independente dos tempos de disparo.

A política de corrida é a utilizada em transições temporizadas no caso deste ser contínuo. A política de pré-selecção é utilizada para transições imediatas, e também em transições temporizadas com tempo discreto ou com uma função distribuição discreta (ex. determinística).

2.7.3.2 Política de Memória

Diferentes mecanismos podem ser adoptadas para definir a forma como o passado do sistema influencia a sua futura evolução, nomeadamente através da memorização do tempo já gasto nas actividades.

Dois mecanismos básicos foram definidos para especificar a forma como é gerido o relógio de uma transição habilitada sempre que ocorre uma mudança de estado provocado pelo disparo de um outra transição (temporizada ou não) [MBC+95]:

- ♦ **Continuação** (*Continue*). O relógio da transição habilitada t_k mantém o seu valor presente a_k e num momento posterior irá continuar a ser incrementado;
- ♦ **Reinicialização** (*Restart*). O relógio da transição habilitada t_k é reinicializado, ou seja a_k é colocado em zero, e um novo valor para o tempo de disparo γ_k será gerado quando necessário. Este tempo é amostrado a partir da função distribuição associada à transição.

Tendo em conta que num sistema real podem existir comportamentos muito variados, torna-se necessário dispor de diferentes forma de memorizar o passado. Combinando de forma diferente os mecanismos de continuação e reinicialização é possível definir vários tipos de **políticas de memória** [MBB+89]:

- ♦ **Memória de re-amostragem** (*resampling memory*). A variável de memória a_k de uma transição habilitada t_k é reinicializada com zero sempre que ocorre o disparo de uma qualquer transição;
- ♦ **Memória de habilitação** (*enabling memory*). A variável de memória a_k de uma transição habilitada t_k contabiliza o tempo decorrido desde o último instante em que a transição foi habilitada. Quando a transição t_k for desabilitada (mesmo sem disparar) a variável de memória é reinicializada com zero;
- ♦ **Memória de idade** (*age memory*). A variável de memória a_k de uma transição habilitada t_k contabiliza o tempo cumulativo em que a transição esteve habilitada desde o seu último disparo. A variável de memória a_k só é reinicializada com zero se a transição t_k disparar.

Nas políticas de memória definidas anteriormente é assumido que de cada vez que o tempo de disparo da transição é reinicializado com um valor, este valor é amostrado da função distribuição associada à transição.

A combinação da política de memória com o instante em que é realizada a amostragem do tempo de disparo (**política de amostragem** (*sampling policy*)) resulta na **política de preempção** [FPT+98] [BKP+95]. Utilizando uma terminologia análoga à das filas de espera, a política de preempção de uma transição é definida como (Fig. 2.23):

- ♦ **Preempção repetida diferente** (PRD) (PRD - *Preemptive Repeat Different*). A transição t_k segue uma política de memória de habilitação e o tempo de disparo y_k é re-amostrado da função distribuição respectiva sempre que a variável a_k é inicializada com zero, o que acontece quando a transição dispara ou é desabilitada;
- ♦ **Preempção e continuação** (PC) (PRS - *Preemptive Resume*). A transição t_k segue uma política de memória de idade e o tempo de disparo y_k é re-amostrado da função distribuição respectiva sempre que a variável a_k é inicializada com zero, o que acontece quando a transição dispara;
- ♦ **Preempção repetida idêntica** (PRI) (PRI - *Preemptive Repeat Identical*). A transição t_k segue uma política de memória de habilitação e o tempo de disparo y_k só é re-amostrado da função distribuição respectiva quando a transição dispara.

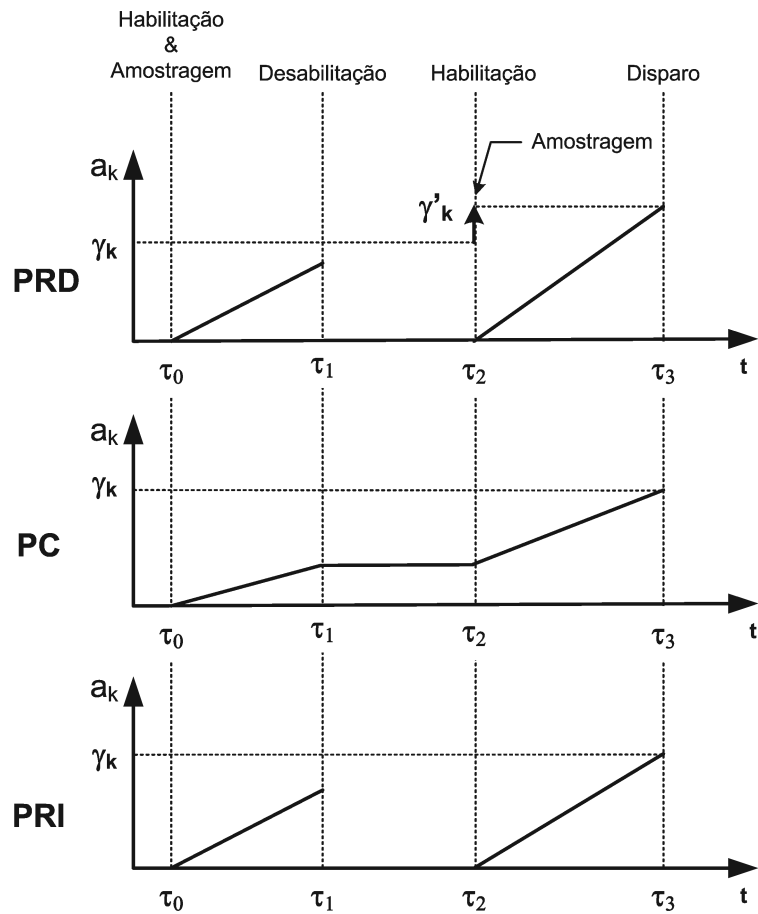


Figura 2.23 - Políticas de preempção.

2.7.3.3 Política de Concorrência

A política de concorrência define o comportamento, ao nível do disparo, de uma transição temporizada cujo grau de activação (*enabling degree*) é superior a um. O grau de activação $ED(t, \mathbf{M})$ indica o número de vezes que uma transição pode disparar numa marcação $\mathbf{M}$. As seguintes propostas foram definidas [MBC+95]:

- ♦ **Servidor único (SU) (Single Server).** O tempo de disparo γ_k é definido quando a transição t_k é habilitada pela primeira vez. Só após o disparo da transição, e se esta continuar habilitada, é que é amostrado um novo tempo de disparo. Isto implica que os testemunhos que estão a habilitar a transição sejam processados em série;
- ♦ **Servidor infinito (SI) (Infinite Server).** O conjunto de testemunhos que estão a habilitar a transição t_k são processados logo que surgem nas posições de entrada da mesma. O correspondente tempo de disparo γ_k é amostrado nesse instante, e os tempos de disparo dos vários conjuntos de testemunhos que estão a habilitar a transição decorrem em paralelo. Isto é equivalente à

existência de vários relógios que operam em paralelo associados à transição. Cada vez que um dos relógios atinge o tempo de disparo a transição dispara;

- ♦ **Servidor múltiplo (SM)** (*Multiple Server*). O conjunto de testemunhos que estão a habilitar a transição t_k são processados logo que surgem nas posições de entrada de forma análoga ao servidor infinito, mas em que existe um limite ao grau de paralelismo. Quando este grau é excedido, os tempos de disparo dos novos conjuntos de testemunhos que estão a habilitar a transição, são apenas definidos quando o grau voltar a ser inferior ao máximo.

2.7.3.4 Redes de Petri Estocásticas

Uma rede de Petri Estocástica (RPE) é definida como [Ger00]

$$RPE = (P, T, I, O, H, \Pi, G, \mathbf{M}_0, ATB) \quad (2.72)$$

onde:

- ♦ $P, T, I, O, H, \Pi, G, \mathbf{M}_0$ são definidos de forma análoga às redes de Petri Extendidas;
- ♦ As transições com prioridade igual a zero são temporizadas, e as transições com prioridade maior do que zero são imediatas. Isto implica que as transições imediatas possuam sempre prioridade em relação às temporizadas;
- ♦ ATB representa os atributos de todas as transições, definidos pelo vector:

$$ATB = (DT, W, MD, PP, CN)^m \quad (2.73)$$

onde:

- $DT \in \mathbb{N}_0^n \rightarrow F$ representa as funções distribuição do tempo de disparo eventualmente dependentes da marcação. F representa o conjunto das possíveis funções distribuição com suporte em $[0, \infty[$;
- $W \in \mathbb{N}_0^n \rightarrow \mathbb{R}^+$ representa os pesos associados às transições com política de pré-selecção eventualmente dependente da marcação;
- $MD \in \{CNT, ESC, DEH\}$ representa o tipo de dependência da função distribuição do tempo de disparo em relação à marcação:
 - CNT : constante e independente da marcação;
 - ESC : dependente através de factores de escala (*scaling factors*). O aspecto (escala) da função distribuição do tempo de disparo pode modificar-se entre a habilitação e o disparo da transição;

- *DEH*: dependente da habilitação (*enabling dependent*). A função distribuição é definida no instante em que a transição é habilitada, e mantém-se inalterada enquanto a transição não disparar.
- $PP \in \{PRD, PC, PRI\}$ representa a política de preempção;
- $CN \in \{SU, SI, SM\}$ representa a política de concorrência.

A presença de transições imediatas implica uma distinção entre marcações. Assim, as marcações onde estejam habilitadas transições imediatas são denominadas marcações instáveis (*vanishing markings*), porque o tempo que o processo passa nestas marcações é nulo. As marcações onde apenas transições temporizadas estão habilitadas são denominadas marcações estáveis (*tangible markings*). Desta forma o conjunto das marcações acessíveis está também subdividido em marcações estáveis e instáveis, embora estas últimas, devido ao seu comportamento, não contribuam para o espaço de estados do sistema.

A dinâmica de uma rede de Petri Estocástica passa a ser definida como [Ger00]:

- ♦ Se a marcação corrente $\mathbf{M}$ é instável, é realizada uma escolha probabilística (política de disparo de pré-selecção) entre as várias transições habilitadas.

Uma transição t é escolhida com probabilidade

$$\frac{W(t, \mathbf{M})}{\sum_{t' \text{ habilitada em } \mathbf{M}} W(t', \mathbf{M})}^{14} \quad (2.74)$$

disparando sem consumir qualquer tempo. As transições habilitadas referidas nesta expressão dizem apenas respeito ao *conjunto de conflitos estendido*¹⁵ (ECS - *Extended Conflict Set*) associado à transição.

- ♦ Se a marcação corrente $\mathbf{M}$ é estável:
 1. No instante em que a marcação $\mathbf{M}$ é atingida, o tempo de disparo para cada transição que fica habilitada é definido a partir da sua política de preempção como:
 - a) **PRD**. É amostrado um novo valor de $F(t, \mathbf{M}, x)$ ¹⁶;

¹⁴ $W(t, \mathbf{M})$ representa o peso associado à transição t na marcação $\mathbf{M}$.

¹⁵ Uma definição formal deste conceito requer uma apresentação detalhada das propriedades das redes de Petri com prioridades que estão para lá dos objectivos desta dissertação. Uma tratamento exaustivo pode ser encontrado em [MBB+89], [CMB+93] e [MBC+95].

¹⁶ $F(t, \mathbf{M}, x)$ representa a função distribuição do tempo de disparo da transição t na marcação $\mathbf{M}$ em relação a x (tempo).

- b) **PC**. Se a variável de memória da transição for zero é amostrado um novo valor de $F(t, \mathbf{M}, x)$ caso contrário o relógio é re-inicializado com o valor corrente da variável de memória;
 - c) **PRI**. É utilizado o último valor amostrado caso a transição não tenha disparado, ou caso contrário é amostrado um novo valor de $F(t, \mathbf{M}, x)$.
2. As transições que já estavam habilitadas na marcação anterior não sofrem alterações no seu relógio (variável de memória a_k), excepto se forem desabilitadas nesta nova marcação. Neste último caso aplicam-se as regras definidas em (§2.7.3.2);
 3. Depois deste instante, os relógios de todas as transições habilitadas são incrementados. Se existirem transições cujos tempos de disparo sejam dependentes através de factores de escala, o tempo restante de disparo é actualizado de acordo;
 4. Quando pelo menos um dos tempos restantes de disparo (RTF) das transições se torna zero, a respectiva transição dispara imediatamente. Se dois ou mais destes tempos se tornam zero no mesmo instante, é realizada uma escolha probabilística para definir, entre essas, qual das transições vai disparar (politica de disparo de pré-selecção);
 5. Logo de seguida transições imediatas podem disparar. O objectivo é que os relógios da transições temporizadas não sejam afectados caso sejam atingidas marcações instáveis temporárias em que as transições temporizadas possam eventualmente perder a sua concessão (*internal preemption*);
 6. Se uma transição t_k perde a sua concessão depois de um disparo e se a sua politica de preempção é PRD ou PRI, a sua variável de memória a_k é re-inicializada com o valor zero. No caso da sua politica de preempção ser PC a sua variável de memória a_k mantém o valor corrente;
 7. Os passos 3 a 5 são repetidos enquanto existirem transições cujo tempo restante de disparo seja igual a zero. Caso contrário o próximo passo é o 1.

O processo estocástico que resulta da evolução temporal da rede de Petri está intimamente relacionado com a evolução da marcação estável da rede. Isto resulta do facto da marcação da rede representar simultaneamente o estado da rede e do sistema modelado. Formalmente a evolução da rede de Petri pode ser caracterizada através da combinação de um percurso estabelecido ao longo do seu grafo de marcações acessíveis. Este percurso é definido como uma sequência de instantes de tempo associados ao disparo consecutivo das várias transições ao longo do percurso, sendo denominado como *sequência de execução temporizada* T_E [BT98]

$$T_E = \{(\tau_0, \mathbf{M}_0); (\tau_1, \mathbf{M}_1); \dots (\tau_i, \mathbf{M}_i); \dots\} \quad (2.75)$$

em que τ_i representa o instante de disparo de uma transição e M_i a marcação estável atingida após este disparo. O intervalo de tempo $(\tau_{i+1} - \tau_i)$ representa o período de tempo em que a rede de Petri permanece na marcação M_i . τ_0 é zero e M_0 é a marcação inicial. Partindo desta sequência, é possível realizar o seu mapeamento num processo estocástico de tempo contínuo denominado *processo de marcação* $\{M(t) | t \geq 0\}$. Este representa a evolução da marcação estável M em função do tempo t , resultando naturalmente no processo estocástico subjacente à rede de Petri [CGL94].

A análise da evolução do processo de marcação faz-se através da construção do *grafo reduzido das marcações acessíveis* (S, A) (*Reduced Reachability Graph*). O conjunto de vértices S deste grafo representam o conjunto de marcações estáveis e o conjunto das ligações entre vértices é definido por $A \subseteq S \times T^* \times S$. $(M, t_i t^*, M')$ pertence a A se existir uma sequência $t_i t^*$ consistindo numa transição temporizada t_i e numa sequência, possivelmente vazia, t^* de transições imediatas que levam de uma marcação estável M a outra marcação estável M' . Na prática este grafo representa o grafo das marcações acessíveis, em que as marcações instáveis foram removidas. Estas marcações são removidas decompondo a rede em sub-redes de transições imediatas para as quais as probabilidades de saída podem ser calculadas baseadas na estrutura da rede, e posteriormente incorporadas de forma apropriada no modelo estocástico do processo de marcação. Os estados deste grafo constituem os estados do processo estocástico subjacente, ou seja, é possível definir um isomorfismo entre ambos.

A resolução do processo estocástico subjacente à rede de Petri implica previamente a necessidade da sua construção a partir das características estruturais e comportamentais da própria rede. Ou seja, é necessário gerar previamente o grafo reduzido das marcações acessíveis, a partir do qual o processo estocástico é construído, para depois posteriormente obter a respectiva solução. Para gerar este grafo é comum recorrer a algoritmos baseados na exploração da estrutura do mesmo ou em métodos matriciais [BCF+87] [CBC+93] [Lin98].

Relacionado com a geração deste grafo, surgem duas questões importantes. Em primeiro lugar a forma de o gerar tem que ser muito eficiente, pois em algumas redes é possível ter que lidar com um número muito elevado de marcações (ex. milhões de estados) resultantes da própria estrutura da rede. É importante notar que as marcações instáveis também contribuem para a complexidade desta tarefa, dado que para obter uma marcação estável é necessário gerar todas as marcações instáveis intermédias. Esta questão tem implicações a dois níveis: o tempo necessário para gerar o grafo, e os recursos computacionais (memória) necessários para o armazenar. Como forma de minimizar estes aspectos foram propostos recentemente algoritmos baseados em operadores de Kroneker [CT96] [Buc04],

que têm como objectivo gerar em cada instante apenas as partes do grafo que são necessárias (*on-the-fly*) para o cálculo do respectivo processo estocástico.

A segunda questão, está relacionada com a necessidade de gerar de um processo estocástico completamente bem determinado e sem inconsistências. Quando isto acontece é dito que a rede de Petri está *bem definida* (*well specified*) [CZ96] [ZCH97]. O problema de uma definição errada, pode acontecer quando existem conflitos e *confusões* (*confusions*) entre transições simultaneamente habilitadas onde seja necessário recorrer a políticas de disparo de pré-selecção [CMB+93]. Grosso modo, o problema reside no facto do percurso probabilístico não ser independente da sequência escolhida para o disparo das transições. Por exemplo, a probabilidade de ir de uma marcação estável para outra depende da sequência de disparo realizada, ficando assim uma dúvida sobre o comportamento estocástico do sistema. Uma das possibilidades de resolver este problema é atribuir pesos e prioridades a todas as transições de tal forma que o problema seja eliminado. Contudo esta abordagem só é prática se a rede for de reduzidas dimensões. Uma alternativa consiste na determinação dos conjuntos de conflitos extendidos (ECS - *Extended Conflict Set*) [CMB+93] [MBC+95] no momento em que o grafo é gerado. Estes conjuntos permitem definir e localizar grupos de transições que apresentem diferentes tipos de conflitos entre si. Com esta informação, mais precisa, e utilizando também os pesos e as prioridades é possível resolver o problema de forma mais simples.

2.7.4 Redes de Petri Estocásticas Markovianas

As redes de Petri Estocásticas Markovianas são caracterizadas pelo facto do seu processo de marcação ser um processo estocástico Markoviano. Este processo, resulta da imposição de um conjunto de restrições às transições da rede. Desta forma, se uma rede de Petri possuir as seguintes características:

- ♦ Todas as transições são temporizadas;
- ♦ As transições possuem um tempo de disparo de acordo com uma distribuição exponencial;
- ♦ As políticas de preempção associadas às transições são do tipo PRD ou PC.

então, o processo de marcação da rede resulta numa cadeia de Markov Contínua, cujo espaço de estados é isomórfico em relação ao grafo das marcações acessíveis da rede de Petri não-temporizada subjacente [Mol82]. Estas redes são denominadas Redes de Petri Estocásticas (SPN - *Stochastic Petri Nets*).

De forma análoga, quando todas as transições possuem uma distribuição geométrica, o processo de marcação da rede resulta numa cadeia de Markov

Discreta [Mol85]. Este último caso é equivalente a uma discretização do tempo em múltiplos de um incremento temporal Δt .

As redes de Petri Estocásticas Generalizadas (GSPN - *Generalized Stochastic Petri Nets*) [MCB84] resultam de uma generalização dos casos anteriores, permitindo a inclusão quer de transições exponenciais, quer imediatas. Neste caso, e se a rede estiver bem definida, o grafo reduzido das marcações acessíveis é também isomórfico em relação a uma cadeia de Markov Contínua. A situação é análoga ao caso anterior para uma distribuição geométrica. Dado que os processos estocásticos subjacentes a estas redes são cadeias de Markov (Contínuas e Discretas), são utilizadas as técnicas de resolução definidas para estas últimas para obter as respectivas soluções.

Importa aqui fazer um comentário importante sobre a terminologia utilizada para denominar as redes. Embora alguns autores o utilizem o termo *redes de Petri Temporizadas e Estocásticas* [Wang98] para se referirem de forma geral à classe das redes de petri que representam processos estocásticos, na esmagadora maioria dos casos a literatura refere-se a estas como *redes de Petri Estocásticas*. Isto pode trazer alguma confusão, porque este termo é o mesmo que é utilizado para definir as SPN. Referido este aspecto e por forma a evitar confusões, optou-se nesta dissertação por adoptar a terminologia vulgarmente utilizada na literatura, ou seja *redes de Petri Estocásticas*.

2.7.5 Redes de Petri Estocásticas Não-Markovianas

As redes de Petri Estocásticas Não-Markovianas foram desenvolvidas com vista a ultrapassar as limitações anteriores, nomeadamente na extensão do tipo de políticas de preempção permitidas e na inclusão de transições cuja função distribuição não necessite de ser exponencial. Como resultado destas extensões, o respectivo processo de marcação é do tipo Não-Markoviano [CGL94].

Dado o interesse em obter processos estocásticos analiticamente tratáveis, é necessário impor várias restrições às características das redes, nomeadamente ao nível das suas transições por forma a obter um processo de marcação adequado. A existência de diferentes técnicas de solução para este tipo de processos estocásticos, cada uma com um conjunto de limitações bem definidas, acabou por resultar numa abordagem, ao nível das propostas para este tipo de redes, fortemente centrada nas soluções dos vários tipos de processos.

Para facilitar a análise deste tipo de redes é comum agrupar as respectivas transições em várias classes, de acordo com as seguintes características:

- ♦ Função distribuição associada às transições: exponencial ou generalizada. Neste último caso, a função distribuição pode incluir qualquer distribuição contínua ou discreta, ou até uma mistura de ambas;
- ♦ Concorrência entre transições [DTG+84]:
 - **Exclusiva.** Uma transição t_i é dita exclusiva se para qualquer marcação M a única transição habilitada é t_i ;
 - **Competitivas.** Seja t_i uma transição não-exclusiva e M uma marcação em que t_i e uma outra transição qualquer t_j estão ambas habilitadas. Se para qualquer transição t_j na marcação M , o disparo de t_j desabilita a transição t_i , então t_i é dita uma transição competitiva;
 - **Concorrentes.** Seja t_i uma transição não-exclusiva e M uma marcação em que t_i e uma outra transição qualquer t_j estão ambas habilitadas. Se para qualquer transição t_j na marcação M , o disparo de t_j não desabilita a transição t_i então t_i é dita uma transição concorrente.

Se todas as transições concorrentes forem do tipo exponencial e se as transições generalizadas forem competitivas e se em ambos os casos a política de execução for do tipo PRD, o processo de marcação da rede corresponde a um processo de Semi-Markov [MC87]. Estas redes são denominadas redes de Petri Estocásticas e Extendidas (ESPN - *Extended Stochastic Petri Nets*).

As redes de Petri Estocásticas Regenerativas (MRSPN – *Markov Regenerative Stochastic Petri Nets*) são caracterizadas pelo facto do seu processo de marcação ser um processo de Markov Regenerativo. As características deste processo, nomeadamente ao nível do seu processo subordinado, são função do tipo de política de execução das transições e do grau de concorrência entre transições generalizadas [TBP96] [BT98]. Saliente-se que nem todas as combinações dos factores anteriores resultam em processos Regenerativos. As situações mais comuns são as seguintes:

- ♦ Em cada marcação M no máximo uma única transição generalizada pode estar concorrentemente habilitada, sendo as restantes transições habilitadas exponenciais. Todas as transições possuem uma política de execução do tipo PRD ou PC. O processo subordinado é uma cadeia de Markov Contínua [CGL94];
- ♦ Todas as transições temporizadas são exponenciais com uma política de execução PRI. O processo subordinado é do tipo Semi-Markov [BKP+95];
- ♦ Em cada marcação M várias transições generalizadas com política de execução PRD podem estar competitivamente habilitadas, desde que tenham sido habilitadas no mesmo instante. O processo subordinado é do tipo Semi-Markov [PST98];

- ♦ Em cada marcação M várias transições generalizadas com qualquer política de execução podem estar competitivamente habilitadas, desde que sejam transições dominantes não-sobrepostas (*non-overlapping cycles*). O processo subordinado é do tipo Semi-Markov [BT95] [TBP96];
- ♦ Em cada marcação M várias transições generalizadas com política de execução PRD podem estar concorrentemente habilitadas. O processo de marcação é do tipo Semi-Markov Generalizado, com uma cadeia de Markov de Espaço de Estados Generalizado como processo subordinado [LS96].

O desenvolvimento deste tipo de redes foi realizado de forma muito próxima com o tipo de técnicas associadas à resolução de processos de Markov Regenerativos [PTT97] [BPT+98]:

- ♦ Teoria da Renovação;
- ♦ Varáveis Suplementares;
- ♦ Expansão por Fases.

A primeira linha de abordagem teve origem num caso particular de redes, onde em cada marcação é permitida apenas uma única transição com tempo de disparo determinístico e política de execução PRD, dando origem às redes de *Petri Determinísticas e Estocásticas* (DSPN - *Deterministic and Stochastic Petri Nets*) [MC87] [CL93]. Uma generalização do modelo anterior permitiu a introdução de transições generalizadas e com política de execução PC [BT95] e PRI [BKP+95]. A abordagem mais geral em que esta técnica foi aplicada, diz respeito à situação em que o período de regeneração é dominado por uma única transição dominante (*non-overlapping dominant transitions*) [TBP96]. Mais recentemente foram desenvolvidas as redes de *Petri Concorrentes e Generalizadas* (CGPN - *Concurrent Generalized Petri Nets*), onde é possível ter várias transições generalizadas concorrentemente habilitadas, desde que todas tenham sido habilitadas no mesmo instante e que nenhuma outra transição generalizada possa ser habilitada enquanto as anteriores não forem desabilitadas [PST98]. De forma equivalente, mas particularizando para as transições determinísticas, resultaram as rede de *Petri Determinísticas e Estocásticas em Cascata* (*Cascaded DSPN*), em que as várias transições generalizadas têm tempos de disparo múltiplos [Ger00].

A segunda linha de abordagem, foi unicamente utilizada para transições generalizadas exclusivas com políticas de execução do tipo PRD e PC [Ger00] [Ger99]. No caso das transições serem determinísticas a restrição pode ser relaxada, permitindo alguma concorrência entre transições generalizadas (no máximo duas transições) [LS96] [Lin98]. Neste último caso, a abordagem passa por definir um processo Generalizado de Semi-Markov, que pode ser interpretado como uma generalização do conceito de variável suplementar. É possível

demonstrar a equivalência entre as variáveis suplementares e os processos de Markov Regenerativos [GLT95] [Ger00].

Na terceira linha de abordagem, o processo estocástico subjacente à rede de Petri é aproximado por uma cadeia de Markov Contínua [HMC97] [BHT04]. Tem como principal vantagem a possibilidade de combinar qualquer política de memória e possuir qualquer número de transições generalizadas concorrentes. A grande desvantagem é a possibilidade da explosão do espaço de estados, que pode ser minimizada através da utilização de operadores de Kroneker em casos particulares [CT96] [Buc04]. Ultimamente esta abordagem têm passado pela utilização de uma expansão por fases em tempo discreto (*Discrete phase-type distributions*) [BH02], de forma em que um processo de tempo contínuo é aproximado por uma cadeia de Markov Discreta [HPS+00]. Deste tipo de abordagem resultaram as redes de *Petri Discretas Determinísticas e Estocásticas* (DDSPN - *Discrete Deterministic and Stochastic Petri Nets*) [ZC96] [ZCH97], em que é permitido ter qualquer grau de concorrência entre transições determinísticas (as transições exponenciais são representadas por distribuições geométricas, a sua “equivalente” em tempo discreto).

Num contexto um pouco diferente, importa referir o conceito de redes de *Petri Fluídas e Estocásticas* (FSPN – *Fluid Stochastic Petri Nets*) [HKN+98] [TK03]. Neste tipo de redes existem dois tipos de posições: discretas, com a mesma interpretação assumida até agora, e contínuas que contêm um *fluido*. Este fluido, corresponde ao equivalente contínuo dos testemunhos. As transições, para além de movimentarem testemunhos, permitem também o fluxo de fluido entre posições. Para além da representação de muitos processos físicos, esta abordagem permite também uma redução na dimensão do grafo das marcações acessíveis.

Por fim importa discutir qual o grau de suporte dos vários tipos de redes de Petri que foram apresentadas neste capítulo e as ferramentas de modelação existentes (com o objectivo de obter uma solução analítica). No caso de tempo contínuo, existem várias ferramentas disponíveis (uma descrição mais detalhada das características das mesmas pode ser encontrada em [HN96] e [Ger00]), contudo a esmagadora maioria suporta apenas uma (1) transição generalizada concorrentemente habilitada (a única excepção conhecida vai para a ferramenta DSPNexpress [Lin98]), sendo as políticas de memória limitadas aos tipos PRD e PC (PRD é o caso mais frequente). No caso de tempo discreto, o número de ferramentas é muito mais limitado, contudo existe pelo menos uma (TimeNet [ZFG+99]) que suporta as redes de Petri Discretas Determinísticas e Estocásticas com políticas de memória PRD e PC. Para os restantes tipos de redes de Petri Estocásticas as soluções são normalmente obtidas utilizando programas de computação simbólica (ex. Maple) ou numérica (ex. Matlab).

2.8 Simulação

A simulação é uma das técnicas mais utilizadas para analisar sistemas que possuem comportamentos estocásticos. A sua principal vantagem está no seu poder de modelação, aliada à possibilidade de permitir uma descrição intuitiva do comportamento do sistema por parte de utilizadores que não são necessariamente especialistas. O poder de modelação, reflecte-se no facto de não existirem praticamente quaisquer restrições em relação ao comportamento do sistema que se pretende modelar. Como consequência, os modelos de simulação podem possuir o nível de detalhe que se julgue necessário.

Dado que nesta dissertação a simulação foi apenas utilizada para obter as soluções de modelos baseados em rede de Petri Estocásticas, a discussão realizada nesta secção far-se-á neste contexto.

A utilização de simulação para obter soluções de modelos baseados em redes de Petri Estocásticas pode decorrer de dois motivos. O mais comum, resulta da impossibilidade de obter uma solução analítica/numérica do modelo. Como foi discutido nas secções anteriores, a existência de uma solução deste tipo só é possível quando a rede possui características estruturais particulares. O segundo motivo, está relacionado com a dimensão do espaço de estados. Mesmo existindo a possibilidade de uma solução analítica/numérica, o espaço de estados gerado pode ser de tal forma elevado que os recursos computacionais necessários para obter uma solução deste tipo não são exequíveis.

Embora um modelo de simulação possa ser desenvolvido de muitas formas, existem várias vantagens em utilizar uma abordagem baseada nas redes de Petri, das quais se destacam as seguintes:

- ♦ Poder de modelação elevado, permitindo modelar de forma compacta comportamentos muito complexos;
- ♦ Validação formal do modelo utilizando as propriedades e as estruturas de análise derivadas da rede de Petri;
- ♦ Desenvolvimento flexível devido ao carácter gráfico do modelo;
- ♦ A existência de uma semântica bem definida permite desenvolver e validar modelos de forma mais simples do que a construção de uma programa de simulação de raiz;
- ♦ Mapeamento imediato entre a interpretação semântica da rede e os sistemas de eventos discretos (DES – *Discrete Event Systems*) [Cass93] [Haa02];

- ♦ Independência entre o modelo desenvolvido e as técnicas computacionais utilizadas para obter a respectiva solução;
- ♦ Existência de muitas ferramentas de modelação de redes de Petri que já incorporam de forma automática as várias técnicas de simulação [JM88] [TM93] [HN96].

Dado que os conceitos associados às redes de Petri foram já apresentados, no restante desta secção far-se-á uma apresentação dos aspectos mais importantes associados à simulação de eventos discretos.

A simulação de *eventos discretos* (*Discrete-Event Simulation*) [Tys99] [LK00] [BCN+00] [Ros02] é das técnicas existentes a mais utilizada para obter modelos de simulação de sistemas que possuem comportamentos estocásticos. Esta técnica baseia-se em dois conceitos chave: a geração de números aleatórios e a gestão de eventos. A geração de números aleatórios é um aspecto muito crítico, porque estes estão directamente associados à amostragem dos valores das funções distribuição (de probabilidade). Como o objectivo é representar o comportamento de um processo estocástico, resulta de forma imediata que o grau de exactidão com que este comportamento vai ser representado está directamente relacionado com a qualidade (*aleatoriedade*) dos números aleatórios gerados.

A gestão de eventos está associada com a evolução temporal do sistema. Esta evolução pode ser descrita através de uma sequência de eventos que ocorrem em instantes particulares do tempo e que representam mudanças no estado do sistema. Esta sequência temporal de eventos implica que o modelo de simulação necessite de manter um relógio (interno) como o objectivo de definir as acções a executar no modelo nos instantes de ocorrência dos eventos.

Utilizando estes dois conceitos a execução de um modelo de simulação de eventos discretos pode ser descrito de acordo com o seguinte algoritmo:

1. O modelo é inicializado;
2. Se o período de simulação expirou ou se as condições de paragem estão satisfeitas, a simulação é considerada terminada;
3. O relógio associado à simulação avança automaticamente para o próximo evento. Este é definido como aquele cujo instante temporal de ocorrência é o mais próximo. Neste sentido os eventos são mantidos numa lista apropriada (*event list*) por ordem cronológica de ocorrência;
4. Executar as acções (no modelo) referentes à ocorrência deste evento. Da execução destas acções resultam:
 - i. A definição de um novo estado para o sistema;

- ii. Novos eventos são adicionados à lista de eventos. Os instantes de ocorrência dos novos eventos são obtidos a partir da amostragem das respectivas funções distribuição;
- iii. Remoção de eventos previamente definidos da lista de eventos.

5. Re-ordenar a lista de eventos e ir para o ponto 2.

Conforme pode ser observado a partir das características do algoritmo anterior, a execução do modelo de simulação pode ser realizada de forma muito eficaz. O comportamento do sistema só é observado em instantes particulares do tempo (quando ocorrem os eventos), avançando de forma imediata para o instante seguinte. Disto resulta uma compressão do factor tempo. Dito de outra forma, a evolução do sistema durante um intervalo de tempo real é obtida, tipicamente, num tempo de simulação várias ordens de grandeza inferior. Esta compressão do tempo, é muito dependente das características do sistema que se está a simular, tais como: a sua complexidade, a escala temporal dos seus parâmetros, o número de eventos a processar, etc.

Em termos formais, a execução do modelo de simulação é equivalente à realização de uma experiência de acordo com um conceito probabilístico, e os resultados obtidos a partir desta execução são simplesmente amostras de variáveis aleatórias. No contexto desta dissertação, estas variáveis são simplesmente as medidas de interesse que se pretende obter do modelo.

Dado o carácter probabilístico das medidas, é necessário recorrer a uma estatística para estimar as propriedades destas variáveis aleatórias. Como a respectiva função distribuição é geralmente desconhecida, as propriedades estimadas concentram-se, na maioria dos casos, na estimação de parâmetros desta distribuição, como é o caso do valor esperado (média) e da variância. Este tipo de estimação é realizada por intermédio de uma função denominada *estimador pontual* [Cass93] [LK00]. Um estimador é uma função de variáveis aleatórias¹⁷, que possui um conjunto de propriedades bem definidas:

- ♦ A função distribuição do estimador é conhecida, ou o seu comportamento pode ser aproximado por funções distribuições conhecidas;
- ♦ Alguns dos parâmetros da função distribuição do estimador, reflectem directamente parâmetros equivalentes da função distribuição das medidas que se pretendem obter.

¹⁷ Um estimador é também uma variável aleatória.

A estimação da média e da variância de uma variável aleatória implica a construção de dois estimadores (variáveis aleatórias), cujo valor esperado corresponde ao valor esperado (média) e à variância das medidas que se pretendem determinar. Um estimador que se comporte desta forma é dito *não-enviesado*.

Sem perda de generalidade e como objectivo apenas de concretizar os conceitos apresentados, admita-se que o modelo de simulação é executado de forma independente M vezes, resultando cada uma dessas execuções numa amostra X_i da variável aleatória X referente à medida que se pretende obter.

Um estimador $\hat{\mu}$ não enviesado do valor esperado de X ($\mu = E[X]$) é dado pela média amostral

$$\hat{\mu} = \frac{1}{M} \sum_{i=1}^M X_i \quad (2.76)$$

O estimador $\hat{\mu}$ é uma variável aleatória e que possui as seguintes propriedades

$$\begin{aligned} \mu &= E[\hat{\mu}] \\ \text{Var}[\hat{\mu}] &= \frac{\sigma^2}{M} \end{aligned} \quad (2.77)$$

em que $\sigma^2 = \text{Var}[X]$. Como pode ser observado, à medida que o número de amostras aumenta a sua variância vai-se reduzindo, o que significa que o estimador toma valores cada vez mais próximos do (verdadeiro) valor esperado. A questão que falta definir é qual esse grau de proximidade.

Se a função distribuição do estimador for conhecida (ou estimada), é possível definir um *intervalo de confiança* para o estimador. Concretizando de novo através do exemplo acima indicado, a variância amostral é definida como

$$s^2 = \hat{\sigma}^2 = \frac{1}{M-1} \sum_{i=1}^M (X_i - \hat{\mu})^2 \quad (2.78)$$

que por sua vez é também um estimador não enviesado da variância $\sigma^2 = \text{Var}[X]$. A quantidade s^2/M é uma estimativa da variância de $\hat{\mu}$.

Partindo destes elementos, demonstra-se que a variável aleatória [Triv02]

$$t = \frac{\hat{\mu} - \mu}{s/\sqrt{M}} \quad (2.79)$$

tem uma função distribuição aproximadamente *t-Student* com $M-1$ graus de liberdade. Assim, se $t_n(x)$ for o percentil x (%) da distribuição *t-Student* com n graus de liberdade

$$Prob\left\{t_{M-1}\left(\frac{\alpha}{2}\right) \leq \frac{\hat{\mu} - \mu}{s/\sqrt{M}} \leq t_{M-1}\left(1 - \frac{\alpha}{2}\right)\right\} = 1 - \alpha \quad (2.80)$$

Como a distribuição *t-Student* é simétrica em torno de 0, a expressão anterior pode ser re-escrita obtendo-se o seguinte **intervalo de confiança**

$$\mu \in \left[\hat{\mu} - t_{M-1}\left(1 - \frac{\alpha}{2}\right) \frac{s}{\sqrt{M}}, \hat{\mu} + t_{M-1}\left(1 - \frac{\alpha}{2}\right) \frac{s}{\sqrt{M}} \right] \quad (2.81)$$

em que

$$t_{M-1}\left(1 - \frac{\alpha}{2}\right) \frac{s}{\sqrt{M}} \quad (2.82)$$

é a semi-amplitude do intervalo, e $(1 - \alpha)$ é o *nível de confiança* do intervalo.

O intervalo de confiança deve ser interpretado da seguinte forma: o intervalo tem uma probabilidade $(1 - \alpha)$ de conter o valor $\mu = E[X]$. Tipicamente pretendem-se intervalos com pequenas amplitudes (largura) e com níveis de confiança elevados.

Embora este seja apenas um exemplo, é possível extrair um conjunto de conclusões que permitem caracterizar os vários aspectos relacionados com a determinação do intervalo de confiança. O primeiro aspecto, relaciona-se com a necessidade de se obter um número elevado de amostras M para obter um intervalo de confiança com uma largura estreita (admitindo um nível de confiança elevado). Este número de amostras pode ser obtido realizando muitas simulações ou uma simulação de longa duração. O segundo aspecto, relaciona-se com a variância das amostras. Quanto maior é a variância, mais largo é o intervalo (para o mesmo número de amostras). Esta situação pode ocorrer quando se simulam sistemas onde os eventos que se pretendem observar são muito raros de acontecer, contribuindo assim para uma elevada variância dos resultados. O terceiro aspecto, está associado ao facto de que a construção dos estimadores se basear na definição de uma função de variáveis aleatórias independentes e identicamente distribuídas. Isto implica que a execução do modelo de simulação tenha que ser realizada por forma a obter amostras com estas características.

A construção do estimador, é função do tipo de simulação pretendida. Esta é normalmente classificada da seguinte forma [LK00]:

- ♦ **Terminal** (*Terminating Simulation*). O modelo de simulação é executado até ocorrer um evento específico. Este evento pode ser definido por um valor do relógio (de simulação) pré-definido ou por uma condição obtida do estado do sistema. Este tipo de simulação é por vezes denominada *transitória*;
- ♦ **Não-terminal** (*Non-terminating Simulation*). Neste caso não existe um ponto de paragem natural, dado que em teoria está-se interessado no comportamento do sistema quando $t \rightarrow \infty$. É assim utilizada para analisar o comportamento do sistema em regime estacionário. Quanto mais longa for a simulação, melhores serão os resultados obtidos. No entanto questões práticas limitam esta abordagem.

Um dos problemas que existe com este tipo de simulação diz respeito ao facto de que as amostras obtidas durante o período inicial da simulação (*warm-up period*) não serem representativas do comportamento em regime estacionário e assim contribuirão para obter estimadores enviesados. É assim necessário dispor de técnicas que permitam limitar ou minimizar o problema.

Neste tipo de simulação os resultados são tipicamente obtidos recorrendo a uma das seguintes técnicas [Tys99] [LK00] [BCN+00] [Ros02]:

- **Replicações independentes**¹⁸. O modelo de simulação é executado N vezes, em que cada execução é considerada independente das restantes. Uma das formas de garantir esta independência é utilizar diferentes sequências de números aleatórios em cada simulação. A utilização desta técnica implica contudo que a componente transitória das amostras seja removida ou minimizada (o que nem sempre é fácil de realizar) por forma a obter estimadores credíveis;
- **Média dos lotes** (*Batch Means*). O modelo de simulação é executado uma única vez, mas os resultados obtidos são divididos N lotes com uma dimensão conhecida. A ideia por detrás desta abordagem está no pressuposto de que lotes são independentes. Obtêm-se assim uma técnica semelhante às replicações independentes, mas com a vantagem de não ser necessário remover (excepto um vez) a componente transitória das amostras. É assim mais eficaz. O problema que se coloca nesta técnica, é determinar qual a dimensão dos lotes, estando este aspecto directamente associado com a respectiva independência. Isto pode ser estimado obtendo a função de covariância entre lotes e garantindo que esta função toma valores muito reduzidos, ou em alternativa, utilizar lotes com uma grande dimensão;

¹⁸ Esta técnica é também utilizada nas simulações terminais.

- **Método regenerativo.** Neste caso é assumido que em termos estocásticos o sistema se comporta como um processo regenerativo. Tal como na técnica anterior, é apenas realizada uma simulação mas em que os lotes são definidos pelos pontos de regeneração do processo. Garante-se assim à partida que os lotes são estocasticamente independentes. A dificuldade em utilizar esta técnica está na determinação dos pontos de regeneração do processo, que em muitos dos casos podem ser muito longos.

A utilização de técnicas de simulação para obter a solução de modelos de avaliação da confiança no funcionamento, traduz-se normalmente pela existência de alguns problemas de eficácia ao nível do tempo de simulação exigido. Este aspecto é uma consequência das características deste tipo de modelos (sistemas), nomeadamente pelo facto de que os eventos de interesse (falhas, avarias, etc.) ocorrerem com escalas de temporais muito grandes (*rare events*) [LK00] [Ros02] [NSN03]. Ao nível da simulação, isto implica que sejam obtidas comparativamente poucas amostras (das medidas de interesse) num determinado horizonte temporal, o que por sua vez geralmente implica também uma maior variância das mesmas. Como foi discutido previamente, estes dois factores são os que mais contribuem para a definição do intervalo de confiança. Ou seja, se se pretender um intervalo de largura reduzida com um nível de confiança elevado, é necessário dispor simultaneamente de um número elevado de amostras com uma variância reduzida. No contexto destes sistemas, isto é equivalente a realizar um número muito grande de simulações o que se traduz em tempos de simulação também muito grandes. Ao nível dos sistemas muito fiáveis (*high & ultra-dependable systems*) e devido à raridade com que os fenómenos de interesse ocorrem, este problema toma dimensões dramáticas, sendo na maioria dos casos praticamente impossível obter uma solução num tempo de simulação exequível [GS89].

Como resultado deste problema, a utilização da simulação para obter a solução de modelos de avaliação da confiança no funcionamento foi sempre vista com algumas reticências por parte da comunidade científica associada a esta área [PT00]. Este foi também um dos motivos pelos quais esta comunidade realizou nas últimas décadas um forte investimento na investigação e utilização de técnicas analíticas/numéricas para obter a solução deste tipo de modelos. O objectivo é aliar a ausência de restrições ao nível do desenvolvimento do modelo, com uma maior eficácia ao nível da sua solução, permitindo obter assim uma técnica de avaliação extremamente poderosa. Estas técnicas são normalmente denominadas *de simulação rápida* (*Fast Simulation Techniques*) [NSN03], sendo usualmente classificadas da seguinte forma:

- ♦ **Estrutura computacional.** A rápida evolução tecnológica têm vindo a disponibilizar um conjunto de novos recursos computacionais (CPU, memória, etc.) quer permitem obter níveis cada vez mais elevados de poder de

cálculo, a custos cada vez mais baixos. Isto têm permitido que os tempos de simulação tenham vindo a decrescer de forma dramática.

Além disso, hoje em dia é relativamente simples poder dispor destes recursos sob a forma de um ambiente de computação distribuído [PMM+03]. Permite-se assim que a simulação possa ser distribuída por várias máquinas, acelerando ainda mais o processo de obter a solução. No contexto da simulação com redes de Petri, são várias as ferramentas que já disponibilizam um ambiente deste tipo [HN96] [ZFG+99] [CCD+01];

- ♦ **Técnicas de minimização da variância.** As técnicas de minimização da variância baseiam-se na redefinição do estimador através da introdução no mesmo de outras medidas que estejam correlacionadas com a medida que se pretende obter [Tys99] [LK00] [BCN+00] [Ros02]. Como mero exemplo para concretizar este conceito, considere-se que se dispõe de uma outra medida Y (no modelo) em que $\alpha = E[Y]$, e em que V é um estimador de $\mu = E[X]$ definido da seguinte forma

$$V = \hat{\mu} + \alpha - \hat{\alpha} \quad (2.83)$$

em que $\hat{\alpha}$ é um estimador de $\alpha = E[Y]$. Como se pode verificar

$$E[V] = E[\hat{\mu}] + \alpha - E[\hat{\alpha}] = \mu - \alpha + \alpha = \mu \quad (2.84)$$

Calculando a variância de V obtêm-se

$$\text{Var}[V] = \text{Var}[\hat{\mu}] + \text{Var}[\alpha] - 2\text{Cov}[\hat{\mu}, \hat{\alpha}] \quad (2.85)$$

em que $\text{Cov}[\hat{\mu}, \hat{\alpha}]$ é a função de co-variância. A variância do novo estimador V será sempre menor que $\text{Var}[\hat{\mu}]$ desde que $\text{Cov}[\hat{\mu}, \hat{\alpha}] > \frac{1}{2}\text{Var}[\hat{\alpha}]$.

As técnicas mais comuns a este nível são: as variáveis de controlo (*control variates*), as variáveis antitéticas (*antithetic variates*), os valores esperados condicionados e os geradores de números aleatórios comuns.

Embora estas técnicas não envolvam qualquer modificação no modelo, a dificuldade da sua utilização está em determinar outra medida que esteja correlacionada (de forma positiva ou negativa) com a aquela que se pretende obter.

Ao nível das redes de Petri, a ferramenta TimeNet [Kel95] implementa de forma automática a técnica das variáveis de controlo a partir de uma análise estrutural da rede de Petri.

- ♦ **Amostragem de importância** (*Importance Sampling*). Esta técnica baseia-se na modificação do comportamento estocástico do modelo, através do aumento da frequência da ocorrência dos eventos que interessa observar [GS89] [Ros02] [NSN03]. O modelo está assim enviesado, mas este enviesamento é compensado através da introdução de uma função denominada *razão de verosimilhança* (*likelihood ratio*), que remove o enviesamento e permite reduzir a variância. Informalmente esta razão é a relação entre a probabilidade de obter as observações utilizando a medida de probabilidade original e a probabilidade de obter as observações utilizando a nova medida.

Com o objectivo de concretizar este conceito considere-se o seguinte exemplo. Suponha-se que se pretende estimar o valor esperado de uma função $g(x)$ de uma variável aleatória X com uma função densidade de probabilidade $f_x(x)$

$$E[g(X)] = \int g(x) f_x(x) dx \quad (2.86)$$

Em vez de amostrar $X_1, \dots, X_n$ a partir da distribuição $f_x(x)$ e obter a estimativa utilizando a média amostral, as amostras são obtidas de uma outra distribuição $\hat{f}_x(x)$, definindo-se uma nova medida Y como

$$Y = \frac{1}{n} \sum_{i=1}^n \frac{g(X_i) f_x(X_i)}{\hat{f}_x(X_i)} \quad (2.87)$$

Note-se $E[Y] = E[g(X)]$ e que a respectiva variância é dada por

$$V[Y] = \frac{1}{n} \left[\int \frac{g^2(x) f_x^2(x)}{\hat{f}_x^2(x)} \hat{f}_x(x) dx - E[g(X)]^2 \right] \quad (2.88)$$

Se se escolher $\hat{f}_x(x) = f_x(x) g(x) / E[g(x)]$ o estimador passa a ter variância zero e desta forma seria apenas necessário executar uma única simulação! Note-se contudo, que o valor do denominador da expressão anterior é precisamente o valor que se pretende estimar.

Embora a técnica ofereça condições excepcionais ao nível da simulação, sofre do problema da determinação da função razão de verosimilhança. Esta determinação é normalmente complexa e depende muito das características do sistema que se está a analisar. Mesmo com estas limitações, esta tem sido a técnica onde ultimamente se têm realizado mais progressos [NSN03].

Ao nível das redes de Petri, esta técnica foi implementada na ferramenta UltraSAN [OS94] (antecessor da ferramenta Möbius utilizada no capítulo 5) e também nas ferramentas SHARPE (*Symbolic Hierarchical Automated Reliability and Performance Evaluator*) [HSZ+00] e SPNP (*Stochastic Petri Nets Package*) [HTT02].

Ainda no contexto das técnicas de simulação rápidas, importa referir a simulação de eventos raros implementada na ferramenta TimeNet [Kel96] [KH96]. Esta técnica é baseada no princípio da divisão (*splitting*) dos percursos amostrais, que a seguir se descreve. Admita-se que se pretende avaliar o comportamento de um sistema através da definição de um conjunto de medidas associadas a uma parte A do respectivo espaço de estados, e que essa parte é visitada muito raramente. Segundo esta técnica, é definida uma região do espaço de estados próxima da parte que interessa, e sempre que o processo de simulação entra nesta área são criadas várias cópias do processo de simulação. Estas cópias são executadas a partir do ponto de entrada na região de interesse e terminam a sua execução até saírem da mesma área. Durante a execução das cópias, se estas entrarem em regiões ainda mais próximas de A o processo de gerar novas cópias repete-se de novo, e assim sucessivamente. O objectivo é assim obter várias instâncias do processo a gastarem mais tempo nas regiões de interesse onde os eventos mais raros ocorrem mais frequentemente. A partir das amostras obtidos durante estas simulações e utilizando estimadores apropriados são obtidas as medidas pretendidas.

Por fim, importa referir o desenvolvimento recente de novas abordagens à simulação de processos estocásticos, que não utilizam o conceito de simulação de eventos discretos, nomeadamente a utilização de métodos baseados *proxels* [Hor02] [MH03]. Um *proxel*¹⁹ $P=(p, S)$ é utilizado para representar o estado S do processo e da respectiva probabilidade p . Para cada estado S , são determinados os seus sucessores e calculadas as respectivas probabilidades de transição, num intervalo de tempo discreto ΔT . À medida que a simulação decorre, os *proxels* são criados, guardados temporariamente e por fim destruídos. Esta é assim uma técnica determinística, que actua directamente ao nível do espaço de estados do processo e está muito próxima das equações que governam o comportamento do processo. Contudo, ao utilizar uma abordagem algorítmica, as equações que governam o comportamento do sistema não são resolvidas (tipicamente sistemas de equações diferenciais), mas usadas como suporte para determinar a evolução do sistema.

2.9 Conclusões

Neste capítulo foram apresentados os vários conceitos associadas à confiança no funcionamento, bem como as respectivas técnicas de avaliação baseadas em modelos estocásticos.

Na primeira parte do capítulo são apresentados os vários aspectos subjacentes à definição de confiança no funcionamento com o objectivo de introduzir os vários

¹⁹ O nome resulta de uma analogia ao termo *pixel* utilizado no processamento de imagem.

conceitos utilizados em capítulos subsequentes. Além disso, procurou-se utilizar uma terminologia em Português que tivesse sido alvo de discussão nos meios científicos nacionais com o objectivo de reforçar a adopção, por esta comunidade, de uma terminologia única e comum.

A segunda parte do capítulo foca-se essencialmente nas técnicas de avaliação da confiança no funcionamento baseadas em modelos de estocásticos. Inicialmente são apresentados e discutidos um conjunto de critérios com o objectivo de permitir realizar uma escolha fundamentada sobre o tipo de técnicas mais adequadas para avaliar os sistemas em causa. Desta discussão concluiu-se que as técnicas mais adequadas são as Cadeias de Markov (contínuas e discretas) e as Redes de Petri Estocásticas.

Subjacente às técnicas de modelação escolhidas estão um conjunto de processos estocásticos (Markovianos e Não-Markovianos), cujas características mais importantes são apresentadas e discutidas em detalhe. São apresentadas as metodologias mais actuais com vista a obter as soluções analíticas dos mesmos.

A importância das redes de Petri nos modelos propostos nesta dissertação mereceu uma discussão muito alargada, sendo apresentadas as características deste formalismo, quer ao nível estrutural quer de dinâmica de funcionamento, bem como os vários tipos de extensões que foram adicionadas à definição original. Tendo em conta que as redes de Petri Estocásticas tem a capacidade de suportar diferentes tipos de processos estocásticos, são apresentadas e analisadas as características estruturais de diferentes alternativas que foram propostas ao longo das últimas décadas, sendo também discutidas as respectivas limitações.

Dado que em alguns dos modelos propostos nesta dissertação a solução foi obtida através de simulação (porque a solução analítica não era exequível), a última parte do capítulo é dedicada à discussão dos conceitos associados à utilização da simulação, na perspectiva do modelo ser desenvolvido utilizando redes de Petri. São apresentadas diferentes técnicas de simulação, incluindo as de simulação rápida, e discutidas as respectivas características e limitações. Dento deste contexto, são também indicados as funcionalidades suportadas pelas várias ferramentas de modelação de redes de Petri.

Esta página foi intencionalmente deixada em branco

Capítulo 3

Avaliação Preliminar da Confiança no Funcionamento

3.1 Introdução

Durante as fases iniciais do ciclo de vida de um sistema, nomeadamente durante a respectiva especificação e desenvolvimento, está-se muitas vezes confrontado com a necessidade de dispor de metodologias adequadas que permitam realizar de forma eficaz uma avaliação preliminar da respectiva confiança no funcionamento. Esta avaliação vai permitir identificar as principais limitações existentes a diferentes níveis e assim contribuir para a melhoria da especificação do sistema [Lap95] [IEC04a].

Enquanto que uma avaliação global e completa (ou seja a validação do sistema) é essencial para avaliar até que ponto o funcionamento do sistema está de acordo com a sua especificação, uma avaliação preliminar é importante no sentido em que permite obter resultados que podem ser utilizados na tomada de decisões entre propostas de concepção alternativas durante a fases iniciais do seu ciclo de vida. Além disso, é frequente durante estas fases não se dispor de informações que permitam realizar uma avaliação global, ou porque estas simplesmente não estão disponíveis ou porque ainda não se dispõe de uma caracterização completa e

detalhada sobre o comportamento do sistema. Acresce por fim, que o tempo e os recursos utilizados para realizar este tipo de avaliações nestas fases são tipicamente reduzidos.

Mesmo após se ter obtido um modelo de avaliação que se julga adequado, é ainda necessário que os parâmetros utilizados na sua resolução estejam em conformidade com a realidade. Muitas das vezes, e por ausência de informações, o modelo é desenvolvido com base em pressupostos pessimistas, e às vezes também irrealistas, o que por sua vez implica que as medidas de interesse obtidas possuam uma natureza semelhante.

No contexto do desenvolvimento de sistemas de controlo distribuídos suportados por redes de campo, os problemas enfrentados são semelhantes pelo que é necessário dispor também de uma forma de realizar esta avaliação preliminar.

Ao nível das redes de campo existe ainda uma situação, que embora num contexto completamente diferente, acaba por resultar num problema com características semelhantes ao anteriormente descrito. Quando se adquire uma rede comercial é frequente não se dispor de quaisquer tipo de informações que permitam caracterizar a ocorrência de avarias nos componentes da rede. Desta forma, os parâmetros dos modelos são praticamente desconhecidos, podendo apenas ser estimados com base em comportamentos típicos. Além disso, como se desconhece em detalhe o processo de especificação e desenvolvimento da rede, não é possível desenvolver um modelo de avaliação que inclua muitos detalhes, sendo desta forma necessário adoptar uma abstracção de alto nível. Estes dois factos combinados, resultam num problema semelhante ao da avaliação preliminar.

Face a este conjunto de aspectos é legítimo colocar as seguintes questões: “o que se pretende avaliar?”, e “que tipos de modelos se pretendem desenvolver?”.

Nas fases iniciais do ciclo de vida está-se particularmente interessado em realizar avaliações comparativas entre diferentes tipos de arquitecturas da rede²⁰, que por sua vez resultam da estrutura funcional que foi adoptada ao nível da aplicação distribuída que controla o processo físico. As avaliações exigem uma consistência ao nível dos pressupostos, dos parâmetros e da metodologia utilizada, mas não no rigor absoluto dos resultados, dado que se tratam de avaliações comparativas. Esta avaliação é tipicamente realizada utilizando cenários de falhas simples, pois a informação existente é muito reduzida, e às quais correspondem as consequências mais significativas para o comportamento da rede.

²⁰ Por forma a simplificar a descrição do trabalho realizado, adoptou-se o termo *rede* para representar o sistema de controlo distribuído suportado por uma rede de campo. Esta terminologia será também empregue, quando necessária, nos capítulos subsequentes.

Neste contexto e face às restrições existentes ao nível dos dados existentes sobre funcionamento da rede e dos parâmetros associados às avarias dos componentes, o modelo de avaliação resulta necessariamente de uma abstracção de alto nível e focado nos componentes mais importantes da rede ao nível do seu funcionamento. Esta importância decorre das consequências das falhas no comportamento global da rede, sendo possível identificar os seguintes cenários: falhas permanentes que afectam os componentes da rede e falhas transitórias que afectam as comunicações.

As falhas permanentes vão causar o maior impacto quando afectam os nós onde são executadas as tarefas que compõem a aplicação de controlo distribuída. Devido à sua persistência temporal, estas falhas impedem a execução das tarefas ou inibem a possibilidade destas comunicarem. Isto produz um profundo impacto ao nível do controlo do processo, podendo eventualmente levar mesmo à ausência de acções de controlo sobre o mesmos, com todas as consequências daí decorrentes. No contexto das falhas transitórias, a sua ocorrência vai implicar a corrupção dos dados que são transmitidos entre as várias tarefas da aplicação. Este efeito reflecte-se na existência de perdas e atrasos nas respectivas transmissões, que por sua vez podem dar origem a perturbações na dinâmica de controlo do processo [SK92] [KS94]. Estes aspectos tornam-se especialmente críticos quando existem requisitos de tempo-real associados à transmissão de mensagens de controlo [Kop97].

Em ambos os cenários e decorrente do contexto da avaliação, os modelos propostos deverão ser suficientemente simples e com um número reduzido de parâmetros, mas com um rigor adequado que lhes permita reflectir de forma mais fiel possível o comportamento real da rede. Além disso, devem possuir uma natureza genérica mas suficientemente flexível que lhes permita serem adaptados ao comportamento específico de cada rede. A sua solução deverá exigir recursos computacionais simples, nomeadamente através da possibilidade de extrair dos modelos expressões analíticas facilitando assim a obtenção da solução.

Este capítulo representa uma versão extendida dos trabalhos publicados pelo autor em [PA01a] e [PA01b].

3.2 Discussão dos Trabalhos Relevantes Previamente Desenvolvidos

A esmagadora maioria dos trabalhos associados ao desenvolvimento de modelos para a avaliação da confiança no funcionamento de redes de campo, centram-se na obtenção de modelos que focam cenários de falhas muito específicos e tipicamente muito detalhados. São assim modelos que estão fora do contexto dos objectivos

definidos para este capítulo, e aos quais será dada a respectiva atenção nas discussões dos modelos propostos nos capítulos 4, 5 e 6.

São assim muitíssimo reduzidos o número de trabalhos que se podem classificar como de desenvolvimento de modelos visando uma avaliação preliminar. A discussão apresentada nesta secção focar-se-á assim nestes trabalhos, mas também em outros, que embora não versem directamente as redes de campo, o seu contexto e a respectiva abordagem estão próximos dos objectivos pretendidos neste capítulo justificando assim a respectiva menção.

Em [CMP+90] é realizada uma análise do comportamento de uma rede local industrial, com uma topologia baseada num anel duplo (*token ring*). São consideradas falhas permanentes que ocorrem nos nós da rede e que têm como consequência a respectiva avaria, o que afecta a redundância existente entre os mesmos. A avaliação é realizada através de cadeias de Markov Contínuas que resultam dos diferentes modos de avaria dos nós da rede. São definidas medidas de capacidade de desempenho (*performability*) e de confiança no funcionamento. Embora o trabalho seja simples e focado em aspectos muito particulares, permite avaliar a capacidade de modelação das cadeias de Markov para obter modelos de avaliação que descrevem o comportamento de uma rede industrial.

A arquitectura tolerante a falhas DELTA-4 é analisada em [KP91]. São consideradas falhas permanentes que afectam a interligação dos nós à rede (NAC – *Network Attachment Controllers*) e cuja avaria os impede de comunicar. Isto implica uma reconfiguração da rede por forma manter um certo nível de redundância entre nós. São analisadas topologias em *bus*²¹ e em anel, ambas com um meio físico de transmissão simples ou redundante. A avaliação é realizada através de cadeias de Markov Contínuas. A partir destes modelos são também obtidas expressões analíticas das medidas de interesse, tais como a fiabilidade e a disponibilidade da rede. Os objectivos e a metodologia utilizados neste trabalho estão muito próximos dos que se pretendem utilizar no trabalho apresentado neste capítulo (embora obviamente o contexto seja diferente).

Em [CTR99] é realizada uma análise do comportamento de uma rede campo quando sujeita a falhas transitórias durante as comunicações. Os dados são transmitidos sob a forma de mensagens, que são difundidas periodicamente. São consideradas falhas transitórias com duração não nula, e o comportamento da rede é analisado por intermédio de uma cadeia de Markov Discreta. Embora a abordagem seja interessante, o trabalho contém um conjunto de erros formais que invalidam muitos dos pressupostos assumidos.

²¹ Optou-se por utilizar a terminologia em Inglês por se julgar mais adequada ao contexto.

[PS02] [Pim03] É proposto um modelo baseado em redes de Petri Estocásticas, para avaliar o comportamento de várias arquitecturas baseada em redes de campo, em que os nós estão agrupados em FTUs (*Fault Tolerant Units*). As falhas consideradas são apenas do tipo permanente e afectando unicamente os nós (que desta perspectiva são considerados atómicos). As medidas obtidas estão centradas na fiabilidade dos FTUs. De forma estranha, tendo em conta os pressupostos assumidos, é utilizada simulação par obter a solução do modelo. Os modelos propostos são muito básicos e em alguns pontos mal formalizados.

3.3 Metodologia Proposta

Face os objectivos propostos e aos trabalhos previamente desenvolvidos, justifica-se a adopção de uma nova abordagem que permita realizar uma avaliação preliminar do comportamento das redes de campo em cenários de falhas permanentes e transitórias, nos contextos previamente referidos.

Neste sentido, foi estabelecida uma metodologia prévia para o desenvolvimento dos modelos com o objectivo de definir o contexto e os pressupostos gerais que lhes estão subjacentes, bem como os requisitos mínimos que estes devem respeitar. A metodologia utilizada resultou da conjugação entre os objectivos definidos e a análise prévia das várias condicionantes ao desenvolvimento dos modelos, quer ao nível do tipo de soluções possíveis, quer da informação associada aos parâmetros dos mesmos. A forma como esta metodologia foi definida pode ser sistematizada nos seguintes aspectos:

- ♦ **Estrutura de rede**

A rede é constituída por um conjunto de nós interligados por intermédio de uma infra-estrutura de comunicações (Fig. 3.1).

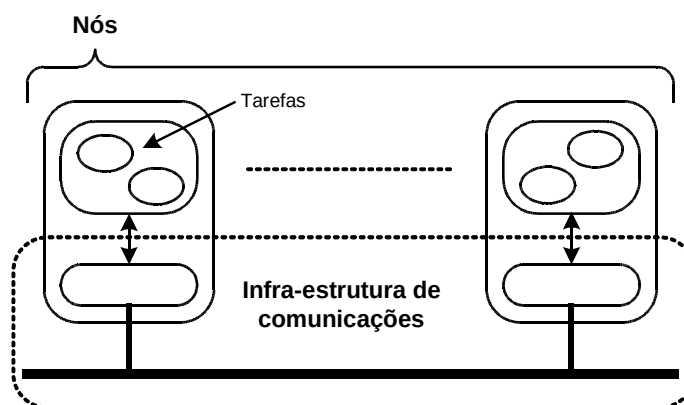


Figura 3.1 - Estrutura da rede.

Cada nó é constituído por vários componentes de *hardware* e *software*, que asseguram a execução de várias tarefas (*tasks*). O conjunto das tarefas que são executadas nos vários nós, representam a aplicação distribuída que controla o processo físico. Estas tarefas trocam mensagens entre si utilizando a infra-estrutura de comunicações existente.

A infra-estrutura de comunicações é constituída pelo meio físico de transmissão (*bus*) e por um conjunto de componentes de *hardware* e *software* locais aos nós. Estes componentes são por exemplo: conectores, processadores de comunicações, *stack*²² de comunicações, etc. Assim, existem componentes que fisicamente pertencem ao nó, mas que do ponto de vista funcional estão incluídos na infra-estrutura de comunicações.

Desta forma, o termo *nó* ao longo desta dissertação deverá ser interpretado de duas formas de acordo com o seguinte contexto. Nas discussões onde a rede é abordada de forma geral, o termo deve ser interpretado como o nó físico, no sentido lato do termo. Quando as discussões versarem sobre a ocorrência e os efeitos das falhas, deverá ser assumida a interpretação definida nesta secção²³.

A decomposição estrutural proposta enquadra-se numa abstracção de alto nível com o objectivo de identificar os componentes mais importantes da rede, na perspectiva da ocorrência de falhas e face às limitações existentes ao nível dos dados disponíveis. Estas limitações resultam da ausência de um conhecimento detalhado sobre o funcionamento da rede, e da escassez da informações sobre a ocorrência de falhas nos componentes. Esta abstracção reflecte assim uma abordagem que procura, ao nível do modelo de avaliação, representar os comportamentos dos quais resultem os maiores níveis de impacto sobre a operação da rede;

♦ Falhas permanentes

Num contexto de ausência de informações detalhadas sobre o funcionamento da rede, não é possível desenvolver modelos com um elevado nível de pormenor. É assim necessário definir um nível de abstracção adequado para o qual exista informação disponível e que simultaneamente permita descrever com um mínimo de rigor o comportamento da rede. Além disso e tendo em conta que um dos principais objectivos é permitir a uma avaliação comparativa entre diferentes arquitecturas, é possível identificar dois grupos de componentes sobre os quais os efeitos das falhas permanentes vão causar o maior impacto no funcionamento da rede: os nós e a infra-estrutura de comunicações.

²² Optou-se por utilizar a terminologia em Inglês por se julgar mais adequada ao contexto.

²³ A adopção de uma terminologia única resultou do facto de ser difícil encontrar um termo em Português para cada interpretação. Em Inglês utilizam-se por vezes os termos *node* e *computational device* para representar ambos. Embora fosse possível adoptar estes termos, o segundo termo não surge de forma natural na linguagem corrente.

Um falha permanente neste tipo de componentes vai implicar a respectiva avaria. Tendo em conta a persistência temporal da falha, a reposição do serviço, da rede ou do componente, só pode ser efectuada após uma reparação para a qual é necessária uma intervenção externa. Fica assim patente que as respectivas consequências dão origem a um impacto importante ao nível do controlo do processo. Estas passam, entre outras, pela indisponibilidade da aplicação de controlo executar as funções para as quais foi desenvolvida.

As avarias dos nós vão afectar de forma directa a estrutura funcional da aplicação de controlo distribuída, porque resultam na ausência de execução das respectivas tarefas. No sentido de tolerar estas situações, a aplicação terá que, eventualmente, dispor de mecanismos de tolerância a falhas que lhe permita lidar com este tipo de situações. Estes mecanismos são específicos da aplicação e passam tipicamente pela existência de nós redundantes e por serviços que permitam a reconfiguração da aplicação (rede) nestas situações.

As avarias na infra-estrutura de comunicações vão impedir a troca de mensagens entre as várias tarefas que são executadas nos nós, afectando assim de forma indirecta o funcionamento da aplicação distribuída. O próprio conceito de aplicação distribuída assenta no facto que a sua execução é profundamente dependente da troca de dados entre as várias componentes (tarefas) desta aplicação. Assim, qualquer impedimento à troca de dados compromete fortemente a execução da aplicação. Tal como no caso anterior, a aplicação pode possuir mecanismos que lhe permitam lidar com estes tipos de situações;

♦ Falhas transitórias

As redes de campo podem estar sujeitas a vários tipos de falhas transitórias. Entre estas, as mais comuns são as que estão associadas às perturbações electromagnéticas (EMI – *Electromagnetic Interference*) que ocorrem no seu ambiente envolvente. Estas falhas podem ter várias origens, tais como: contactos intermitentes, ligações físicas mal realizadas, sistemas electrónicos, motores, descargas atmosféricas, etc. Os seus efeitos são de vários tipos e podem afectar diferentes componentes da rede, sendo contudo os mais frequentes os que dão origem a erros durante a transmissão de mensagens entre as várias tarefas da aplicação distribuída. Isto normalmente resulta do facto de os componentes associados à infra-estrutura de comunicações serem aqueles que estão mais expostos a falhas com estas características, particularmente o meio físico de transmissão pela sua vulnerabilidade.

A esmagadora maioria dos sistemas de controlo suportados por redes de campo possui requisitos de tempo-real associados à transmissão das mensagens associadas ao controlo do processo [Tho98]. Por forma a garantir que estes requisitos são respeitados é necessário que o tempo entre o envio de uma mensagem por uma tarefa, a sua transmissão e a respectiva recepção por

outra tarefa num outro nó, seja majorado [TWB95] (Fig. 3.2). É comum encontrar estes requisitos formalizados através da definição de uma *deadline*²⁴ associada à transmissão da mensagem [Kop97]. Assim, a *deadline* de uma mensagem representa um intervalo de tempo que não deve ser excedido para efectuar a respectiva transmissão com sucesso (incluindo o envio e a recepção), caso contrário a dinâmica de controlo do processo pode ser seriamente perturbada [SK92] [KS94]. Destas perturbações podem resultar situações de avaria da rede que impliquem consequências graves para o processo, ou para as pessoas e o ambiente. A ocorrência de falhas transitórias neste contexto, traduz-se geralmente por atrasos e perdas na transmissão das mensagens, o que por sua vez podem levar à violação das respectivas *deadlines* com as consequências atrás descritas.

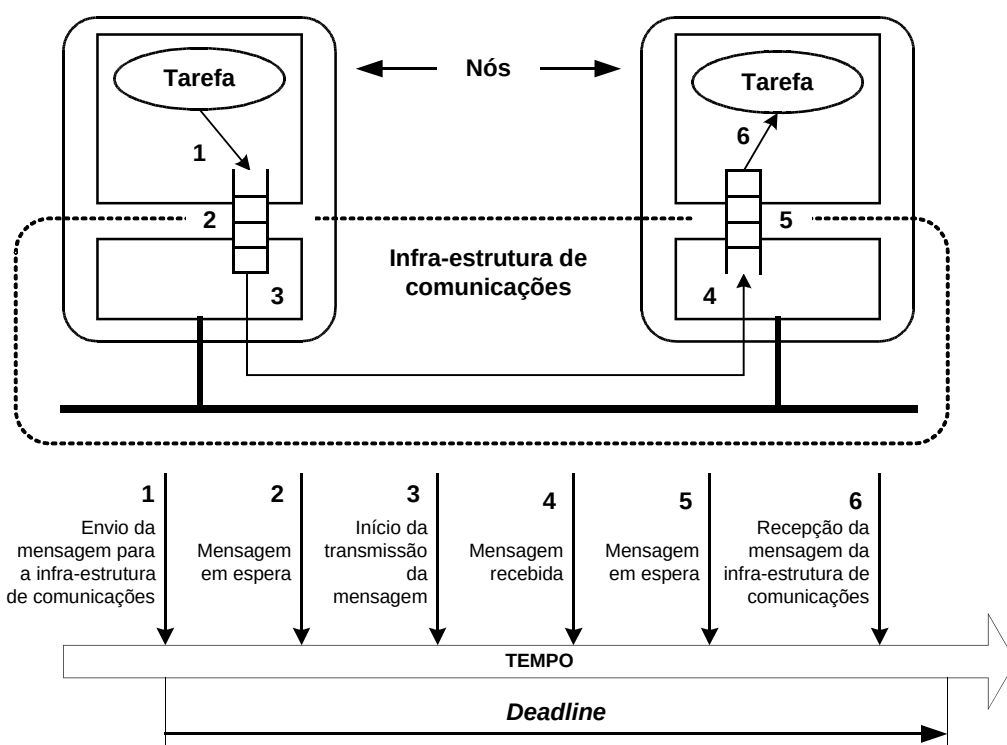


Figura 3.2 - Noção de *deadline* associada à transmissão de uma mensagem.

As condições que caracterizam os efeitos das falhas transitórias durante as comunicações, estão intimamente relacionadas com as características da aplicação distribuída e com a *stack* de comunicações empregue. A grande diversidade entre as *stacks* de comunicação utilizadas pelas redes de campo, impede o desenvolvimento de um modelo único e capaz de ser utilizado para avaliar o comportamento de qualquer tipo de rede. Desta forma, um modelo desenvolvido neste contexto terá que ser necessariamente detalhado para manter um nível elevado de rigor. Exemplos destes tipos de modelos são os

²⁴ Optou-se por utilizar a terminologia em Inglês por se julgar mais adequada ao contexto.

que são propostos para as redes Profibus-DP e CAN, e que serão apresentados e discutidos nos capítulos 5 e 6 respectivamente.

Neste contexto a situação é de um aparente impasse. A ausência de informações detalhadas sobre o funcionamento da rede não permite o desenvolvimento de um modelo detalhado. No entanto um modelo com estas características é necessário para uma avaliação rigorosa.

Este impasse pode ser resolvido adoptando uma atitude pragmática em relação aos objectivos definidos, nomeadamente em relação ao interesse em dispor de uma avaliação comparativa e preliminar. Nesta perspectiva, o modelo não tem que ser absolutamente rigoroso, mas tem que dispor de um nível de rigor adequado que permita confiar nas conclusões dele obtidas a partir de uma avaliação comparativa. Dito de outra forma, as medidas obtidas de um modelo podem não ser absolutamente rigorosas, contudo quando comparadas com outras medidas obtidas no mesmo contexto, a avaliação comparativa “minimiza” de certa forma esses mesmos erros, porque estes resultam do mesmo tipo de aproximações realizadas. Desta forma, os pressupostos assumidos no desenvolvimento do modelo podem ser relaxados no sentido da sua simplificação. Coloca-se contudo a questão de definir qual o nível de abstracção que é necessário adoptar para obter um modelo com estas características.

De acordo com o modelo de referência OSI, a *stack* de comunicações de uma rede de campo está estruturada numa arquitectura com três camadas: física, ligação de dados e aplicação. A camada de ligação de dados (DLL – *Data Link Layer*) representa a parte mais crítica desta arquitectura, dado que o respectivo protocolo comunicações têm que garantir requisitos de tempo-real [Tho98]. A sua importância é de tal ordem, que a esmagadora maioria dos trabalhos sobre o escalonamento de mensagens em redes de campo focam-se exclusivamente nos protocolos definidos nesta camada (ver capítulos 5 e 6).

Surge assim de forma perfeitamente razoável que o nível de abstracção para o desenvolvimento deste tipo de modelos se situe nesta camada. Embora esta abordagem implique que os modelos não sejam totalmente rigorosos, se o comportamento desta camada for adequadamente modelado os modelos obtidos estarão próximos da forma como a rede opera em aplicações reais.

No entanto, existe ainda a questão relacionada com a existência de protocolos de comunicações muito distintos ao nível desta camada. Se se adoptar uma abordagem mais centrada na forma como as mensagens são de facto transmitidas, é possível abstrair (e simplificar) muitas das particularidades dos protocolos, sem contudo deixar de obter um bom nível de rigor. Este aspecto será abordado com maior detalhe durante a discussão dos modelos propostos;

♦ Características dos modelos

Por forma obter modelos que não exijam recursos computacionais significativos para a respectiva solução, procurou-se obter modelos dos quais resultem processos estocásticos Markovianos. Embora este facto imponha necessariamente um conjunto de restrições ao nível dos pressupostos assumidos, em muitos dos casos estes são totalmente compatíveis com a realidade, nomeadamente nos relacionados com a ocorrência das falhas com origem física. Tal como discutido em [KP91] [Bri94] [Bri95], embora o modelo resulte de uma aproximação em relação ao comportamento real, o contexto em que avaliação é realizada garante que os resultados obtidos são consistentes. Além disso, é comum encarar estes resultados como uma aproximação de primeira ordem, dado que em termos de comportamento estocástico o primeiro momento (valor esperado) determina a ordem de grandeza dos resultados²⁵, e os restantes momentos resultam em refinamentos dos mesmos.

Dentro deste mesmo objectivo, seria de todo interessante a possibilidade de dispor ou de extrair expressões analíticas a partir dos modelos desenvolvidos. Com isto garante-se que a avaliação se possa realizar de forma eficiente, quer através da redução a complexidade computacional do modelo, quer pela ausência da necessidade de utilizar ferramentas de modelação especializadas. Trabalhos realizados em contextos diferentes, mas com objectivos semelhantes, [Bri94] [Bri95] demonstram a importância desta abordagem;

♦ Tipo de solução e medidas pretendidas

Pretende-se que as soluções dos modelos sejam obtidas através de técnicas analíticas/numéricas, dado todo o conjunto de vantagens, nomeadamente computacionais, disponibilizadas por este tipo de solução.

A comparação entre as diferentes arquitecturas pode ser realizada através da definição de medidas associadas à fiabilidade e disponibilidade da rede.

3.4 Modelo de Falhas Permanentes

As secções seguintes descrevem a aplicação da metodologia previamente definida para a obtenção dos modelos de falhas permanentes.

²⁵ Como o modelo é Markoviano a distribuição exponencial está subjacente ao mesmo. Esta distribuição pode ser encarada como correspondendo ao conhecimento do primeiro momento.

A utilização da hipótese Markoviana para o desenvolvimento dos modelos, combinada com o interesse em extrair dos mesmos expressões analíticas referentes às medidas de interesse, obriga à adopção de um formalismo de modelação que seja capaz de suportar ambos os aspectos. Embora existam vários formalismos que suportam processos Markovianos (ex. redes de Petri Estocásticas), a obtenção de expressões analíticas directamente dos mesmos não é geralmente suportada devido à complexidade e à dificuldade do seu cálculo. As cadeias de Markov contínuas são uma excepção neste panorama, tendo sido alvo de vários estudos que focam precisamente este último aspecto [PG80] [Bri94] [Bri95]. O facto do seu comportamento em regime estacionário poder ser descrito de forma relativamente simples (um sistema de equações lineares), potenciou o desenvolvimento de um conjunto de métodos algébricos neste contexto que permitem descrever o seu comportamento, geralmente de forma aproximada mas com um bom nível de rigor.

Desta forma e como resultado da necessidade de combinar os aspectos previamente referidos, as cadeias de Markov contínuas foram adoptadas como o formalismo de suporte ao desenvolvimento dos modelos propostos nesta secção.

Neste sentido, inicialmente é apresentado o método dos estados de funcionamento crítico, que permite obter expressões analíticas de modelos baseados em cadeias de Markov contínuas. Posteriormente, são discutidos os pressupostos que serviram de base para a construção dos modelos, sendo estes concretizados através da análise de duas arquitecturas de base consideradas típicas. Por fim é apresentado um caso de estudo com o objectivo de validar a metodologia proposta.

3.4.1 Método dos Estados de Funcionamento Crítico

Considere-se um sistema com um espaço de estados discreto, definido por um comportamento Markoviano homogéneo e do qual se pretendem obter medidas associadas à confiança no funcionamento, nomeadamente a fiabilidade e a disponibilidade. Sem perda de generalidade, admita-se que o espaço de estados M_S do sistema pode ser decomposto em dois conjuntos disjuntos com as seguintes características:

- ♦ N estados (M_N) que representam o funcionamento normal;
- ♦ Um (1) estado (M_A) que representa a situação de avaria do sistema;
- ♦ A evolução entre estados M_N é governada pelos processos de falhas e *reparações*²⁶ dos componentes do sistema;
- ♦ Após uma avaria o sistema mantém-se nesse estado (M_A).

²⁶ Utiliza-se o termo em itálico porque em termos reais pode não ser exactamente uma reparação.

Neste cenário, o comportamento do sistema pode ser modelado por intermédio de uma cadeia de Markov contínua com $N+1$ estados, em que o estado de avaria do sistema é considerado absorvente. As medidas pretendidas podem ser obtidas através da aplicação das técnicas tradicionais de resolução de cadeias de Markov Contínuas [Dyr89] [Kao97] [Triv02].

Contudo, o conjunto destas técnicas permite apenas obter as soluções numéricas do problema. Para obter expressões analíticas é necessário recorrer a outros formalismos, tais como transformadas de Laplace ou à utilização de *software* de computação simbólica, tais como o Maple V [Map04] e o Mathematica [Math04].

A utilização da transformada de Laplace, recorre à transformada inversa para obter a solução no domínio do tempo, sendo apenas exequível para sistemas com uma dimensão reduzida. A utilização de ferramentas de *software* de computação simbólica resulta normalmente na produção de soluções de dimensão muito extensa, dado que estas não discriminam a importância relativa dos vários factores em causa.

O método dos *estados de funcionamento crítico* [PG80], aparece como uma possível solução para os problemas acima indicados, dado que permite extrair de cadeias de Markov Contínuas um conjunto de expressões analíticas de reduzida dimensão e complexidade. Embora estas resultem de um conjunto de aproximações ao funcionamento do sistema, verifica-se experimentalmente (através da garantia de um conjunto de condições) que o erro de aproximação cometido é mínimo.

No campo de aplicações deste método, saliente-se, entre vários, a sua utilização na avaliação da arquitectura de comunicações tolerante a falhas Delta-4 [KP91]. Embora comparativamente sejam diferentes, o sistema que é proposto avaliar neste trabalho possui um conjunto de elementos estruturais semelhantes, o que permite antever que o método apresentado possa ser aplicado de forma eficiente.

3.4.1.1 Descrição das Condições de Aplicação

No caso de um sistema com as características indicadas na secção anterior, demonstra-se que se este for considerado muito fiável, o respectivo processo de absorção tem um comportamento assintótico ($t \rightarrow \infty$) descrito por um processo de Poisson homogéneo [PG80].

Este comportamento pode ser justificado a partir de um raciocínio simples. Considere-se um estado i do sistema: cada vez que o sistema deixa esse estado, há uma probabilidade q de retornar a esse mesmo estado (sem passar pelo estado de avaria), e uma probabilidade $p=1-q$ do sistema avariar, ou seja, em termos de modelo de ser absorvido no estado de avaria. Como se trata de um sistema Markoviano homogéneo, a probabilidade p não depende do tempo. Se se repetir

esta experiência m vezes, o número de vezes em que se verificará uma absorção é definido por uma distribuição Binomial com parâmetros (m, p) . Ora se p for muito pequeno, ou seja se o sistema for muito fiável, a distribuição Binomial pode ser aproximada por uma distribuição de Poisson. Esta aproximação é tanto melhor, quanto mais pequena for a probabilidade p .

Formalmente e em termos do modelo, a aproximação anterior é válida quando se verificam os seguintes pressupostos:

- ♦ A cadeia de Markov constituída pelos estados de funcionamento normal M_N constitui uma grafo fortemente conexo (Fig. 3.3);

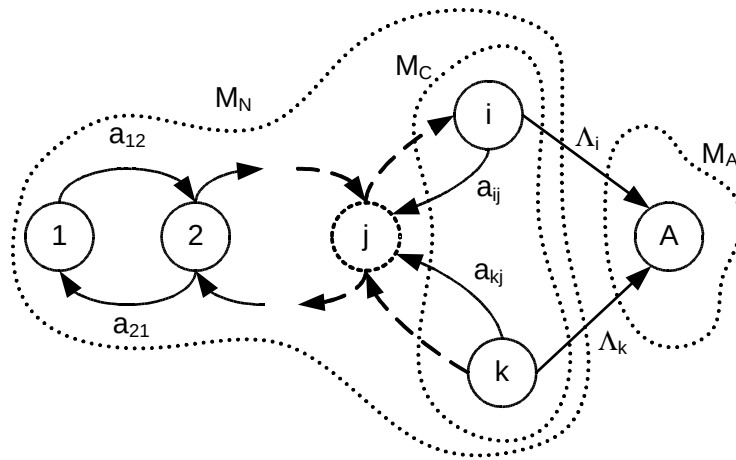


Figura 3.3 - Aplicação do método dos estados de funcionamento críticos: sistemas não reparáveis.

- ♦ Os estados que possuem pelo menos uma transição para o estado de avaria M_A são denominados *estados de funcionamento crítico*, $M_C \subset M_N$. A taxa de transição de um estado crítico i para o estado de avaria definida como Λ_i , deve respeitar a seguinte condição:

$$\frac{\Lambda_i}{\sum_{j \neq i \in M_N} a_{ij}} \ll 1 \quad \forall i \in M_C \quad (3.1)$$

em que a_{ij} representa a taxa de transição do estado i para o estado j . Esta condição reflecte o pressuposto de que o sistema é considerado muito fiável, dado que as taxas de falha críticas Λ_i são muito inferiores às taxas de reparação $\sum_{j \neq i \in M_N} a_{ij}$. A qualidade desta aproximação está intimamente relacionada com o grau de satisfação desta condição.

Considerando-se satisfeitas ambas as condições anteriores, a taxa do processo de Poisson equivalente λ_e obtêm-se a partir da cadeia de Markov original através da aplicação da seguinte expressão

$$\lambda_e = \sum_{\text{percursos entre os estados inicial e de avaria}} \left\{ \frac{\prod \text{taxas do percurso}}{\prod_{\text{estados do percurso} \notin (\text{inicial}, \text{avaria})} (\sum \text{taxas de saída do estado})} \right\} \quad (3.2)$$

em que o estado inicial²⁷ representa a situação em que todos os componentes dos sistema estão operacionais.

Esta aproximação pode ser interpretada como uma transformação da cadeia de Markov original numa cadeia reduzida (Fig. 3.4), constituída apenas por dois estados: um estado de funcionamento normal (o estado inicial, **1**) e um estado de avaria (**A**). A taxa de transição assintótica entre ambos λ_e pode ser considerada como a taxa equivalente de avarias do sistema.

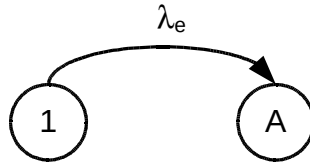


Figura 3.4 -Cadeia de Markov reduzida: sistemas não reparáveis

A fiabilidade do sistema é dada por

$$R(t) \approx \exp(-\lambda_e \cdot t) \quad (3.3)$$

O método anterior pode também ser aplicado a sistemas reparáveis. Para além dos pressupostos anteriores, é assumido neste caso que posteriormente a uma avaria do sistema, este é reparado e colocado a funcionar num estado operacional. As condições exigidas neste caso são semelhantes às anteriores e estão relacionadas com o comportamento do sistema em regime estacionário (Fig. 3.5).

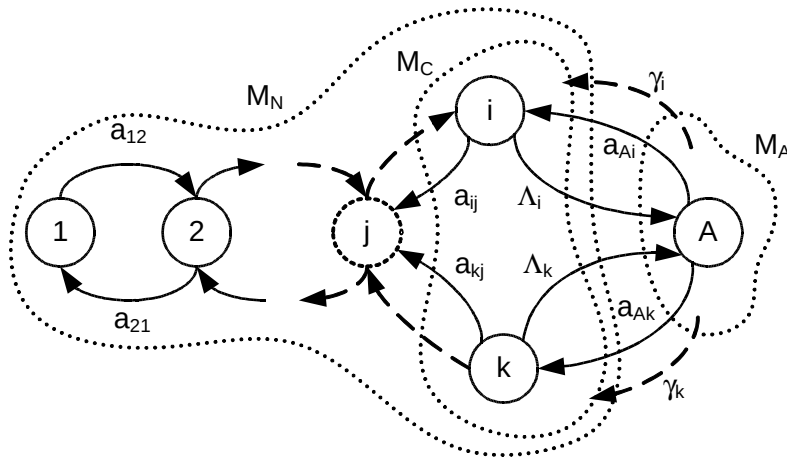


Figura 3.5 - Aplicação do método dos estados de funcionamento críticos: sistemas reparáveis.

²⁷ Representado nas figuras pelo valor **1**.

$$\begin{aligned}
 \gamma_i &= \sum_{j \in M_c} a_{ij} \cdot \frac{Prob_j(\infty)}{\sum_{j \in M} a_{ij} Prob_j(\infty)} \\
 \gamma_c &= - \sum_{i=1}^n \gamma_i \\
 \sum_{i \in M_c} \left(\frac{\gamma_i}{\gamma_c} \right) \cdot \frac{\Lambda_i}{\sum_{j \neq i \in M_N} a_{ij}} &\ll 1 \quad \forall i \in M_c
 \end{aligned} \tag{3.4}$$

em que $Prob_j(\infty)$ representa a probabilidade em regime estacionário do sistema ocupar o estado j . As taxas γ_i representam o equivalente estacionário das taxas de *reparação* do sistema entre o estado de avaria M_A e os estados de funcionamento crítico M_c . A equação (3.4) mostra que as condições para uma boa aproximação são semelhantes às apresentadas na equação (3.1).

Como se trata de um sistema reparável, a indisponibilidade assintótica é obtida como

$$\bar{A}(\infty) = \lim_{t \rightarrow \infty} 1 - A(t) = \sum_{\text{percursos entre os estados iniciais e de falha}} \left\{ \frac{\prod \text{taxas do percurso}}{\prod_{\substack{\text{estados do percurso} \\ \notin \text{iniciais}}} (\sum \text{taxas de saída do estado})} \right\} \tag{3.5}$$

Esta aproximação pode ser interpretada de forma análoga à realizada para os sistemas não-reparáveis. A cadeia de Markov original é transformada numa cadeia reduzida, constituída apenas por dois estados: um estado de funcionamento normal (1) e um estado de avaria (A) (Fig. 3.6).

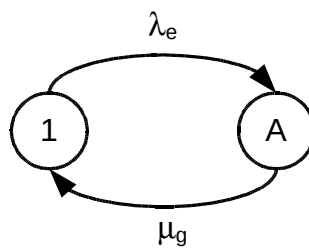


Figura 3.6 - Cadeia de Markov reduzida: sistemas reparáveis.

Neste caso μ_g representa a taxa de reparação global do sistema, ou seja o mecanismo pela qual após uma avaria o sistema é reposto num estado onde todos os seus componentes estão operacionais. A indisponibilidade assintótica do sistema é dada por

$$\bar{A} = \frac{\lambda_e}{\mu_g} \tag{3.6}$$

3.4.2 Pressupostos Assumidos

Partindo da metodologia previamente definida, estabeleceram-se os seguintes pressupostos no desenvolvimento dos modelos:

- ♦ A rede é constituída por dois tipos de componentes: n nós com características, em termos de falhas, semelhantes e por uma infra-estrutura de comunicações.

A infra-estrutura de comunicações é constituída pelo meio físico de transmissão, pelos *transceivers*, por eventuais processadores de comunicações e pelos conectores mecânicos;

- ♦ As falhas²⁸ nos componentes da rede ocorrem de acordo com um processo de Poisson (taxa de ocorrência constante) [Triv02] e têm origem no respectivo *hardware*. Este pressuposto é baseado na abordagem clássica da ocorrência das falhas em componentes de eléctricos ou electrónicos [MIL91];
- ♦ Da perspectiva da ocorrência das falhas, os nós e infra-estrutura de comunicações são considerados atômicos, com taxas de avarias λ_n e λ_b respectivamente. Estas taxas têm em conta a contribuição de todos os seus componentes constituintes. A ausência de decomposição interna destes componentes resulta do contexto em que esta avaliação é realizada, nomeadamente devido às limitações existentes ao nível dos dados disponíveis sobre o comportamento da rede;
- ♦ Uma avaria nos componentes da infra-estrutura de comunicações que estão fisicamente nos nós, dependendo das suas características, pode contribuir para as falhas do nó, para falhas da infra-estrutura de comunicações, ou para ambas. Como este é um aspecto complexo, optou-se por separar (probabilisticamente) as respectivas taxas de avaria em locais e globais. A primeira é incorporada na taxa de avaria dos nós, e a segunda na taxa de avaria da infra-estrutura de comunicações. Esta abordagem resulta de uma tentativa de separar os efeitos das falhas, sendo perfeitamente razoável em termos de hipótese. Desta forma, assume-se que as taxas λ_n e λ_b já reflectem este aspecto.

No sentido de facilitar a descrição dos modelos e tendo em conta o contexto da discussão, a infra-estrutura de comunicações é denominada deste ponto em diante como *meio de transmissão*;

²⁸ De acordo com a terminologia adoptada no capítulo 2, as avarias de um componente resultam numa falha para o sistema que o contém. Por forma a utilizar um vocabulário que não induza a ambiguidades adoptou-se a seguinte interpretação: uma avaria de um componente do sistema é referida como uma *falha* no sistema. Desta forma, utiliza-se o termo *avaría* apenas ao nível do sistema (*system failure*). Note-se que se está apenas perante uma questão de perspectiva. Esta terminologia será adoptada ao longo de toda a dissertação.

- ♦ Quando ocorre uma falha num nó ou no meio de transmissão, é considerado que isto implica a avaria permanente do mesmo. A reposição do serviço é assegurada por intermédio de uma reparação ou substituição. Esta reparação ocorre de acordo com uma distribuição exponencial com taxa μ . Este pressuposto é imposto pela necessidade de se obter um modelo Markoviano. É importante notar que em termos reais pode não constituir uma boa aproximação da realidade. Assume-se também que é reparado um componente de cada vez (esta restrição pode ser facilmente relaxada se necessário);
- ♦ A existência de mecanismos de tolerância a falhas permite (dentro de certas restrições) que após a avaria de um componente a rede possa ser reconfigurada num intervalo de tempo desprezável, continuando a fornecer o serviço respectivo. A eficácia destes mecanismos é modelada sob a forma de uma probabilidade, denominada **factor de cobertura** (*coverage factor*) [BCS69] [DT89] [ACC+93] [PMA+95] [PCA+96] [Hil00] [Con03].

Formalmente o factor de cobertura c é definido como

$$c = \text{Prob}[\text{Sistema continuar a operar} | \text{Ocorreu uma falha}] \quad (3.7)$$

Dito de outra forma, a probabilidade da ocorrência de uma falha ao nível de um componente não impedir o funcionamento normal da rede é c .

Como os mecanismos de tolerância a falhas actuam em escalas temporais muito inferiores às associadas à ocorrência de falhas nos componentes [PCA+96] [KS00], o seu comportamento estocástico pode ser condensado directamente no factor de cobertura [DT89]. Este aspecto será abordado com maior nível detalhe no capítulo 4.

Função da propriedade de decomposição do processo de Poisson [Triv02] (ver Capítulo 2), este factor pode ser facilmente incluído no modelo como o produto da taxa de avarias pelo factor de cobertura (Fig. 3.7);

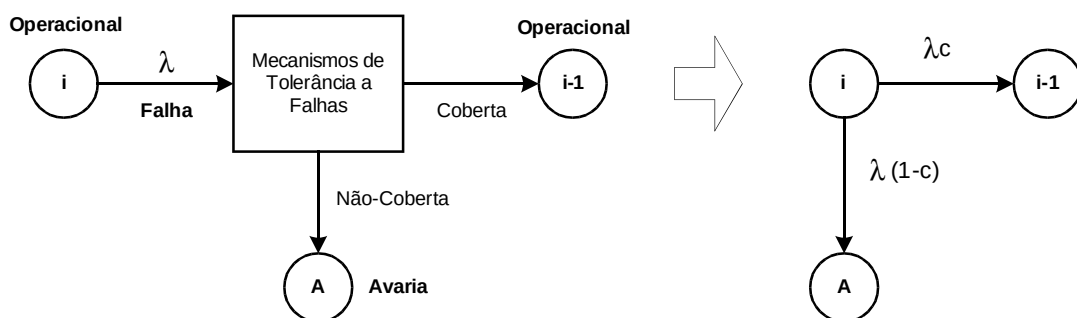


Figura 3.7- Factor de cobertura.

- ♦ Entende-se por falha coberta (Fig. 3.7), uma avaria de um componente que não impede o funcionamento normal da rede. Ou seja, os mecanismos de tolerância a falhas existentes funcionaram forma adequada, garantindo a

continuidade do serviço fornecido. Assim, quando ocorre uma avaria deste tipo, a reparação e/ou substituição e/ou isolamento do componente afectado não altera o funcionamento normal da rede. Neste sentido também se assume que a reparação têm sempre sucesso, e que a reintegração do componente se pode realizar sem interromper o funcionamento da rede²⁹;

- ♦ A ocorrência de uma falha não-coberta (Fig. 3.7) implica a avaria completa da rede. Posteriormente a esta situação, todos os componentes avariados são reparados antes da rede reiniciar o seu funcionamento normal. Esta reparação ocorre de acordo com uma distribuição exponencial com taxa μ_g ;
- ♦ Não se assumem (para já) quaisquer distinções referentes à gravidade dos modos de avaria da rede.

3.4.3 Arquitecturas Avaliadas

O desenvolvimento de um modelo tem como ponto de partida a existência de uma arquitectura da rede que se pretende avaliar. Como existem muitas possibilidades, não é possível propor um único modelo capaz de incorporar todas as situações.

Desta forma, optou-se por definir duas arquitecturas genéricas, e de certa forma ortogonais, mas suficientemente representativas de situações reais e que possam ser facilmente adaptadas a casos concretos.

Estas arquitecturas encontram-se representadas através dos seguintes modelos:

- ♦ **Modelo m nós Redundantes (mR).** A rede suporta até m falhas cobertas consecutivas referente a avarias dos nós, ou seja o número mínimo de nós com que a rede pode funcionar é $n-m$. Este modelo foi desenvolvido tendo em conta a possibilidade de existir de redundância de funcionamento entre vários nós da rede;
- ♦ **Modelo m nós Dispensáveis (mD).** Dos n nós existentes, m são considerados dispensáveis, e $n-m$ são considerados essenciais para o funcionamento da rede. Assim, uma avaria de um nó dispensável, desde que corresponda a uma falha coberta, não afecta o funcionamento da rede. Uma avaria de um nó essencial, implica a avaria da rede. Este modelo foi desenvolvido tendo em conta a possibilidade da rede funcionar num modo degradado e assim fornecer um serviço mínimo.

²⁹ A reintegração nem sempre pode ser realizada desta forma, ou seja existem situações em que a operação da rede tem que ser suspensa. Contudo e face às restantes grandezas temporais do modelo, o respectivo tempo é várias ordens de grandeza inferior, pelo que se justifica esta abordagem. Saliente-se contudo, este aspecto pode ser facilmente incluído nos modelos.

Os modos de avaria da rede são assim definidos a partir das condições de operação definidas para cada arquitectura avaliada. Dito de outra forma, quando estas condições não forem asseguradas dá-se uma avaria da rede.

Refira-se que não é intenção dos modelos propostos incorporar a representação de todas as arquitecturas possíveis. Por exemplo, em termos de aplicações concretas não será comum dispor de uma rede que opere de acordo com o modelo **mR**, dado que este modelo implica um funcionamento que exige um nível de redundância que abrange todos os nós da rede. Coloca-se então a questão: “porquê propor estes modelos?”. A resposta pode ser dada observando que os modelos de avaliação associados a arquitecturas tipicamente utilizadas em situações reais podem ser obtidos quer por simplificação dos modelos propostos, quer por composição dos mesmos. Tomemos como exemplo os seguintes casos:

- ♦ Uma rede com n nós dos quais k são de facto constituídos por unidades tolerantes a falhas (FTU – *Fault Tolerant Units*) com r nós cada. Esta arquitectura pode ser representada através da composição k modelos do tipo **mR**;
- ♦ Uma rede com n nós dos quais k são redundantes e m são dispensáveis. Esta arquitectura seria representada pela composição dos modelos **mR** com **mD**.

A opção pela sua utilização justifica-se assim pelo facto de representarem os modelos base das arquitecturas tipicamente utilizadas em situações reais.

Nas secções seguintes são apresentados exemplos de aplicação do método dos estados de funcionamento crítico aos modelos desenvolvidos para as arquitecturas propostas. O objectivo principal é determinar uma expressão para a taxa equivalente de avarias do sistema λ_e , a partir da qual podem ser obtidas as várias medidas de interesse. As análises realizadas partem dos pressupostos definidos previamente, extendendo-os posteriormente através da introdução de meios de físicos de comunicação redundantes e de avarias em modo-comum.

Note-se que independentemente da obtenção da taxa equivalente de avarias, os modelos obtidos são cadeias de Markov Contínuas, pelo que a sua solução pode ser obtida também através das técnicas numéricas descritas no capítulo 2.

3.4.4 Modelo **mR** : m nós redundantes

Segundo esta arquitectura, existe da avaria da rede quando ocorrer uma das seguintes situações:

- ♦ Uma falha não-coberta de um nó qualquer;

Avaliação Preliminar da Confiança no Funcionamento

- ♦ A exaustão da redundância (por falhas cobertas) de um número mínimo de $m+1$ nós;
- ♦ Uma falha do meio de transmissão.

O modelo referente a esta arquitectura é representado pela cadeia de Markov da Figura 3.8. Os estados da cadeia numerados de $0 \dots M$ referem-se aos estados operacionais da rede, e o índice de cada estado indica o respectivo número de nós num estado de avaria. O estado **A** representa a avaria da rede. As expressões nas transições entre estados referem-se às respectivas taxas.

Definiu-se c_n como o factor de cobertura associado às falhas dos nós, e admitiu-se, por simplicidade nos modelos, que este parâmetro não depende do número de nós operacionais.

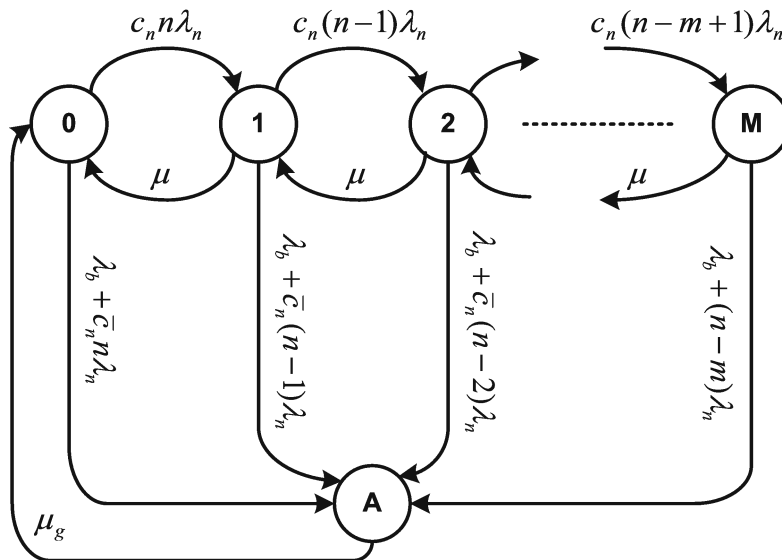


Figura 3.8 - Cadeia de Markov associada ao modelo mR.

Como se trata do primeiro modelo apresentado, proceder-se-á uma descrição mais detalhada da forma como foram obtidas as taxas de transição. Esta descrição será suficiente para perceber os modelos apresentados nas próximas secções.

Considere-se assim o estado **1**. Este estado corresponde à situação em que um (1) nó se encontra avariado. A partir deste estado a rede pode evoluir de 3 formas distintas:

1. Ocorre uma falha coberta de um nó e o sistema transita para o estado **2**, o qual representa a existência de dois nós avariados. Como as falhas ocorrem de acordo com um processo de Poisson, a taxa total de falhas é dada pela soma das taxas de falhas individuais dos nós operacionais $(n-1)\lambda_n$. Isto resulta directamente da sua propriedade aditiva [Triv02]. Além disso, é necessário entrar em conta com o factor de cobertura. Devido à propriedade de

decomposição que este tipo de processos goza, a taxa final é obtida como o produto da taxa total pela factor de cobertura $c_n(n-1)\lambda_n$;

2. Ocorre uma falha do meio de transmissão ou uma falha não coberta de um nó e o sistema transita para o estado **A**. De acordo com discussão anterior a taxa de transição é dada por $\lambda_b + \bar{c}_n(n-1)\lambda_n$, em que $\bar{c}_n = (1 - c_n)$;
3. Ocorre uma reparação do nó avariado e o sistema transita para o estado **0**. A taxa de transição é dada por μ .

Tendo em conta que na esmagadora maioria dos casos $\min\{\mu, \mu_g\} \gg \max\{\lambda_n, \lambda_b\}$, ou seja as taxas de reparação têm ordens de grandeza muito superiores às taxas de avaria, a expressão de λ_e pode ser obtida, com grande aproximação, considerando apenas os termos de primeira ordem³⁰

$$\lambda_e \approx \lambda_b + \bar{c}_n n \lambda_n + \frac{c_n n \lambda_n}{\mu} [\lambda_b + \bar{c}_n(n-1)\lambda_n] \quad (3.8)$$

Dos três termos desta expressão, e de acordo com os pressupostos assumidos previamente, só os dois primeiros são relevantes.

A taxa de falhas do meio transmissão λ_b representa o valor mínimo de λ_e dado que em termos funcionais a rede pode ser vista como uma ligação série do meio de transmissão com os nós.

O segundo termo $\bar{c}_n n \lambda_n$, contribui significativamente para o resultado final dado a gama de valores que pode assumir. Além disso, este termo é fortemente condicionado pelo factor de cobertura c_n . Admitindo que n e λ_n são pré-determinados, a diminuição deste termo só é possível através do aumento do factor de cobertura c_n , ou seja, através da introdução de mecanismos adicionais de tolerância a falhas, ou através da melhoria da eficácia dos mesmos.

3.4.5 Modelo mD : m nós dispensáveis

Segundo esta arquitectura, existe uma avaria da rede quando ocorrer uma das seguintes situações (Fig. 3.9):

- ♦ Uma falha não-coberta de um dos m nós;
- ♦ Uma falha de um dos $n-m$ nós;
- ♦ Uma falha do meio de transmissão.

³⁰ Este princípio será adoptado para todas as expressões deduzidas nestas secções.

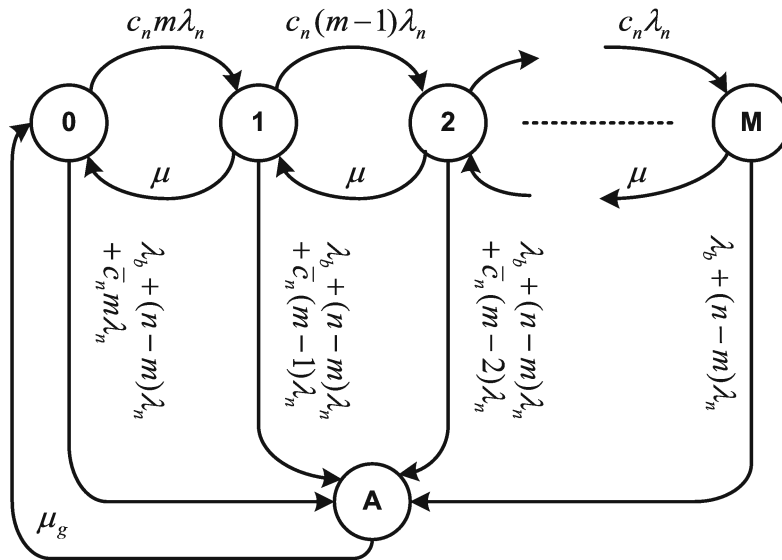


Figura 3.9 - Cadeia de Markov associada ao modelo mD.

A expressão de λ_e é

$$\lambda_e \approx \lambda_b + (n-m)\lambda_n + \bar{c}_n m \lambda_n + \frac{c_n m \lambda_n}{\mu} [\lambda_b + (n-m)\lambda_n + \bar{c}_n (m-1)\lambda_n] \quad (3.9)$$

Dos quatro termos da expressão, e de acordo com os pressupostos assumidos anteriormente, só os três primeiros são relevantes. O primeiro e o terceiro foram já previamente discutidos. O segundo resulta do facto de que uma falha (de qualquer tipo) num dos $(n-m)$ nós implicar a imediata avaria da rede, dado não existir qualquer tipo de redundância a este nível.

Embora os modelos **mR** e **mD** sejam muito semelhantes, é importante ter em conta que o factor de cobertura c_n não é necessariamente o mesmo em ambos. No modelo **mR** os mecanismos de reconfiguração abrangem todos os nós, enquanto que no modelo **mD** abrangem apenas uma pequena parte. Desta forma, a sua eficácia será previsivelmente maior, implicando assim valores mais elevados para o factor de cobertura.

Por uma questão de espaço de apresentação, e sem perda de generalidade, nas secções seguintes as análises vão restringir-se apenas o modelo **mR**.

3.4.6 Avarias em Modo-Comum

Os modelos apresentados até ao momento assumem que as falhas são independentes. Contudo, é possível admitir cenários onde as falhas estejam relacionadas (ex. uma descarga atmosférica que afecta simultaneamente vários componentes de um sistema). Estes tipos de falhas causam erros com

características semelhantes, que por sua vez dão origem a avarias de causa-comum (*common-cause failures*). Pelo tipo de impacto que têm no funcionamento de um sistema, este tipo de avarias assume uma elevada importância, tendo sido alvo de múltiplos trabalhos [CC86] [FT97] [Vau98] [KBJ00] [MSM00] [Zio02] [Xin03]. O trabalho realizado em [Mau00] constitui uma referência onde uma discussão mais profunda deste assunto é realizada.

Uma das questões clássicas deste problema tem a ver com a terminologia utilizada, utilizando-se por vezes o termo avarias em modo-comum (*common-mode failures*). Partindo da terminologia apresentada na Tabela 3.1 [Mau00], a definição de avarias em modo-comum implica que os modos de avaria dos componentes afectados sejam todos iguais, o mesmo não se podendo dizer das avarias de causa-comum.

Avaria Dependente	Tipo de avaria de um sistema cuja probabilidade não pode ser expressa como o produto das probabilidades incondicionais das avarias dos seus componentes.	
	Avaria de Causa-Comum	Tipo de avaria dependente, onde a avaria simultânea dos componentes resulta de uma causa comum.
	Avaria em Modo-Comum	Tipo de avaria de causa-comum, em que os componentes avariaram todos da mesma forma.
	Avaria em Cascata	Todo o tipo de avarias que não são de causa comum.

Tabela 3.1 - Terminologia associada às avarias de causa-comum.

Como as avarias em modo-comum implicam que os todos componentes afectados avariaram da mesma forma, a sua modelação pode realizar-se com maior segurança dado que os modos de avaria são conhecidos. No caso das avarias de causa-comum esta tarefa é mais complexa, pois podem existir múltiplos modos de avaria e é necessário determinar o seu efeito combinado no comportamento global, o que geralmente é complexo. Neste sentido, optou-se nesta dissertação por apenas analisar as consequências das avarias em modo-comum no comportamento dos sistemas.

São vários os modelos propostos para representar este tipo de avarias [Zio02], sendo no entanto mais utilizados os seguintes:

- ♦ **Modelo β .** Segundo este modelo a taxa de avarias de total de um componente é a soma da taxa de avarias normal com a taxa de avarias em modo-comum, sendo esta última definida como uma fracção (β) da taxa total [Gob98]

$$\begin{aligned}
 \lambda_{Total} &= \lambda_{Normal} + \lambda_{Modo-Comum} \\
 \lambda_{Modo-Comum} &= \beta \cdot \lambda_{Total}
 \end{aligned}
 \tag{3.10}$$

O modelo assume implicitamente, que existem apenas dois tipos de processos de avarias: um processo, dito normal, que afecta apenas o componente, e um processo de avarias em modo-comum que afecta todos os componentes. De acordo com [Zio02], os resultados obtidos por este modelo são tipicamente pessimistas;

- ♦ **Modelo Binomial.** De acordo com este modelo, a causa comum das avarias (uma falha) ocorre com taxa λ_c , e existe uma probabilidade p que um componente específico avarie devido a esta falha [Atw86]. A probabilidade de avariarem k em m componentes é dada por uma distribuição binomial

$$\binom{m}{k} p^k (1-p)^{(m-k)} \quad (3.11)$$

Este modelo assume a existência simultânea de vários processos de avaria que afectam grupos de componentes com uma dimensão específica. A taxa de cada processo resulta do produto de λ_c com a probabilidade definida em (3.11).

3.4.7 Modelo mRCS : mR Modo-comum simplificado

O modelo mais simples para analisar os efeitos das avarias em modo-comum, consiste em considerar que existem $m+1$ processos de falhas³¹ que afectam todos os componentes que estão operacionais no momento. Esta abordagem representa assim a aplicação directa do modelo β definido previamente.

Desta forma, está associado a cada conjunto de j componentes (nós e meio de transmissão) um processo de falhas em modo-comum, com uma taxa λ_c^j de ocorrência. Tendo em conta as características deste modelo, não faz sentido assumir a existência de falhas em modo-comum cobertas, dado quando estas ocorrem todos os elementos operacionais ficam num estado de avaria. Trata-se assim de um modelo que produzirá naturalmente resultados pessimistas.

Neste modelo (considerando apenas as falhas em modo-comum), existe uma avaria da rede quando ocorrer uma falha em modo-comum envolvendo os componentes operacionais. Por questões de simplicidade de apresentação a cadeia de Markov terá apenas as transições referentes às falhas em modo-comum (Fig. 3.10).

A expressão de λ_e é

$$\lambda_e \approx \lambda_b + \bar{c}_n n \lambda_n + \lambda_c^{n+1} + \frac{c_n n \lambda_n}{\mu} [\lambda_b + \bar{c}_n (n-1) \lambda_n + \lambda_c^n] \quad (3.12)$$

³¹ Assume-se a interpretação referente às falhas discutida previamente.

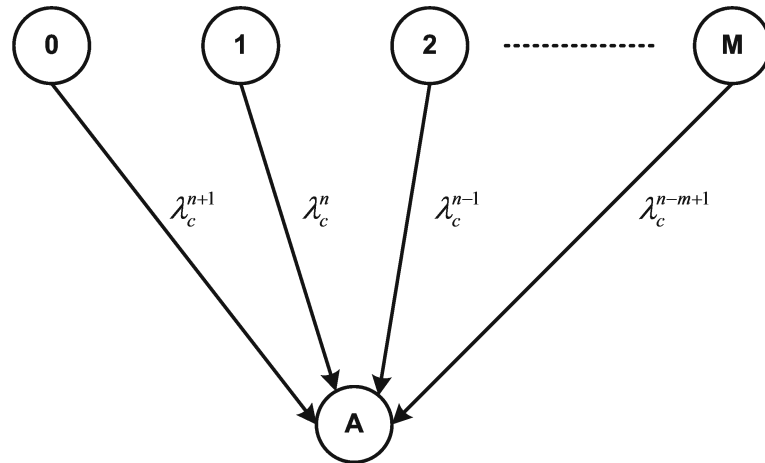


Figura 3.10 - Cadeia de Markov associada ao modelo mRCS.

Analizando apenas os termos associados às falhas em modo-comum, e de acordo com os pressupostos assumidos previamente, verifica-se que o único termo relevante é λ_c^{n+1} que funciona como limitador do valor mínimo de λ_e .

3.4.8 Modelo mRCC : mR Modo-comum complexo

No contexto das avarias em modo-comum, um modelo mais complexo, e também mais realista, considera a existência de vários processos de falhas em modo-comum que são função do número de componentes que avariam simultaneamente. Esta abordagem representa a aplicação do modelo Binomial definido previamente.

Desta forma, a um conjunto de j nós específicos está associado um processo de falhas em modo-comum com uma taxa λ_{cn}^j de ocorrência. De acordo com o modelo Binomial esta taxa pode ser obtida como

$$\lambda_{cn}^j = p^j (1-p)^{(n-j)} \cdot \lambda_c \quad (3.13)$$

Tendo em conta que na rede existem n nós e que os processos de falhas que afectam j nós têm todas as mesmas características, a taxa de ocorrência de todos os processos de falhas em modo-comum deste tipo é

$$\binom{n}{j} \lambda_{cn}^j \quad j \geq 2 \quad (3.14)$$

Além disso, é também necessário considerar que este tipo de falhas afecta também o meio de transmissão. Admitindo que um processo de falhas em modo-comum que afecta simultaneamente j nós e o meio de transmissão possui uma taxa λ_{cnb}^j de ocorrência, então a taxa de ocorrência de todos os processos de falhas em modo comum que envolvem o meio de transmissão é

$$\sum_{j=1}^n \binom{n}{j} \lambda_{cnb}^j \quad (3.15)$$

Ao contrário do modelo anterior, tem toda a lógica assumir que algumas destas falhas (função do número de componentes que avariam) podem ser cobertas. Desta forma e para não tornar o modelo ainda mais complexo, assumiu-se que o factor de cobertura associado às falhas dos nós c_n é considerado independente do número de nós que falham simultaneamente³².

Assim e de acordo com este modelo, a avaria da rede devida a falhas de modo-comum verifica-se quando ocorrer uma das seguintes situações:

- ♦ Uma falha não-coberta de um grupo qualquer de $j \leq n$ nós;
- ♦ Uma falha de um grupo qualquer de $j > m \leq n$ nós;
- ♦ Uma falha de um grupo qualquer que envolva o meio de transmissão.

Por questões de simplicidade de apresentação, a cadeia de Markov (Fig. 3.11) terá apenas as transições referentes às falhas em modo-comum.

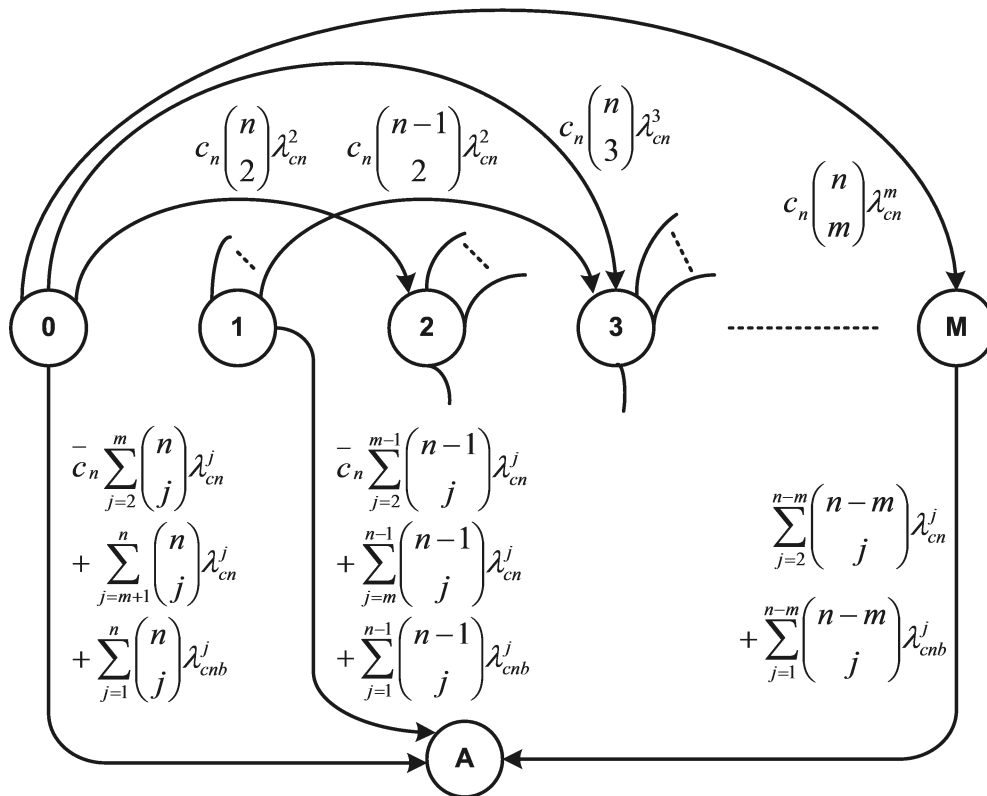


Figura 3.11 - Cadeia de Markov associada ao modelo mRCC.

³² Seria simples introduzir um factor de cobertura função do número j de nós que avariam em simultâneo.

A expressão de λ_e é

$$\begin{aligned}
 \lambda_e \approx & \lambda_b + \bar{c}_n n \lambda_n + \bar{c}_n \sum_{j=2}^n \binom{n}{j} \lambda_{cn}^j + \sum_{j=m+1}^n \binom{n}{j} \lambda_{cn}^j + \sum_{j=1}^n \binom{n}{j} \lambda_{cnb}^j \\
 & + \frac{c_n n \lambda_n}{\mu} \left[\lambda_b + \bar{c}_n (n-1) \lambda_n + \bar{c}_n \sum_{j=2}^{m-1} \binom{n-1}{j} \lambda_{cn}^j \right. \\
 & \left. + \sum_{j=m}^{n-1} \binom{n-1}{j} \lambda_{cn}^j + \sum_{j=1}^{n-1} \binom{n-1}{j} \lambda_{cnb}^j \right] \\
 & + \frac{c_n \binom{n}{2} \lambda_{cn}^2}{\mu} \left[\lambda_b + \bar{c}_n (n-2) \lambda_n + \bar{c}_n \sum_{j=2}^{m-2} \binom{n-2}{j} \lambda_{cn}^j \right. \\
 & \left. + \sum_{j=m}^{n-2} \binom{n-2}{j} \lambda_{cn}^j + \sum_{j=1}^{m-2} \binom{n-2}{j} \lambda_{cnb}^j \right] \\
 & + \frac{c_n \binom{n}{3} \lambda_{cn}^3}{\mu} [\dots] \\
 & + \dots \\
 & + \frac{c_n \binom{n}{m} \lambda_{cn}^m}{\mu} \left[\lambda_b + (n-m) \lambda_n + \sum_{j=2}^{n-m} \binom{n-m}{j} \lambda_{cn}^j + \sum_{j=1}^{n-m} \binom{n-m}{j} \lambda_{cnb}^j \right]
 \end{aligned} \tag{3.16}$$

A avaliação deste modelo resulta necessariamente complexa, dado que é necessário conhecer o valores de todos os λ_{cn}^j e λ_{cnb}^j cuja determinação se torna na prática, difícil de obter (de acordo com o modelo Binomial seria necessário determinar p e λ_c). Contudo, tendo em conta as expressões obtidas e o conhecimento real do funcionamento da rede, é possível realizar algumas simplificações baseadas nos seguintes pressupostos:

- ♦ A evolução de $\binom{n}{j}$ em função de j , segue uma curva de “aspecto” gaussiano;
- ♦ Para valores crescentes de j , λ_{cn}^j e λ_{cnb}^j tomam, com uma probabilidade muito elevada, valores fortemente decrescentes. Este pressuposto baseia-se no facto (observável) de que é muito mais provável falharem simultaneamente j componentes, do que $j+1$.

Assim, partindo dos pressupostos anteriores e assumindo um exemplo concreto, em que as falhas em modo-comum mais relevantes são as que envolvem até 3 componentes, a expressão anterior seria simplificada para

$$\begin{aligned}
 \lambda_e \approx & \lambda_b + \bar{c}_n \left[n \lambda_n + \binom{n}{2} \lambda_{cn}^2 + \binom{n}{3} \lambda_{cn}^3 \right] + n \lambda_{cnb}^1 + \binom{n}{2} \lambda_{cnb}^2 \\
 & + \frac{c_n n \lambda_n}{\mu} \left[\lambda_b + \bar{c}_n \left[(n-1) \lambda_n + \binom{n-1}{2} \lambda_{cn}^2 + \binom{n-1}{3} \lambda_{cn}^3 \right] \right. \\
 & \left. + (n-1) \lambda_{cnb}^1 + \binom{n-1}{2} \lambda_{cnb}^2 \right] \\
 & + \frac{c_n \binom{n}{2} \lambda_{cn}^2}{\mu} \left[\lambda_b + \bar{c}_n \left[(n-2) \lambda_n + \binom{n-2}{2} \lambda_{cn}^2 + \binom{n-2}{3} \lambda_{cn}^3 \right] \right. \\
 & \left. + (n-2) \lambda_{cnb}^1 + \binom{n-2}{2} \lambda_{cnb}^2 \right] \\
 & + \frac{c_n \binom{n}{3} \lambda_{cn}^3}{\mu} \left[\lambda_b + (n-m) \lambda_n + \bar{c}_n \left[\binom{n-m}{2} \lambda_{cn}^2 + \binom{n-m}{3} \lambda_{cn}^3 \right] \right. \\
 & \left. + (n-m) \lambda_{cnb}^1 + \binom{n-m}{2} \lambda_{cnb}^2 \right]
 \end{aligned} \tag{3.17}$$

Analisando apenas os termos associados às falhas em modo comum, e de acordo com os pressupostos assumidos previamente, os termos mais relevantes são apenas os 4 primeiros.

Os dois primeiros assumem um papel determinante, dado que $\binom{n}{2} \lambda_{cn}^2$ e $\binom{n}{3} \lambda_{cn}^3$ podem tomar valores significativos, sendo contudo fortemente condicionados pelo factor de cobertura c_n . Qualquer aumento, mesmo que mínimo, deste factor terá implicações importantes na redução dos efeitos das falhas em modo-comum, e consequentemente melhoria do comportamento da rede.

Os dois últimos termos, $n \lambda_{cnb}^1$ e $\binom{n}{2} \lambda_{cnb}^2$, acabam por ter um efeito muito mais limitado, dado que representam, geralmente, taxas de ocorrência mais reduzidas face às anteriores. Contudo, representam o limite mínimo para λ_e .

3.4.9 Modelo mRD : mR Dual

Muitas redes de campo permitem a utilização de meios físicos de transmissão redundantes, que na esmagadora maioria dos casos se resumem a um meio de transmissão dual.

Se for assumido que a rede possui um meio físico de transmissão dual, é necessário alterar os modelos propostos anteriormente para contemplar este aspecto,

nomeadamente para incluir a possibilidade de existirem falhas cobertas ao nível do meio de transmissão.

Neste sentido e após a falha de um dos meios de transmissão³³, existe uma probabilidade da rede ser reconfigurada com sucesso e continuar a funcionar com um só meio físico. A eficiência destes mecanismos é representada através do factor de cobertura do meio de transmissão c_b . Assumiui-se também que no caso de existir mais dos que um componente avariado, a reparação do meio de transmissão tem prioridade sobre os restantes³⁴.

Neste modelo existe uma avaria da rede quando, para além das situações descritas para o modelo **mR**, ocorrer uma falha não-coberta do meio de transmissão.

Na cadeia de Markov do modelo (Fig. 3.12) o sub-índice de cada estado representa o número de meios de transmissão operacionais.

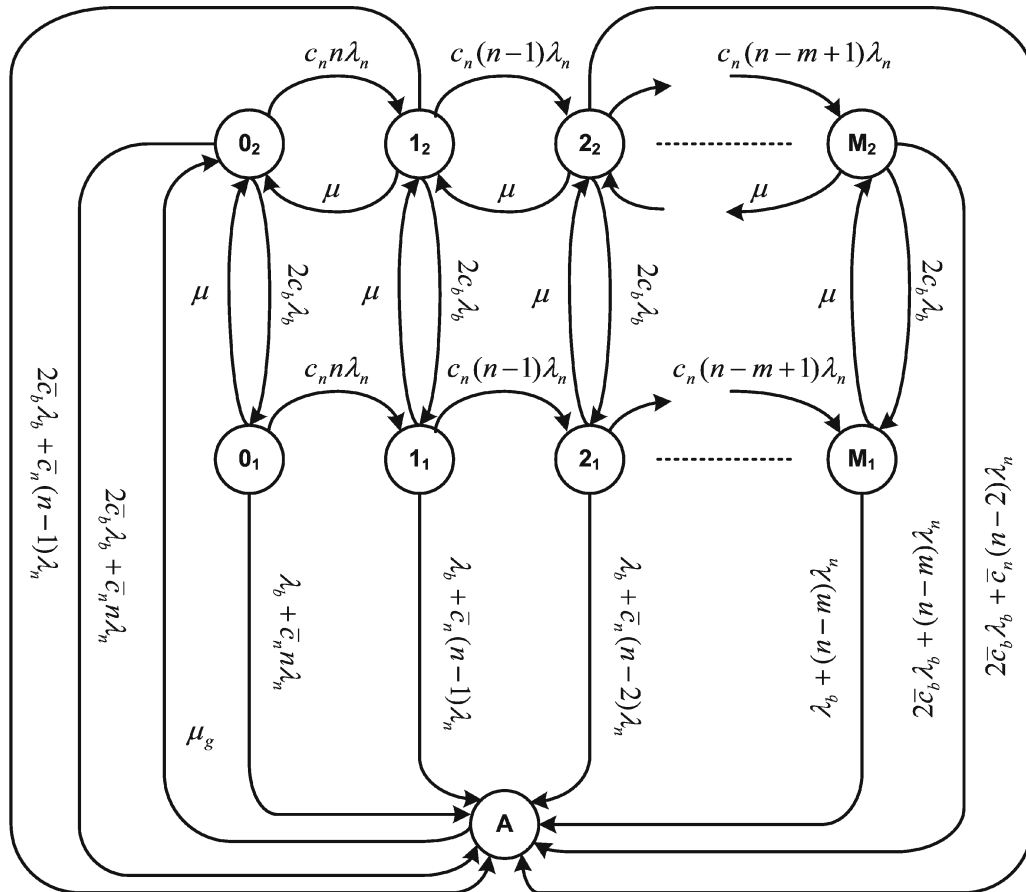


Figura 3.12 - Cadeia de Markov associada ao modelo **mRD**.

33 Assume-se que ambos os meios têm características semelhantes em termos de ocorrência de falhas, mantendo-se assim a taxa de avarias igual para ambos.

34 Este aspecto é arbitrário e pode ser facilmente modificado.

A expressão de λ_e é

$$\begin{aligned}\lambda_e \approx & 2\bar{c}_b\lambda_b + \bar{c}_n n\lambda_n \\ & + \frac{c_n n\lambda_n}{\mu} [2\bar{c}_b\lambda_b + \bar{c}_n(n-1)\lambda_n] \\ & + \frac{2c_b\lambda_b}{\mu} (\lambda_b + \bar{c}_n n\lambda_n)\end{aligned}\tag{3.18}$$

Dos quatro termos da expressão, e de acordo com os pressupostos assumidos previamente, só os dois primeiros são relevantes. Os factores de cobertura c_n e c_b têm uma importância fundamental, dado que limitam de forma muito importante os valores de λ_b e λ_n , respectivamente, implicando assim um aumento da confiança no funcionamento da rede.

3.4.10 Modelo mRDCC : mR Dual modo-comum complexo

A introdução de um meio de transmissão redundante obriga também a considerar os efeitos de avarias em modo-comum que afectem ambos os meios de transmissão. É assim necessário alterar os modelos propostos anteriormente, por forma a contemplar este aspecto.

Tendo em conta as diferenças que existem ao nível da complexidade entre os vários componentes, procurou-se neste modelo ser um pouco mais rigoroso na definição dos factores de cobertura. Desta forma, assumiram-se os seguintes factores de cobertura: c_n para as falhas que afectam só nós, c_{nb} para as falhas que afectam simultaneamente os nós e um dos meios de transmissão e c_b para as falhas que só afectem os meios de transmissão.

Como neste caso existe um meio de transmissão redundante, é necessário considerar os processos de falhas em modo-comum que afectam ambos os meios de transmissão. Neste sentido, um processo que afecte simultaneamente j nós e ambos os meios de transmissão é caracterizado por uma taxa de ocorrência λ_{cnbb}^j , e um processo que só afecte ambos os meios de transmissão é caracterizado por uma taxa de ocorrência λ_{cbb} . Para os processos de falhas em modo-comum que afectam apenas os nós, manteve-se as denominações previamente indicadas.

A avaria da rede decorre da combinação das condições já apresentadas para os modelos mRCC e mRD.

Por questões de simplicidade de apresentação, a cadeia de Markov (Fig. 3.13) terá apenas as transições referentes às falhas em modo-comum.

$$\begin{aligned}
 & + \frac{2c_{nb}n\lambda_{cnb}^1}{\mu} \left[\lambda_b + \bar{c}_n(n-1)\lambda_n + \sum_{j=1}^{n-1} \binom{n-1}{j} \lambda_{cnb}^j + \bar{c}_n \cdot \sum_{j=2}^{m-1} \binom{n-1}{j} \lambda_{cn}^j + \sum_{j=m}^{n-1} \binom{n-1}{j} \lambda_{cn}^j \right] \\
 & + \frac{2c_{nb} \binom{n}{2} \lambda_{cnb}^1}{\mu} [\dots] \\
 & + \frac{2c_{nb} \binom{n}{m} \lambda_{cnb}^m}{\mu} + \left[\lambda_b + (n-m)\lambda_n + \sum_{j=1}^{n-m} \binom{n-m}{j} \lambda_{cnb}^j + \sum_{j=2}^{n-m} \binom{n-m}{j} \lambda_{cn}^j \right] \\
 & + \frac{c_n \binom{n}{2} \lambda_{cn}^2}{\mu} + \left[2\bar{c}_b \lambda_b + \bar{c}_n(n-2)\lambda_n + 2\bar{c}_{nb} \cdot \sum_{j=1}^{m-2} \binom{n-2}{j} \lambda_{cnb}^j + 2 \cdot \sum_{j=m-1}^{n-2} \binom{n-2}{j} \lambda_{cn}^j \right. \\
 & \left. + \bar{c}_n \cdot \sum_{j=2}^{m-2} \binom{n-2}{j} \lambda_{cn}^j + \sum_{j=m-1}^{n-2} \binom{n-2}{j} \lambda_{cn}^j + \sum_{j=1}^{n-2} \binom{n-2}{j} \lambda_{cnbb}^j + \lambda_{cbb} \right] \\
 & + \dots \\
 & + \frac{c_n \binom{n}{m} \lambda_{cn}^m}{\mu} + \left[2\bar{c}_b \lambda_b + (n-m)\lambda_n + 2 \cdot \sum_{j=1}^{n-m} \binom{n-m}{j} \lambda_{cnb}^j \right. \\
 & \left. + \sum_{j=2}^{n-m} \binom{n-m}{j} \lambda_{cn}^j + \sum_{j=1}^{n-m} \binom{n-m}{j} \lambda_{cnbb}^j + \lambda_{cbb} \right]
 \end{aligned}$$

Também neste caso a avaliação deste modelo resulta necessariamente complexa, dado que é necessário conhecer os valores associados aos parâmetros λ_{cn}^j , λ_{cnb}^j , λ_{cnbb}^j e λ_{cbb} , cuja determinação se torna, na prática, extremamente difícil.

Contudo, tendo em conta as expressões obtidas e o conhecimento real do funcionamento da rede, é possível realizar algumas simplificações baseadas nos seguintes pressupostos:

- ♦ A evolução de $\binom{n}{j}$ em função de j segue uma curva com “aspecto” gaussiano;
- ♦ Para valores crescentes de j , λ_{cn}^j , λ_{cnb}^j , λ_{cnbb}^j tomam, com uma probabilidade muito elevada, valores fortemente decrescentes. Este pressuposto baseia-se no facto (observável) de que é muito mais provável falharem simultaneamente j componentes, do que $j+1$.

Assim, partindo dos pressupostos anteriores e assumindo um exemplo concreto, em que as falhas em modo-comum mais relevantes são as que envolvem até 3 componentes, a expressão anterior seria simplificada para

$$\begin{aligned}
 \lambda_e \approx & 2\bar{c}_b\lambda_b + \bar{c}_n \left[n\lambda_n + \binom{n}{2}\lambda_{cn}^2 + \binom{n}{3}\lambda_{cn}^3 \right] + 2\bar{c}_{nb} \left[n\lambda_{cnb}^1 + \binom{n}{2}\lambda_{cnb}^2 \right] + n\lambda_{cnbb}^1 + \lambda_{cbb} \\
 & + \frac{c_n n \lambda_n}{\mu} \left[2\bar{c}_b\lambda_b + \bar{c}_n \left[(n-1)\lambda_n + \binom{n-1}{2}\lambda_{cn}^2 + \binom{n-1}{3}\lambda_{cn}^3 \right] \right. \\
 & \left. + \left[(n-1)\lambda_{cnb}^1 + \binom{n-1}{2}\lambda_{cnb}^2 \right] + (n-1)\lambda_{cnbb}^1 + \lambda_{cbb} \right] \\
 & + \frac{c_b\lambda_b}{\mu} \left[\lambda_b + \bar{c}_n \left[n\lambda_n + \binom{n}{2}\lambda_{cn}^2 + \binom{n}{3}\lambda_{cn}^3 \right] + n\lambda_{cnb}^1 + \lambda_{cbb} \right] \\
 & + \frac{c_{nb} n \lambda_{cnb}^1}{\mu} \left[\lambda_b + \bar{c}_n \left[(n-1)\lambda_n + \binom{n-1}{2}\lambda_{cn}^2 + \binom{n-1}{3}\lambda_{cn}^3 \right] + (n-1)\lambda_{cnb}^1 + \lambda_{cbb} \right] \\
 & + \frac{c_{nb} \binom{n}{2} \lambda_{cnb}^2}{\mu} \left[2\bar{c}_b\lambda_b + \bar{c}_n \left[(n-2)\lambda_n + \binom{n-2}{2}\lambda_{cn}^2 + \binom{n-2}{3}\lambda_{cn}^3 \right] \right. \\
 & \left. + 2\bar{c}_{nb} \left[(n-2)\lambda_{cnb}^1 + \binom{n-2}{2}\lambda_{cnb}^2 \right] + (n-2)\lambda_{cnbb}^1 + \lambda_{cbb} \right]
 \end{aligned} \tag{3.20}$$

Analisando apenas os termos associados às falhas em modo-comum, e de acordo com os pressupostos assumidos previamente, só os quatro primeiros termos são relevantes. Estes por sua vez podem ser agrupados da seguinte forma:

- Os que limitam o valor mínimo de λ_e , tais como λ_{cnbb}^1 e λ_{cbb} , que estão directamente associados à falha simultânea de ambos os meios de transmissão;
- Os que dependem fortemente dos factores de cobertura c_{nb} e c_n . Comparando estes dois grupos é simples de verificar que este último assume um papel muito mais relevante.

3.4.11 Extensões dos Modelos

Como foi discutido previamente, é possível através da composição dos modelos propostos obter modelos que permitam a avaliação de outras arquitecturas com características mais específicas.

Um aspecto interessante que resulta da aplicação do método dos estados de funcionamento crítico, está relacionado com o facto de que as expressões obtidas para estas novas arquitecturas não diferirem em muito das obtidas para os modelos já apresentados. Isto à primeira vista pode parecer um pouco estranho, mas se se analisar com cuidado a forma como as expressões de λ_e são obtidas, e

que tipos de simplificações são realizadas, comprovar-se-á³⁵ esta situação. De facto as expressões obtidas correspondem à “leitura” dos percursos “mais curtos” entre o estado **0** e o estado **A**, sendo os restantes percursos na prática ignorados. Desta forma, como os modelos das novas arquitecturas são o resultado da composição dos modelos base estudados, os percursos obtidos são muito semelhantes (não são exactamente iguais mas muito próximos) e o mesmo se passando com as expressões da taxa equivalente de avarias.

Um outro aspecto que se poderia incluir nos modelos, seria considerar a existência de modos de avaria distintos. Assim, e de acordo com [IEC04a] poder-se-ia adoptar a existência de avarias seguras (*fail safe*) e não-seguras (*fail not-safe*). Ao nível dos modelos desenvolvidos, isto resultaria na introdução de um segundo estado de avaria, e obviamente também numa redefinição quer das taxas de avarias dos componentes, quer nas características dos mecanismos de tolerância a falhas. Em ambos os casos seria necessário considerar que existem falhas seguras e não-seguras dos componentes, que podem dar origem a avarias da rede com a mesma natureza [Gob98]. Ainda neste contexto e ao nível das medidas, seria interessante calcular o tempo médio até uma avaria (da rede) não-segura (*Mean Safe Time*), cujo interesse é de especial importância nos sistemas de segurança crítica (*safety-critical systems*). Partindo da metodologia proposta em [CWT98], onde se define uma abordagem para calcular o MTTF condicional, a medida em causa poderia ser facilmente obtida.

3.4.12 Caso de Estudo

Com o objectivo de validar a metodologia proposta foi definido um caso de estudo.

Com este estudo pretendeu-se atingir dois objectivos. Em primeiro lugar analisar a qualidade da aproximação introduzida pelo método dos estados de funcionamento crítico. Esta análise foi realizada utilizando a ferramenta de modelação de cadeias de Markov SURF-2 [BAA+93] [MAA94] para a resolução dos modelos propostos. Os resultados obtidos são posteriormente comparados com as expressões analíticas desenvolvidas, sendo a diferença apresentada a sob a forma do erro cometido na aproximação.

O segundo objectivo, consiste em identificar quais os parâmetros dos modelo que possuem maior influência no comportamento da rede. Com isto pretende-se ir ao encontro dos objectivos do desenvolvimento de um modelo preliminar. Ou seja, através da identificação dos parâmetros mais importantes, simplificar os modelos propostos considerando apenas os mais relevantes. Esta identificação foi realizada através de uma análise de sensibilidade dos modelos.

³⁵ Este aspecto foi verificado.

Como o caso de estudo é bastante extenso, optou-se por incluí-lo em separado no Anexo 2. Para cada um dos modelos propostos é realizado um estudo detalhado e são identificados os parâmetros mais importantes. No entanto, para permitir uma contextualização das conclusões optou-se por incluir neste capítulo alguns dos resultados obtidos. Isto não deve dispensar contudo a consulta deste anexo, dado que neste são descritos de forma pormenorizada a forma como os resultados foram obtidos.

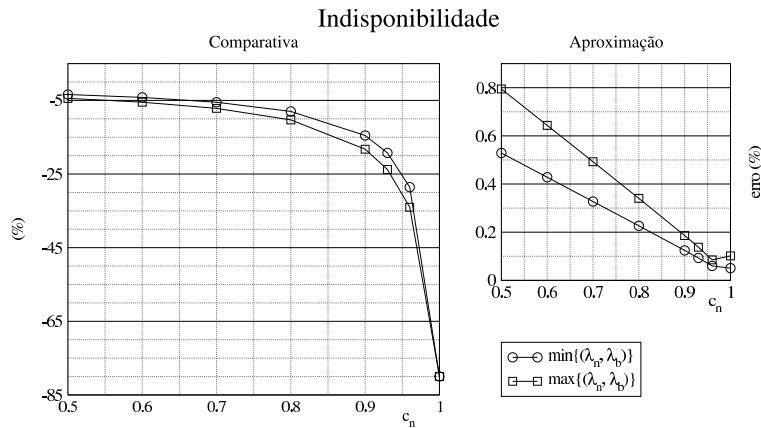


Figura 3.14 - Modelo mRD vs. mR: indisponibilidade (c_n).

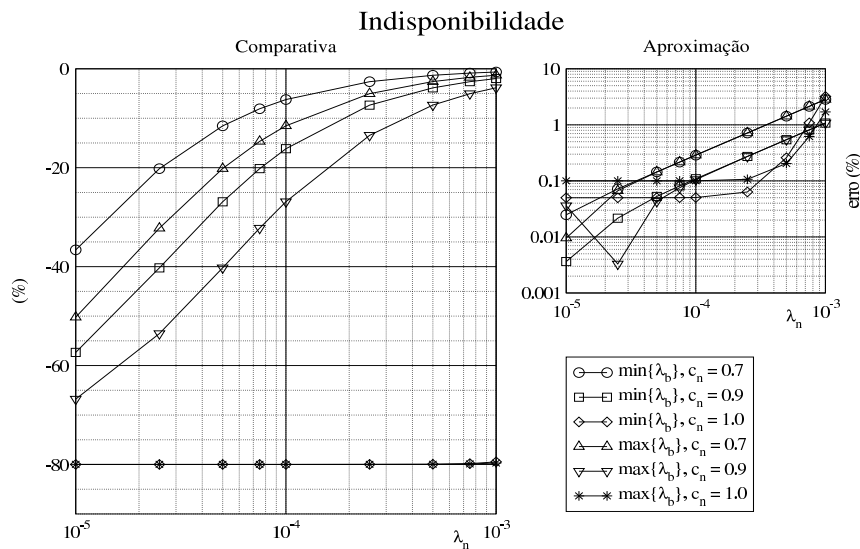


Figura 3.15 - Modelo mRD vs. mR: indisponibilidade (λ_n).

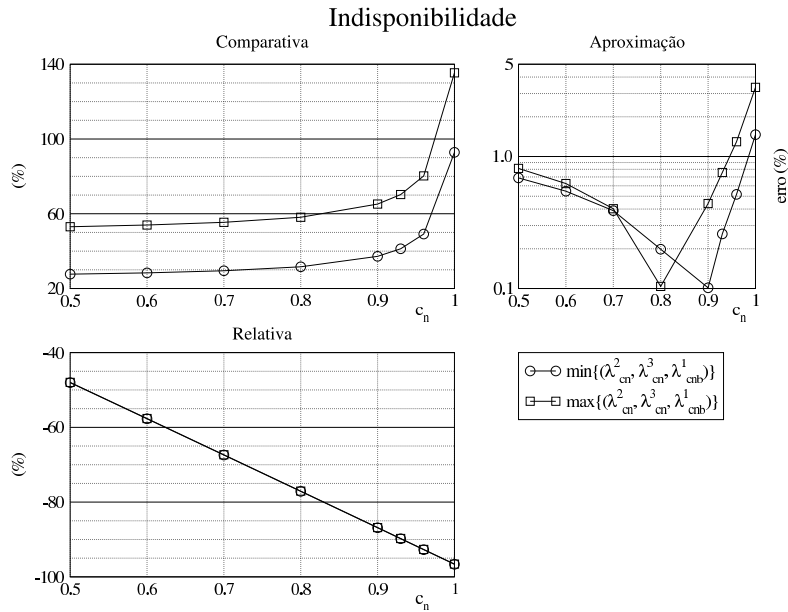


Figura 3.16 - Modelo mRCC vs. mR: indisponibilidade (c_n).

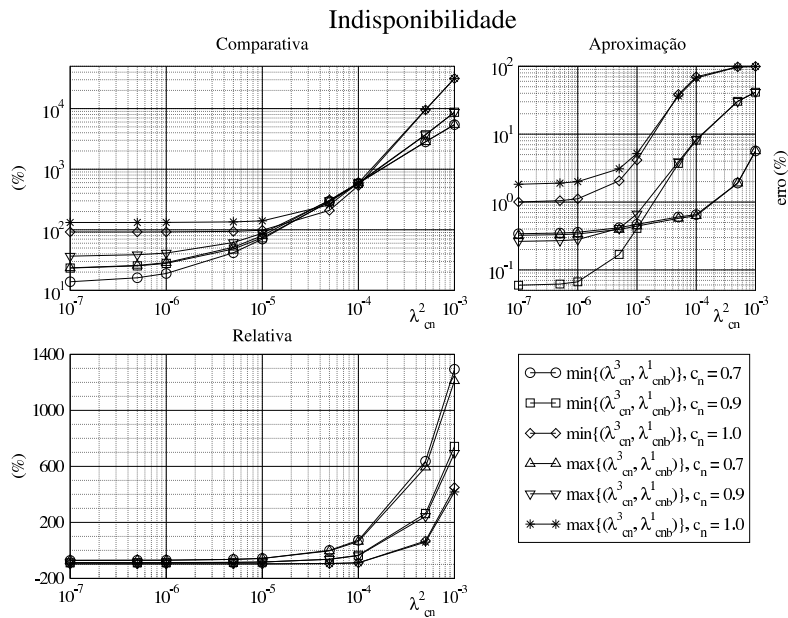


Figura 3.17 - Modelo mRCC vs. mR: indisponibilidade (λ_{cn}^2).

3.4.12.1 Avaliação de Resultados

A partir do caso de estudo realizado é possível extrair as seguintes conclusões:

- ♦ O factor de cobertura c_n é o parâmetro que mais influencia a confiança no funcionamento da rede;
- ♦ Quando $c_n < 0.9$ (Fig. 3.14) a utilização de um meio de transmissão redundante não implica um aumento apreciável da qualidade das medidas obtidas. Isto resulta do facto, de quando λ_b é reduzido (como é típico) as medidas são praticamente condicionadas por c_n e λ_n ;
- ♦ As variações relativas de λ_n são apenas significativas quando este parâmetro toma valores pequenos, correspondendo à situação em que λ_n toma valores próximos de λ_b (Fig. 3.15). Isto permite concluir que a avaliação poder ser realizada sem ter que dispor de valores muito rigorosos para λ_n ;
- ♦ O efeito de λ_b é de certa forma limitado como resultado, em parte, do pequeno valor utilizado neste caso de estudo (Fig. 3.15);
- ♦ As falhas em modo-comum no seu conjunto têm um efeito muito considerável (mesmo com percentagens muito pequenas) no comportamento da rede (Fig. 3.16);
- ♦ O efeito de λ_{cn}^2 assume um papel muito importante quando este toma valores elevados (Fig. 3.17). Isto resulta do facto deste parâmetro estar associado a termos do tipo $\binom{n}{2}$, que tipicamente também tomam valores elevados;
- ♦ Os erros cometidos entre os valores reais e aproximados das medidas, são, para quase todas os casos, muito reduzidos (tipicamente $\sim 1\%$), confirmando assim, que a utilização do método dos estados de funcionamento crítico permite obter resultados com um rigor muito elevado.

3.5 Modelo de Falhas Transitórias

As secções seguintes descrevem a aplicação da metodologia definida inicialmente para o desenvolvimento dos modelos de falhas transitórias que ocorrem durante as comunicações.

O desenvolvimento destes modelos foi realizado de acordo com uma abordagem muito próxima da estrutura da *stack* de comunicações. Numa primeira fase são obtidos modelos que definem o comportamento ao nível da camada física, que posteriormente são utilizados para definir os modelos ao nível da camada de

ligação de dados. São analisados e modelados diferentes tipos de tráfego na rede e por fim é apresentado um caso de estudo com o objectivo de validar a metodologia proposta.

3.5.1 Pressupostos Assumidos

Partindo da metodologia definida previamente, estabeleceram-se os seguintes pressupostos no desenvolvimento dos modelos:

- ♦ A rede é constituída por um conjunto de nós interligados por intermédio de uma infra-estrutura de comunicações. É assumido, até referência em contrário, que o meio físico de transmissão não é redundante;
- ♦ O controlo do processo físico é assegurado por intermédio de uma aplicação distribuída constituída por um conjunto de tarefas que são executadas nos vários nós da rede. Estas tarefas estão organizadas de acordo com uma estrutura funcional adequada ao processo em causa;
- ♦ A troca de dados de controlo entre tarefas é assegurada pelo envio e recepção de mensagens, utilizando para isso os serviços disponibilizados pela *stack* de comunicações, que por sua vez faz parte da infra-estrutura de comunicações. De acordo com os objectivos definidos previamente, é apenas considerado o comportamento da *stack* até à camada de ligação de dados (inclusivé);
- ♦ As mensagens são encapsuladas em tramas para posterior transmissão. A transmissão de uma mensagem pode implicar a transmissão de várias tramas. O comprimento das tramas é assumido variável de acordo com uma função distribuição $F_{CT}(t)$ com valor esperado (médio) T_{CT} ;
- ♦ O sistema de controlo que se pretende avaliar possui requisitos de tempo-real associados às comunicações, concretizados através da definição de *deadlines* associadas às mensagens transmitidas;
- ♦ As falhas ocorrem de acordo com um processo de Poisson com taxa λ_f [Triv02]. Este pressuposto corresponde à abordagem clássica da ocorrência de falhas num ambiente industrial com interferências electromagnéticas [Sho94] [KWS00]. A duração da falha é assumida variável de acordo com uma função distribuição $F_{DF}(t)$ com valor esperado (médio) T_{DF} . Assume-se além disso, que os seus efeitos são vistos por todos os nós da rede de forma semelhante;
- ♦ Se durante a transmissão de uma mensagem ocorrer uma falha, assume-se que esta provoca a destruição do conteúdo da mesma através da introdução de erros. É assumido que os erros são sempre detectados pelos nós, induzindo a execução de mecanismos específicos (ex. retransmissões);
- ♦ A principal medida de interesse é determinar qual a probabilidade das *deadlines* associadas às mensagens serem violadas.

3.5.2 Camada Física

Nas secções seguintes desenvolvidos vários modelos que definem as condições necessárias para a transmissão com sucesso de uma trama³⁶. Estas condições são formalizadas através da determinação de um conjunto de expressões analíticas que definem a respectiva probabilidade de sucesso.

Neste sentido, são propostos dois modelos de falhas alternativos que descrevem a sua ocorrência e os seus efeitos sobre as tramas. Estes modelos são posteriormente extendidos para incluir a possibilidade de dispor de um meio físico de transmissão redundante, considerando-se também os efeitos das avarias em modo-comum no mesmo.

3.5.2.1 Falhas Pontuais

A situação mais simples consiste em considerar que após a ocorrência de uma falha, a sua duração pode ser considerada desprezável face ao comprimento típico de uma trama. Ou seja, as falhas podem ser consideradas pontuais. Neste caso a trama só é afectada se ocorrerem falhas durante a respectiva transmissão.

De acordo com este modelo, a probabilidade $Prob(n, t)$ de ocorrerem n falhas durante um intervalo de tempo t é dada por

$$Prob(n, t) = e^{-\lambda_f t} \cdot \frac{(\lambda_f t)^n}{n!} \quad (3.21)$$

A probabilidade $Prob_{FTT}(n)$ de ocorrerem n falhas durante a transmissão de uma trama pode ser obtida a partir da regra da probabilidade total

$$Prob_{FTT}(n) = \int_0^{+\infty} Prob(n, t) \cdot dF_{CT}(t) \quad (3.22)$$

em que o integral é do tipo Stieltjes [Kao97], cobrindo assim distribuições contínuas, discretas ou a mistura de ambas.

A probabilidade de transmitir uma trama com sucesso $Prob_{TS}$, é igual à probabilidade de não ocorrerem quaisquer falhas durante a respectiva transmissão

$$Prob_{TS} = Prob_{FTT}(n=0) = \int_0^{+\infty} e^{-\lambda_f t} dF_{CT}(t) \quad (3.23)$$

³⁶ Embora o conceito de *trama* diga respeito à camada de ligação de dados, a definição destes modelos como pertencendo à camada física justifica-se pelo facto dos efeitos das falhas se situarem a este nível. Assim, esta classificação deve ser interpretada neste contexto.

Avaliação Preliminar da Confiança no Funcionamento

Para calcular a expressão anterior é necessário conhecer qual a função distribuição associada ao comprimento da trama, o que normalmente é desconhecido. No entanto verifica-se que em termos práticos as aplicações de controlo suportadas por redes de campo são caracterizadas por uma troca de mensagens cujo comprimento apresenta tipicamente uma baixa variância (ex. muitas das mensagens possuem um comprimento fixo). Desta forma, este modelo pode ainda ser mais simplificado se for considerado que a trama possui um comprimento constante. Nestas condições, a função densidade de probabilidade é definida por uma função de Dirac

$$\frac{dF_{CT}}{dt} = \delta(T_{CT}) \quad (3.24)$$

e a probabilidade de transmissão com sucesso reduz-se a

$$Prob_{TS} = \exp(-\lambda_f \cdot T_{CT}) \quad (3.25)$$

Em situações em que o comprimento da trama é predominantemente curto, exibindo além disso uma variância média/elevada, é possível utilizar a distribuição exponencial (ou variantes como Weibull ou Erlang), como aproximações razoáveis da função distribuição. Note-se contudo, que a formalização deste aspecto deve ser precedida de uma análise do tráfego da rede.

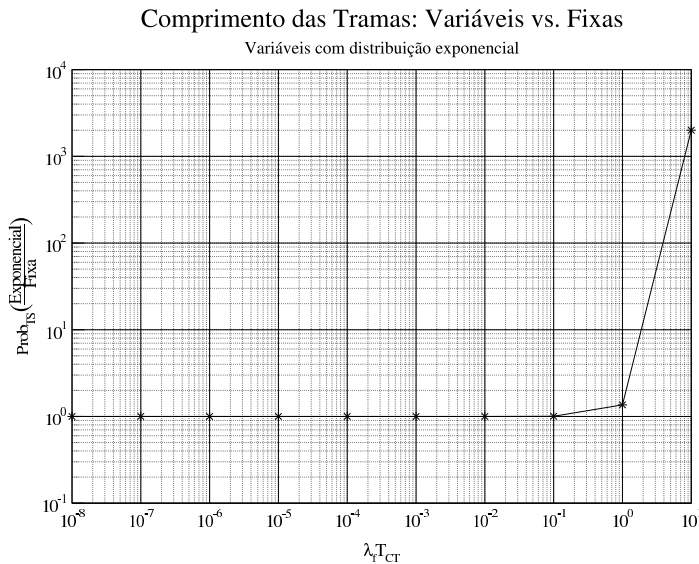


Figura 3.18 - Probabilidade de transmissão de uma trama com sucesso: duração variável (exponencial) vs. fixa.

Como forma de analisar a importância relativa entre estas duas abordagens, foi realizado um pequeno estudo em que foi comparada uma trama de duração variável de acordo com uma distribuição exponencial, com uma trama de duração

fixa (Fig. 3.18). Esta análise foi realizada através da determinação da evolução do quociente entre ambas as probabilidades de sucesso $Prob_{TS}$ em função de $\lambda_f T_{CT}$.

A evolução comparativa entre ambas, demonstra que a diferença entre as duas abordagens só se torna significativa quando o comprimento da trama T_{CT} se torna próximo de $1/10$ do tempo médio entre falhas $1/\lambda_f$. Tipicamente este cenário só se verifica quando as velocidades de transmissão da rede são muito reduzidas e a taxa de ocorrência das falhas muito elevada. É assim possível concluir, que em situações normais de operação da rede assumir que as tramas possuem um comprimento fixo é perfeitamente razoável, introduzindo um erro mínimo.

3.5.2.2 Falhas com Duração Associada

As falhas típicas que ocorrem num ambiente industrial podem em muitas situações exibir um comportamento em que os pressupostos do modelo anterior deixam de ser válidos. O caso mais comum ocorre quando se verifica que a falha³⁷ possui uma duração que não pode se considerada desprezável.

Neste caso, quando a falha ocorre os seus efeitos estendem-se no tempo, podendo contribuir para afectar futuras transmissões. Se for assumido que a falha possui uma duração variável, as condições necessárias para a transmissão de uma trama com sucesso são as seguintes:

- ♦ Não podem ocorrer falhas durante a transmissão da trama. Esta cenário foi discutido e analisado no modelo anterior. Assume-se que a respectiva probabilidade de sucesso é $Prob_{TS}^d$;
- ♦ As falhas que ocorrem previamente à transmissão da trama não a podem afectar. A quantificação deste efeito pode ser obtida a partir da análise da Figura 3.19.

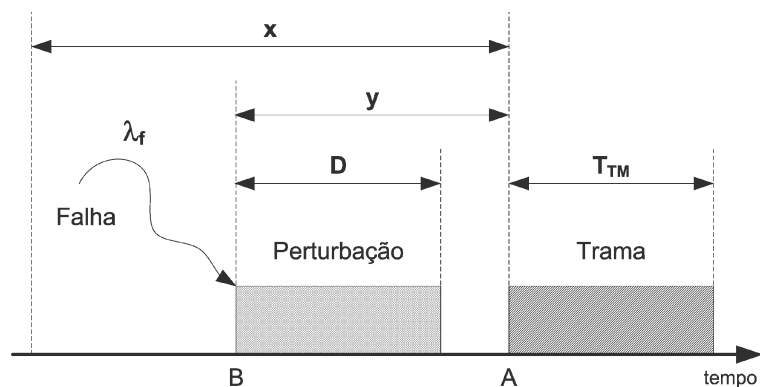


Figura 3.19 - Falhas que ocorrem previamente à transmissão.

³⁷ O termo mais correcto talvez fosse perturbação, dado que a falha é a causa do erro (perturbação da trama). No entanto dado o contexto da discussão preferiu-se manter o termo *falha*.

Considere-se que no instante **A** uma trama de comprimento T_{TM} (fixa ou variável) é transmitida, e que no instante **B** uma falha (perturbação) com duração D ocorreu. Se $D \geq y$ então a falha vai afectar a trama³⁸. De acordo com os pressupostos definidos para a ocorrência das falhas, demonstra-se que se n falhas ocorrem num intervalo de tempo x então o seu instante de ocorrência é uniformemente distribuído [Kao97] [Triv02].

Nestas condições a probabilidade $Prob_{TS}^a$ que a trama não seja afectada por uma falha que ocorra antes da sua transmissão pode ser obtida aplicado a regra da probabilidade total

$$\begin{aligned} Prob_{TS}^a &= \sum_{n=0}^{+\infty} Prob(x, n) \cdot (1 - Prob_y)^n \\ Prob_y &= \frac{1}{x} \int_0^x [1 - F_{DF}(t)] dt \end{aligned} \quad (3.26)$$

em que $Prob_y$ é a probabilidade de uma falha ocorrer num instante $B \in [0 \dots x]$ com duração superior a y .

Desta forma

$$Prob_{TS}^a = \exp(-\lambda_f x Prob_y) \quad (3.27)$$

Considerando o limite $x \rightarrow +\infty$, ou seja considerando o efeito de todas as falhas desde $t = -\infty$

$$\lim_{x \rightarrow \infty} x \cdot Prob_y = \int_0^{+\infty} [1 - F_{DF}(t)] dt = T_{DF} \quad (3.28)$$

em que T_{DF} é o valor esperado (médio) da duração da falha. Assim, das duas expressões anteriores obtêm-se

$$Prob_{TS}^a = \exp(-\lambda_f \cdot T_{DF}) \quad (3.29)$$

É interessante verificar que este resultado é de certa forma muito intuitivo. O que a expressão indica é que a contribuição de todas as falhas vai ser equivalente ao efeito de uma falha de comprimento médio.

Devido às propriedades do processo de Poisson $Prob_{TS}^d$ e $Prob_{TS}^a$ são independentes e a probabilidade $Prob_{TS}$ de transmissão de uma trama com sucesso é obtida como

³⁸ Assume-se que se a perturbação atingir a trama, mesmo que minimamente, causa um erro.

$$Prob_{TS} = Prob_{TS}^a \cdot Prob_{TS}^d \quad (3.30)$$

Uma análise detalhada dos termos desta expressão mostra que $Prob_{TS}$ é independente do instante de tempo referente à transmissão da trama, dependendo apenas das características temporais das tramas e das falhas (duração e taxa respectivamente). Esta é uma propriedade muito importante que permite que cada trama possa ser analisada de forma independente das restantes.

Para analisar a importância relativa de ambos os modelos de falhas, foi realizado um pequeno estudo em que foi comparada a evolução do quociente de $Prob_{TS}$ em ambos os modelos em função de $\lambda_f T_{DF}$ (Fig. 3.20).

A evolução comparativa de ambos os modelos, demonstra que é apenas necessário considerar o efeito da duração das falhas, se a duração média das mesmas T_{DF} for superior a $1/10^3$ do tempo médio entre estas $1/\lambda_f$. Este cenário só ocorre em situações em que existe uma grande intensidade de falhas com duração muito longa. É assim possível concluir, que em situações normais de operação da rede considerar que a falha é pontual é perfeitamente aceitável, introduzindo um erro mínimo.

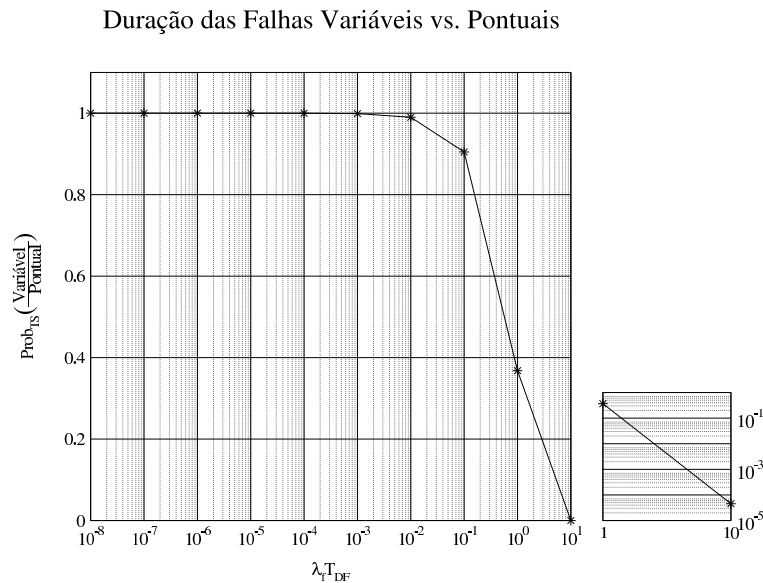


Figura 3.20 - Duração das falhas: variáveis vs. pontuais.

3.5.2.3 Meio Físico de Transmissão Redundante

No caso da rede permitir a utilização de um meio físico de transmissão redundante, os modelos de falhas discutidos previamente tem que ser modificados para incorporar esta situação.

Neste sentido, foram assumidos os seguintes pressupostos adicionais:

- ♦ Existem dois meios físicos de transmissão³⁹, denominados **A** e **B**. As falhas ocorrem de forma independente em cada meio com taxas λ_f^A e λ_f^B respectivamente;
- ♦ Assume-se que a redundância existe apenas ao nível da camada física, sendo a sua gestão realizada de forma transparente pela *stack* de comunicações;
- ♦ Em consequência do pressuposto anterior, assume-se que as tramas são transmitidas de forma independente em cada meio;
- ♦ Prende-se tal como nos modelos anteriores, determinar a probabilidade $Prob_{TS}$ de transmissão com sucesso de uma trama, ao nível dos serviços de comunicação invocados pela camada de ligação de dados.

Uma trama é transmitida com sucesso com probabilidade $Prob_{TS}$, se for transmitida com sucesso em pelo menos um dos meios de transmissão. Desta forma

$$\begin{aligned} Prob_{TS} &= Prob\{T_{AS} \cup T_{BS}\} \\ &= Prob\{T_{AS}\} + Prob\{T_{BS}\} - \{T_{AS} \cap T_{BS}\} \end{aligned} \quad (3.31)$$

em que T_{is} representa a o evento referente à transmissão com sucesso de uma trama no meio $i \in \{A, B\}$ com probabilidade $Prob\{T_{is}\}$, cujo valor pode ser obtido a partir das expressões deduzidas nas secções anteriores.

Como as transmissões são independentes

$$Prob\{T_{AS} \cap T_{BS}\} = Prob_{TS}(\lambda_f^A) \cdot Prob_{TS}(\lambda_f^B) \quad (3.32)$$

em que $Prob_{TS}(\lambda_f^i)$ representa a probabilidade de transmissão com sucesso de uma trama com $\lambda_f = \lambda_f^i$ $i \in \{A, B\}$.

Assim

$$Prob_{TS} = Prob_{TS}(\lambda_f^A) + Prob_{TS}(\lambda_f^B) - Prob_{TS}(\lambda_f^A) \cdot Prob_{TS}(\lambda_f^B) \quad (3.33)$$

3.5.2.3.1 Avarias em Modo-Comum

Embora na secção anterior se tenha assumido que as falhas ocorrem de forma independente em cada meio, é possível que sujam situações em que uma falha

³⁹ A análise poderia ser facilmente extendida a um maior número de meios de transmissão, contudo a situação mais usual é utilizar um meio dual.

possa afectar simultaneamente a transmissão de uma trama em ambos os meios. Ou seja, podem ocorrer falhas em modo-comum⁴⁰.

Para modelar este efeito assumiu-se que as falhas em modo-comum ocorrem de acordo como o modelo β discutido previamente. Como se trata de uma forma de redundância muito simples (apenas 2 elementos) não existe qualquer vantagem na utilização do modelo Binomial, dado que os resultados obtidos seriam os mesmos. Além disso, o modelo proposto tem apenas um parâmetro, sendo assim mais simples estimar o respectivo valor. Neste sentido, foram definidas as seguintes relações entre as taxas de ocorrência das falhas

$$\begin{aligned}\lambda_f^i &= (1 - \beta_i) \lambda_f \\ \lambda_c &= \beta_i \lambda_f^i\end{aligned}\tag{3.34}$$

em que β_i representa a fracção de falhas de modo comum que afectam o meio de transmissão $i \in \{A, B\}$. Existem assim três processos simultâneos de ocorrência de falhas: λ_f^A , λ_f^B e λ_c , em que este último representa as falhas em modo-comum.

Utilizando a mesma abordagem da secção anterior

$$Prob_{TS} = Prob\{T_{AS} \cup T_{BS}\}\tag{3.35}$$

Através da propriedade da independência

$$\begin{aligned}Prob\{T_{AS} \cap T_{BS}\} &= Prob\{T_{AS} | T_{BS}\} \cdot Prob\{T_{BS}\} \\ &= Prob_{TS}(\lambda_f^A) \cdot Prob\{T_{BS}\}\end{aligned}\tag{3.36}$$

se a trama foi transmitida com sucesso no meio **B**, então $Prob\{T_{AS} | T_{BS}\}$ só depende da probabilidade ser transmitida com sucesso no meio **A** (não existem falhas em modo-comum).

Em cada meio, a trama só é transmitida com sucesso se não for afectada por nenhum processo de falhas

$$Prob\{T_{is}\} = Prob_{TS}(\lambda_f^i) \cdot Prob_{TS}(\lambda_c)\tag{3.37}$$

Assim

$$Prob_{TS} = Prob_{TS}(\lambda_c) \cdot [Prob_{TS}(\lambda_f^A) + Prob_{TS}(\lambda_f^B) - Prob_{TS}(\lambda_f^A) \cdot Prob_{TS}(\lambda_f^B)]\tag{3.38}$$

⁴⁰ Está-se a utilizar a terminologia definida previamente.

3.5.3 Camada de Ligação de Dados

A definição das condições de avaria da rede estão intimamente associadas à forma como as mensagens são transmitidas, que por sua vez reflectem as características de protocolo de comunicações⁴¹ utilizado pela rede. Torna-se assim necessário que o desenvolvimento de um modelo que descreva a transmissão das mensagens tenha que incluir, de forma directa ou indirecta, também o comportamento do respectivo protocolo de comunicações.

No contexto da avaliação preliminar discutida previamente, isto implica propor um modelo com um grau de detalhe reduzido e suficientemente genérico para que possa ser adaptado às características de funcionamento de uma rede em particular. Face às várias condicionantes deste problema, utilizou-se uma abordagem, à custa de algumas simplificações, com objectivo de separar este problema em duas partes hierarquicamente relacionadas. Pretende-se assim, obter um modelo analiticamente tratável que descreva o comportamento completo da rede neste tipo de cenários.

Numa primeira fase, são desenvolvidos modelos que descrevam o comportamento durante a transmissão de uma mensagem. Estes modelos têm naturalmente que incluir os aspectos relacionados com o protocolo de comunicações, mas de acordo com os objectivos acima propostos. A segunda fase, passa por utilizar estes modelos conjuntamente com o conhecimento das *deadlines* associadas às mensagens para definir os modos de avaria da rede. Nas secções seguintes são discutidos os vários aspectos relacionados com cada uma das fases.

3.5.3.1 Modelo da Transmissão das Mensagens

A discussão realizada nas secções anteriores, para além de formalizar um conjunto de expressões analíticas que permitem obter a probabilidade da transmissão com sucesso de uma trama, permitiu também demonstrar que esta probabilidade é independente do instante da transmissão das mesmas. Como a transmissão de uma mensagem pode ser decomposta na transmissão de várias tramas, resulta que a respectiva probabilidade de transmissão é também independente do instante em que a transmissão ocorre.

A transmissão de uma mensagem pode ser definida de acordo com um conjunto de acções que são função das características do protocolo de comunicações utilizado. Estas acções caracterizam-se por um conjunto de procedimentos (envio e recepção de tramas) que são executados de acordo com uma certa ordem em instantes particulares do tempo. Seria assim lógico assumir que a modelação deste comportamento incluísse ambos os aspectos. No entanto e partindo das conclusões

⁴¹ Assume-se neste contexto que a referência ao termo *protocolo de comunicações* se refere ao protocolo da camada de ligação de dados.

anteriores, se a transmissão de uma mensagem for analisada como uma transacção atómica, é possível abstrair os aspectos temporais do protocolo e considerar apenas a sequência de procedimentos executada. O modelo assim obtido, seria definido apenas a partir da sequência de procedimentos associados à transmissão da mensagem. Desde que a sequência seja correctamente formalizada o modelo obtido pode ser rigoroso, e sem a necessidade de empregar simplificações de maior.

Contudo, esta abordagem assume implicitamente que a transmissão de uma mensagem (sequência de procedimentos) não é afectada pela transmissão de outras mensagens que também fazem parte do tráfego da rede. Ou seja, a transmissão é independente das restantes. Esta é uma hipótese que necessita de ser analisada com alguma atenção.

Em condições de funcionamento normal, ou seja, se se admitir que o tráfego da rede está estabilizado (transmissão de um conjunto fixo de mensagens) e que não existem falhas durante as comunicações, é possível verificar que a transmissão de uma mensagem segue sempre a mesma sequência de procedimentos (ex. transmitir um determinado número de tramas), embora os instantes em que estes ocorram possam variar de transmissão para transmissão. É assim possível concluir que neste contexto as sequências são de facto independentes. Quando ocorrem falhas durante as comunicações, a sequência de procedimentos referentes à mensagem que é afectada pela falha é obviamente alterada, mas só em casos muito particulares é que as sequências das restantes são afectadas. Mesmo nas situações em que existe alguma influência é possível definir um conjunto de cenários em que erro cometido é reduzido. Assim, desde que a taxa de utilização do meio de transmissão (tráfego de mensagens na rede) não seja muito elevada, como acontece na maioria das situações reais, é razoável assumir que as interferências serão pequenas. Além disso e tendo em conta que o comportamento é estocástico, em termos de regime permanente (estocástico) os erros serão de certa forma mais atenuados (ex. existem casos em que há interferências e outros em que não há).

É assim possível concluir em ambos os casos que as sequências de procedimentos referentes a mensagens diferentes são de facto independentes, ou seja, que o erro cometido nesta aproximação é reduzido face ao contexto assumido. Note-se, que esta conclusão decorre do facto do modelo excluir deliberadamente os aspectos temporais associados à transmissão das mensagens. É perfeitamente claro que, mesmo em cenários de ausência de falhas, podem existir situações em que a transmissão de uma mensagem influencia a transmissão das restantes (ex. mecanismo de acesso ao meio de transmissão). Embora os aspectos temporais sejam afectados (instante em que a mensagem ou as respectivas tramas são transmitidas), a sequência de procedimentos não é afectada.

Os modelos propostos para a transmissão de mensagens são simples e não são exaustivos, no sentido em que só alguns tipos particulares de implementação dos protocolos são analisados. Pretende-se assim definir um conjunto de modelos base, que possam ser extendidos ou adaptados a protocolos específicos de acordo com o tipo de rede avaliada.

Neste sentido, foram definidos os seguintes pressupostos adicionais:

- ♦ A transmissão de uma mensagem decorre de acordo com o protocolo de comunicações utilizado;
- ♦ A transmissão de uma mensagem requer a transmissão de N_{TD} tramas de dados e eventualmente algumas tramas de controlo. As respectivas probabilidades de transmissão com sucesso são $Prob_{TDS}$ e $Prob_{TCS}$. Estes valores podem ser obtidos a partir das expressões deduzidas nas secções anteriores;
- ♦ Se durante as transmissões ocorrerem erros (devido a uma falha), o protocolo suporta a existência de retransmissões. É assumido que os erros nas tramas são sempre detectados;
- ♦ Pretende-se determinar expressões referentes à probabilidade de transmissão com sucesso $Prob_{MS}$ de uma mensagem.

Desta forma, e para diferentes implementações dos protocolos:

1. As tramas de dados são transmitidas sem confirmações ou retransmissões

$$Prob_{MS} = (Prob_{TDS})^{N_{TD}} \quad (3.39)$$

2. Cada trama de dados é confirmada utilizando uma (1) trama de controlo. No caso de existirem erros (na trama de dados ou de controlo), a trama de dados é retransmitida até a um limite R no número de retransmissões

$$Prob_{MS} = \left[\sum_{r=0}^R Prob_{TDS} \cdot Prob_{TCS} \cdot (1 - Prob_{TDS} \cdot Prob_{TCS})^r \right]^{N_{TD}} \quad (3.40)$$

3. Cada mensagem transmitida é confirmada utilizando uma (1) trama de controlo. No caso de existirem erros (na mensagem ou na trama de controlo), a mensagem é retransmitida até a um limite R no número de retransmissões

$$Prob_{MS} = \sum_{r=0}^R (Prob_{TDS})^{N_{TD}} \cdot Prob_{TCS} \cdot \left[1 - (Prob_{TDS})^{N_{TD}} \cdot Prob_{TCS} \right]^r \quad (3.41)$$

3.5.3.2 Modelos de Avaria da Rede

Da perspectiva do acesso à informação, uma rede de campo tem como objectivo partilhar, de forma transparente, os dados do processo controlado entre as várias tarefas que são executadas nos nós. É assim possível abstrair este comportamento como uma base de dados distribuída [RVV94]. Neste sentido, as tarefas que executam nos nós acedem aos dados como se tratassem de variáveis⁴² disponíveis localmente. Este acesso é realizado através de uma interface de comunicações (no contexto deste capítulo ao nível da camada de ligação de dados) utilizando para isso serviços de leitura e escrita específicos. Utilizando esta abstracção, a transmissão de variáveis entre as tarefas pode ser formalizada de forma transparente.

Para representar as relações temporais existentes entre as várias entidades envolvidas na transmissão de variáveis, adoptou-se o modelo Produtor-Consumidor [RRD95] [SBS97]. A utilização deste modelo neste contexto, deve ser interpretada apenas como um meio de formalizar as relações temporais existentes entre as várias entidades e não como uma representação do funcionamento do protocolo de comunicações. O objectivo é apenas dispor um modelo que permita descrever de forma adequada os vários aspectos relacionados com o acesso e a transmissão das variáveis. Este modelo é caracterizado pelas seguintes operações (Fig. 3.21):

- ♦ Uma tarefa que executa num nó, produz uma variável e escreve-a (**W**) num *buffer*⁴³ local utilizando para isso um serviço de escrita da interface de comunicações. A variável tem um período de validade T_d , que estabelece um período de tempo após a sua produção em que esta é considerada válida e pode ser utilizada sem prejuízo para o controlo do processo;
- ♦ Os mecanismos locais da *stack* de comunicações copiam a variável do *buffer* local e encapsulam-na numa mensagem. Esta é posteriormente transmitida (**M**) aos restantes nós utilizando para isso o protocolo de comunicações. Após a mensagem ser recebida num outro nó, os mecanismos locais da *stack* de comunicações extraem a variável da mensagem e copiam-na também para um *buffer* local;
- ♦ Nos nós receptores, a tarefa que está interessada na variável procede à sua leitura (**R**) do *buffer* local, utilizando para isso um serviço de leitura da interface de comunicações. Posteriormente a variável é utilizada pela tarefa para realizar operações locais.

⁴² Utilizou-se esta terminologia para englobar todo o tipo de dados que são trocados entre tarefas.

⁴³ Optou-se por utilizar a terminologia em Inglês por se julgar mais adequada ao contexto.

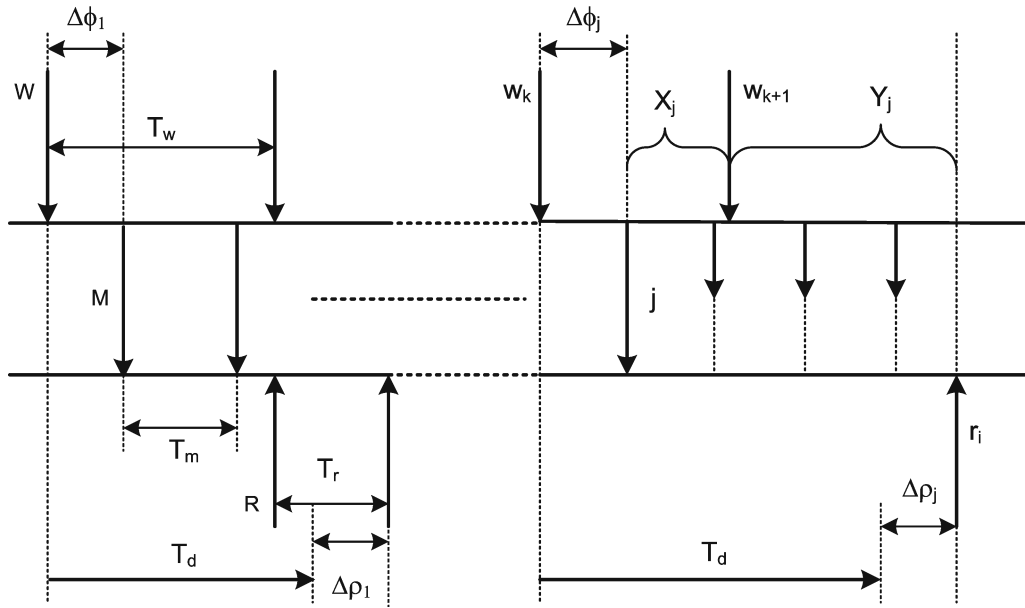


Figura 3.21 - Relações entre parâmetros temporais.

Nos pressupostos assumidos inicialmente, o modo de avaria da rede foi definido como decorrente da violação das *deadlines* associadas às mensagens transmitidas. Embora esta seja a forma clássica de abordar o problema, o modelo proposto nesta secção para a transmissão das variáveis permite estender este conceito de forma mais realista. Como o modelo já inclui alguns aspectos relacionados com o funcionamento real da aplicação distribuída que controla o processo, tem toda a lógica que o modo de avaria da rede seja definido a partir deste modelo. Neste sentido, a definição do modo de avaria da rede é reformulada, sendo agora definida em função da violação do período de validade T_d das variáveis transmitidas.

Como esta violação pode ocorrer segundo perspectivas diferentes, foram considerados dois cenários:

1. Uma tarefa utiliza uma variável cuja validade temporal (*deadline*)⁴⁴ já expirou. Esta condição é a mais realista, dado que admite que a variável pode ser utilizada em qualquer instante entre duas leituras;
2. Uma tarefa lê uma variável cuja validade temporal já expirou. É um cenário mais restrito do que o anterior, dado que admite que a variável só é utilizada no momento da sua leitura.

As características do protocolo de comunicações impõem relações particulares aos parâmetros temporais definidos no modelo Produtor-Consumidor, que podem ser periódicas, esporádicas (o tempo mínimo entre ocorrências é conhecido) ou totalmente estocásticas, implicando assim diferentes modos de funcionamento.

⁴⁴ Por uma questão de simplificação da apresentação a validade temporal das variáveis será referida como a respectiva *deadline*.

Neste sentido, nas próximas secções serão propostos vários modelos que definem as condições de avaria da rede em função destes modos de funcionamento. Para além dos pressupostos definidos previamente, foram adicionalmente considerados os seguintes para o desenvolvimento destes modelos:

- ♦ Assume-se que na total ausência de falhas durante as comunicações a rede opera de forma normal, sem quaisquer violação das *deadlines* associadas às variáveis;
- ♦ Tendo em conta que os efeitos das falhas nas mensagens (variáveis) podem ser consideradas independentes, a análise foca-se apenas na transmissão de uma variável.

O assumir deste pressuposto neste contexto, deve ser analisado com algum cuidado. Nos modelos obtidos na secção anterior foi utilizada uma abstracção em que o comportamento temporal não era considerado. Contudo, nos modelos propostos nesta secção a situação é totalmente oposta (o tempo aqui é importante). É assim necessário avaliar até que ponto os resultados anteriores podem ser utilizados nos modelos que definem a avaria da rede.

É fácil de verificar neste novo contexto que os padrões (sequências de procedimentos + aspectos temporais) associados à transmissão das mensagens não podem ser considerados independente entre si, excepto talvez em algumas redes que função das características do seu protocolo garantem à partida esta independência (ex. WorldFIP, Interbus-S e AS-I). Desta forma, o pressuposto assumido tem que ser encarado como uma aproximação com o objectivo de permitir desenvolver um modelo preliminar.

Numa primeira impressão, esta aproximação pode ser considerada um pouco grosseira. Contudo, uma análise do comportamento da rede em situações de funcionamento normal permite demonstrar que os erros cometidos podem na prática ser reduzidos. Assim, se for assumido que o tráfego da rede não possui uma taxa de utilização do meio de transmissão muito elevada, o que corresponde à maioria das situações, é razoável admitir que os padrões sejam aproximadamente constantes e com poucas interferências entre si. Ou seja, podem ser considerados aproximadamente independentes. Além disso, como se está interessado no comportamento estocástico em regime estacionário (e não numa abordagem do tipo pior caso), este acaba por reflectir o comportamento dominante, ou seja, o da (quase) independência entre padrões. Por fim, como o principal objectivo dos modelos é serem utilizados em avaliações comparativas, os erros cometidos são de certa forma compensados numa análise deste tipo.

Está-se assim perante um compromisso entre a possibilidade de obter um modelo analítico e preliminar, e o rigor com que o modelo é desenvolvido. Contudo e em resultado da discussão anterior, é razoável admitir que os erros

cometidos podem-se considerar perfeitamente aceitáveis face ao contexto da avaliação;

- ♦ As variáveis são escritas T_w e lidas T_r periodicamente. Este comportamento corresponde ao funcionamento clássico das tarefas que são executadas nos nós;
- ♦ Assume-se que a primeira transmissão de cada variável tem sempre sucesso. Este aspecto é apenas necessário para definir um estado a partir do qual se possa assumir com clareza que a rede não está num estado de avaria;
- ♦ A transmissão com sucesso de uma variável (mensagem) pode incluir eventuais retransmissões da respectiva mensagem devido à ocorrência de erros. Neste caso, assume-se que estas retransmissões ocorrem num intervalo de tempo muito curto e adjacente à primeira tentativa de transmissão, podendo assim ser considerado nulo, face aos restantes parâmetros temporais. Este pressuposto é também uma aproximação em relação ao comportamento real. Contudo, na maioria dos protocolos de comunicação das redes de campo este pressuposto pode ser considerado credível;
- ♦ Não são feitas quaisquer considerações sobre a existência de eventuais mecanismos que permitam às tarefas determinar se as *deadlines* das tarefas forma violadas ou não. Assume-se assim a pior situação;
- ♦ O tempo de propagação das tramas no meio de transmissão é considerado desprezável face aos restantes parâmetros temporais.

3.5.3.2.1 Transmissão Periódica de Mensagens

Este tipo de transmissão corresponde ao modo de funcionamento mais comum de uma rede de campo. Neste caso, o período de transmissão das mensagens T_m é constante e as fases iniciais $\Delta\phi_i$ e $\Delta\rho_i$, entre os vários parâmetros temporais podem ou não ser conhecidas (Fig. 3.21). No primeiro caso, é considerado que o funcionamento da rede está inicialmente sincronizado.

Sem Sincronização Inicial

De acordo com a Fig. 3.21, admita-se que j corresponde à j -ésima transmissão da mensagem (variável) de interesse. Sem perda de generalidade, assumase que esta transmissão se realiza imediatamente a seguir a uma escrita w_k e que foi realizada com sucesso. Posteriormente a j , mas antes da próxima escrita w_{k+1} , é possível que ocorram outras transmissões (só depende das relações entre os vários parâmetros temporais). No entanto, o seu sucesso é irrelevante para a análise do problema dado que estas transmissões não contribuem para estender a *deadline* da última transmissão (j) com sucesso.

O número de mensagens X_j transmitidas neste intervalo é

$$X_j = \left\lfloor \frac{T_w - \Delta\phi_j}{T_m} \right\rfloor \quad (3.42)$$

em que $\Delta\phi_j$ é a fase entre a última escrita com sucesso w_k e a j -ésima transmissão⁴⁵. A existência de transmissões com sucesso imediatamente a seguir a w_{k+1} é equivalente à situação anterior, e desta forma não será considerada. Se posteriormente a w_{k+1} todas as transmissões forem sem sucesso por um período de tempo que permita a violação da *deadline* associada à variável, então ocorre uma avaria da rede. Para que este cenário se verifique, o número de mensagens transmitidas sem sucesso Y_j é para o primeiro modo de avaria

$$Y_j = \left\lfloor \frac{T_d - \Delta\phi_j}{T_m} \right\rfloor - X_j \quad (3.43)$$

e para o segundo modo de avaria

$$Y_j = \left\lfloor \frac{T_d + \Delta\rho_j - \Delta\phi_j}{T_m} \right\rfloor - X_j \quad (3.44)$$

em que $\Delta\rho_j$ é a fase entre o fim da *deadline* T_d da última mensagem transmitida com sucesso ($w_k \rightarrow j \rightarrow T_d$) e a primeira leitura r_i imediatamente posterior.

É assim possível concluir, que a avaria da rede está relacionada com a ocorrência de uma sequência de transmissões de mensagens do tipo (X_j, Y_j) , após uma transmissão com sucesso.

As expressões deduzidas dependem dos parâmetros $\Delta\phi_j$ e $\Delta\rho_j$ cujos valores não são conhecidos à partida, mas sobre os quais é possível definir algumas características:

- ♦ No momento em que a operação da rede é iniciada, e admitindo uma ausência de sincronização inicial entre os vários períodos (T_w , T_m e T_r), as fases iniciais $\Delta\phi_1$ e $\Delta\rho_1$ que se estabelecem não são conhecidas à partida;
- ♦ Uma vez definidas as fases iniciais, o funcionamento global é periódico, porque T_w , T_m e T_r são periódicos. Deste facto resulta que $\Delta\phi_j$ e $\Delta\rho_j$ tomam apenas um conjunto de valores discretos. Para cada transmissão j , estes valores podem ser obtidos conhecendo as fases iniciais $\Delta\phi_1$ e $\Delta\rho_1$, os

⁴⁵ É assumido que a transmissão de uma mensagem se refere sempre à anterior escrita. Não se considera assim possível a escrita e transmissão simultânea referente à mesma variável. Um argumento similar é utilizado para a leitura.

valores T_w , T_m , T_r e T_d e o número de transmissões j decorridas desde o início;

- ♦ Em resultado do ponto anterior, o conjunto valores que $\Delta\phi_j$ e $\Delta\rho_j$ podem tomar é finito e periódico.

Tendo em conta o exposto, a determinação dos valores de $\Delta\phi_j$ e $\Delta\rho_j$, e consequentemente de X_j e Y_j , seria relativamente simples de realizar, contudo a ausência de conhecimento sobre os valores de $\Delta\phi_1$ e $\Delta\rho_1$ impede a respectiva determinação. Desta forma, é necessário considerar uma abordagem alternativa para resolver este problema.

Uma vez que as fases iniciais são desconhecidas, é perfeitamente razoável assumir que estas são variáveis aleatórias. Ainda no mesmo sentido, será razoável também assumir que $\Delta\phi_1$ e $\Delta\rho_1$ possam tomar qualquer valor (dentro dos respectivos intervalos) com igual probabilidade

$$\begin{aligned}\Delta\phi_1 &\in] 0...T_w] \\ \Delta\rho_1 &\in [0..T_r[\end{aligned} \tag{3.45}$$

Ou seja, $\Delta\phi_1$ e $\Delta\rho_1$ são variáveis aleatórias uniformemente distribuídas. Este pressuposto, implica também que $\Delta\phi_j$ e $\Delta\rho_j$ sejam variáveis aleatórias, e que também sejam uniformemente distribuídas (estão relacionadas por uma constante).

Decorre da exposição anterior que X_j e Y_j são também variáveis aleatórias, embora não seja possível à partida definir quais as respectivas funções distribuição. No entanto, utilizando as relações anteriores é possível verificar que estas funções distribuição são discretas. Esta conclusão pode ser deduzida a partir das funções distribuição conjunta $X_j \sim F(\Delta\phi_j, \Delta\rho_j)$ e $Y_j \sim F(\Delta\phi_j, \Delta\rho_j)$ através da determinação da respectiva evolução. Em termos práticos isto corresponde a traçar um gráfico bidimensional com eixos $\Delta\phi_j$ e $\Delta\rho_j$ e utilizar as expressões (3.42), (3.43) e (3.44) para calcular X_j e Y_j . Neste gráfico vão existir um conjunto de áreas para as quais X_j e Y_j são constantes devido às características das respectivas expressões. Tendo em conta que $\Delta\phi_j$ e $\Delta\rho_j$ são variáveis aleatórias uniformemente distribuídas, resulta de forma imediata que as áreas (em termos relativos) acima citadas vão corresponder directamente às probabilidades de X_j e Y_j .

Da discussão anterior é assim possível concluir que:

- ♦ Não é necessário considerar as fases iniciais, dado que o comportamento destas se encontra como que “condensado” no comportamento de $\Delta\phi_j$ e $\Delta\rho_j$. Isto resulta do facto de se considerar que o comportamento destas variáveis aleatórias é uniformemente distribuído. Desta forma, não é

necessário considerar o passado, restringindo-se a análise às condições existentes após uma transmissão de uma mensagem com sucesso. A notação fica assim simplificada para

$$\begin{aligned} \Delta \phi_j &\rightarrow \Delta \phi & X_j &\rightarrow X \\ \Delta \rho_j &\rightarrow \Delta \rho & Y_j &\rightarrow Y \end{aligned} \quad (3.46)$$

- ♦ Após a transmissão de uma mensagem com sucesso, existe uma probabilidade $Prob(S_k)$ de ocorrer uma sequência $S_k = \{X, Y\}$ que dê origem a uma avaria da rede. O número de sequências existentes é, teoricamente $\# \{X\} \cdot \# \{Y\}$, contudo nem todas as sequências podem existir na prática. A probabilidade associada cada sequência pode ser obtida da distribuição conjunta $(X, Y) \sim F(\Delta \phi, \Delta \rho)$, que por sua vez pode ser obtida das funções distribuição de $\Delta \phi$ e $\Delta \rho$.

Esta função distribuição é obtida de forma análoga à descrita anteriormente, embora considerando que as áreas relativas correspondem à probabilidade da sequência (X, Y) (Fig. 3.22);

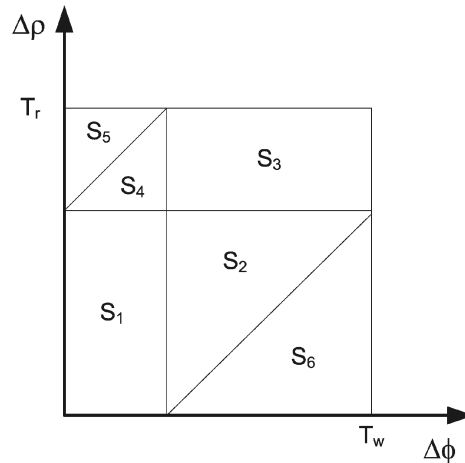


Figura 3.22 - Exemplo da determinação das probabilidades das sequências.

- ♦ Sempre que existe uma transmissão com sucesso, é reposto o estado de funcionamento normal da rede.

Desta forma, o processo estocástico associado à evolução das sequências temporais pode ser modelado por uma cadeia de Markov Discreta definida nos instantes de transmissão. Embora a cadeia assuma um tempo discreto, na prática o intervalo de tempo entre estes instantes corresponde ao período T_m . O valor das probabilidades de transição da cadeia, vão depender das probabilidades de transmissão com sucesso das mensagens e das probabilidades de ocorrência das sequências que levam à avaria da rede. O exemplo seguinte permite demonstrar a respectiva aplicação.

Exemplo

As características associadas à operação da rede são as seguintes:

- ♦ As mensagens são transmitidas com probabilidade de sucesso $Prob_{MS}$. Este valor pode ser determinado a partir dos vários modelos definidos nas secções anteriores. Podem assim incluir todo o tipo de aspectos referidos nestes modelos (meios de transmissão redundantes, avarias em modo-comum, número de tramas de dados e controlo, retransmissões, etc.)
- ♦ Existem três sequências possíveis para a avaria da rede: $S_1=\{1,1\}$, $S_2=\{0,2\}$ e $S_3=\{1,2\}$ com probabilidades de ocorrência P_{S1} , P_{S2} e P_{S3} respectivamente.

A definição da respectiva cadeia de Markov (Fig. 3.23) pode ser realizada de forma simples:

- ♦ O estado **OK** representa o funcionamento normal da rede, ou seja, foi transmitida uma mensagem com sucesso;
- ♦ Os estados X_i representam a sequência de X_i transmissões posteriores a uma transmissão com sucesso, cujo sucesso é indiferente;
- ♦ Os estados Y_i representam a sequência de Y_i transmissões que não tiveram sucesso, posteriores a uma transmissão com sucesso e a X_i transmissões cujo sucesso é indiferente;
- ♦ O estado **A** representa a avaria da rede;
- ♦ Após uma transmissão com sucesso (estado **OK**) pode ocorrer uma sequência do tipo $S_i=\{X_i, Y_i\}$, com probabilidade P_{Si} .

A partir da resolução da cadeia de Markov (conjunto com todas as sequências) é possível obter os seguintes medidas da confiança no funcionamento:

- ♦ A disponibilidade A é obtida como:

$$A = Prob(OK) + \sum_{i,k} Prob(X_{ik}) + \sum_{i,k} Prob(Y_{ik}) \quad (3.47)$$

em que $Prob(j)$ representa a probabilidade de ocupar o estado j ;

- ♦ O $MTTF$ é obtido considerando que o estado **A** é absorvente e calculando o respectivo tempo de absorção;
- ♦ O $MTBF$ é obtido como $MTBF = MUT + MDT$, em que o MUT é o tempo médio de ocupação dos estados **OK**, X_{ik} e Y_{ik} , e o MDT é o tempo médio de ocupação do estado **A**.

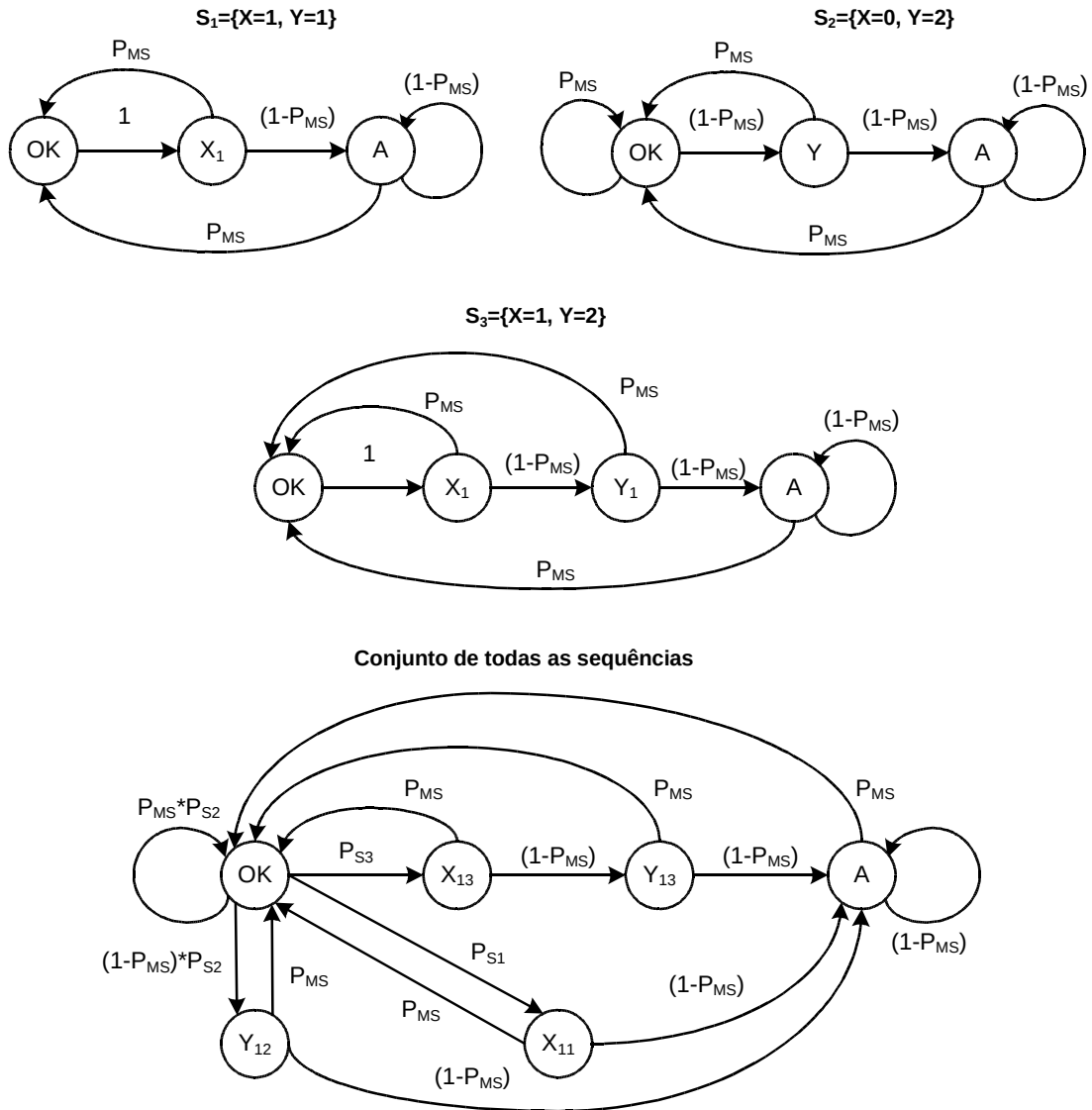


Figura 3.23 - Exemplo sem sincronização inicial.

Com Sincronização Inicial

A existência de sincronização inicial garante que as fases iniciais, $\Delta\phi_1$ e $\Delta\rho_1$, são conhecidas previamente.

Desta forma, para cada transmissão j , $\Delta\phi_j$ e $\Delta\rho_j$ podem ser obtidas a partir das fases iniciais ($\Delta\phi_1$ e $\Delta\rho_1$), dos períodos (T_w , T_m , T_r e T_d) e do número de transmissões j . Como o comportamento é periódico, $\Delta\phi_j$ e $\Delta\rho_j$ também são periódicas, e número de possíveis sequências $S_k = \{X, Y\}$ que levam à avaria da rede é finita. Desta forma, é apenas necessário considerar um grupo inicial de N transmissões, com $N = MMC\{T_w, T_m, T_r\}$ ⁴⁶, e para cada uma destas transmissões determinar a correspondente sequência S_k , $k = \{1 \dots N\}$ de avaria da rede.

46 MMC: Menor Múltiplo Comum

Note-se contudo que ao contrário do modelo anterior, não é necessário determinar as probabilidades de ocorrência das sequências porque o comportamento agora é determinístico, sendo definido pela ordem temporal em que as transmissões ocorrem.

O processo estocástico associado à evolução das sequências temporais pode ser também modelado por uma cadeia de Markov Discreta definida nos instantes de transmissão. O exemplo seguinte permite demonstrar a respectiva aplicação.

Exemplo

As características associadas à operação da rede são as seguintes (Fig. 3.24):

- ◆ As mensagens são transmitidas com probabilidade de sucesso $Prob_{MS}$;
- ◆ A sequência de transmissão das mensagens é: $j_1 \rightarrow j_2 \rightarrow j_3 \rightarrow j_1 \dots$, com as respectivas sequências de falhas $S_1=\{0,1\}$, $S_2=\{0,1\}$ e $S_3=\{1,1\}$;
- ◆ A resolução da cadeia de Markov ⁴⁷, e a obtenção das várias medidas da confiança no funcionamento são realizadas de forma idêntica ao exemplo anterior.

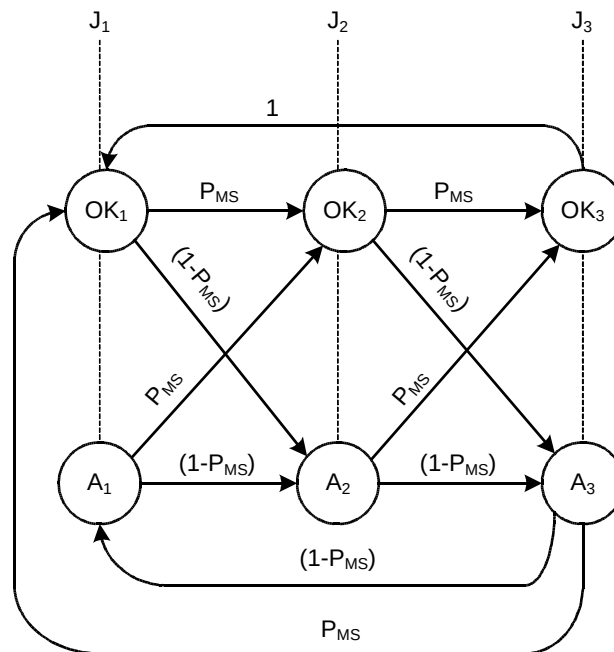


Figura 3.24 - Exemplo com sincronização inicial.

3.5.3.2.2 Transmissão Aperiódica de Mensagens

Neste cenário a transmissão das mensagens deixa de ser realizada periodicamente, e o parâmetro T_m pode ser considerado uma variável aleatória. A obtenção de expressões analíticas (caso possam ser obtidas) para esta situação é uma tarefa

⁴⁷ Os estados OK_i e A_i , têm interpretações equivalentes ao exemplo anterior.

muito complexa, devido à interacção entre processos determinísticos (T_w e T_r) e estocásticos (T_m). Assim e no sentido dos objectivos propostos neste trabalho, este problema foi abordado através de uma aproximação.

Com o objectivo de obter um modelo analítico, será utilizada uma aproximação conservativa. Embora o comportamento de T_m seja estocástico, em condições reais é sempre possível minorar ou majorar o respectivo valor. Com esta informação, e utilizando os modelos definidos nas secções anteriores, são obtidas as sequências de avaria para as piores situações (*worst case*). Se se utilizar o valor mais pequeno de T_m , isto vai resultar em sequências com muitas transmissões, o que por sua vez implica uma baixa probabilidade de avaria (é necessário que existam muitas mensagem consecutivas com erros). Aplicando-se o raciocínio inverso, obtêm-se valores mais elevados. Esta análise conduz assim a limites máximos e mínimos para a probabilidade de avaria.

É possível obter uma variante do modelo anterior mais precisa à custa de um maior esforço computacional. Se as operações de leitura não forem tomadas em conta ($\Delta\rho=0$), uma avaria da rede ocorre se existir uma sequência do tipo (X, Y) num intervalo de tempo T_d após a escrita da variável. Se a função distribuição de T_m for conhecida ou estimada, é possível calcular a probabilidade associada a cada sequência (ver expressões 3.48 e 3.49). Dado que o número de sequências pode ser muito elevado, é razoável considerar apenas as mais prováveis. Utilizando esta informação é possível construir um processo de Renovação de Markov, a partir do qual é possível obter as medidas da confiança no funcionamento pretendidas. A vantagem deste modelo reside no facto de incorporar correctamente o comportamento estocástico associado a T_m . É interessante notar que se se considerar que a avaria ocorre posteriormente a um intervalo de duração T_d+T_r , os resultados obtidos são optimistas. Desta forma, o valor real situa-se entre ambos os casos.

Por fim existe um caso particular que merece ser analisado. Admita-se que a variável é produzida com um período muito grande e cuja fase até à respectiva transmissão $\Delta\phi_j$ não é conhecida à partida, mas que pode ser majorada. De acordo com a discussão anterior, é possível desenvolver um modelo analítico conservativo baseado nas seguintes condições:

- ♦ T_w é uma variável aleatória⁴⁸ com função distribuição F_{Tw} ;
- ♦ $F_{Tw}(T_d)=0$, ou seja T_w é sempre maior que T_d ;
- ♦ T_m é uma variável aleatória com função distribuição F_{Tm} .

⁴⁸ No caso de ser periódico a função distribuição é um Dirac.

Se as operações de leitura não forem consideradas, uma avaria ocorre quando nenhuma mensagem for transmitida com sucesso num intervalo de tempo T_d após a produção da variável. Neste caso a probabilidade de avaria $Prob_A$ é dada por

$$Prob_A = \sum_{n=1}^{+\infty} Prob\{N_{Tm}(T_d)=n\} \cdot (1-Prob_{MS})^n \quad (3.48)$$

em que $N_{Tm}(T_d)$ é o número de mensagens transmitidas durante um intervalo de tempo de duração T_d e $Prob\{N_{Tm}(T_d)=n\}$ a respectiva probabilidade. Este último valor pode ser obtido como

$$Prob\{N_{Tm}(T_d)=n\} = F_{Tm}^n(T_d) - F_{Tm}^{n+1}(T_d) \quad (3.49)$$

em que F_{Tm}^n é a n -ésima convolução de F_{Tm} consigo própria. Se a fase inicial tem uma distribuição conhecida (ou estimada) também é possível inclui-la facilmente no modelo. A resolução analítica directa de (3.49) é uma tarefa complexa, contudo utilizando a transformada de Laplace ou métodos numéricos é possível minimizar a complexidade da mesma.

3.5.4 Caso de Estudo

Com o objectivo de validar a metodologia proposta foi definido um caso de estudo.

A partir dos modelos apresentados nas secções anteriores é possível verificar que as medidas da confiança no funcionamento estão muito dependentes de um conjunto de parâmetros associados à operação da rede. Desta forma, uma primeira abordagem passaria por analisar a sensibilidade dos várias medidas em relação a estes mesmos parâmetros. No entanto esta abordagem implica uma série de dificuldades. Existem relações de interdependência entre alguns parâmetros (ex. garantia de um funcionamento correcto) que implicam limitações nos seus valores, e além disso os processos físicos controlados implicam uma gama muito grande de valores para os mesmos (ex. podem ir de μs até dezenas de horas).

Assim, considerou-se mais lógico abordar um caso típico, considerando que os parâmetros menos importantes são fixos e procedendo uma análise de sensibilidade em relação aos restantes. Dos vários parâmetros operacionais da rede, o período de de transmissão das mensagens T_m destaca-se, quer pelo papel que desempenha na definição das sequências associadas às avarias, quer pela flexibilidade de valores que pode tomar sem comprometer o funcionamento normal da rede. É assim o parâmetro mais adequado para realizar uma análise de sensibilidade.

Para o caso de estudo proposto foram definidos um conjunto de valores representativos de um controlo típico de um processo:

- ♦ Transmissão periódica de mensagens sem sincronização inicial;
- ♦ Modo de avaria por utilização de uma variável inválida;
- ♦ $T_w = 100\text{ ms}$;
- ♦ $T_m = [50, 390]\text{ ms}$;
- ♦ $T_d = 500\text{ ms}$
- ♦ $T_r = 500\text{ ms}$;
- ♦ Possibilidade de utilização de um meio de transmissão simples (sem redundância), ou dual (com taxas de erros iguais) com $\beta = \{0.01, 0.05, 0.2\}$ e $Prob_{MS} = \{0.9, 0.99, 0.999\}$.

Foram escolhidas deliberadamente probabilidades conservativas (mensagens e falhas em modo-comum), com o objectivo de obter medidas pessimistas (como os valores absolutos são muito pequenos torna-se mais simples a comparação). As medidas obtidas estão relacionadas com a disponibilidade e a fiabilidade.

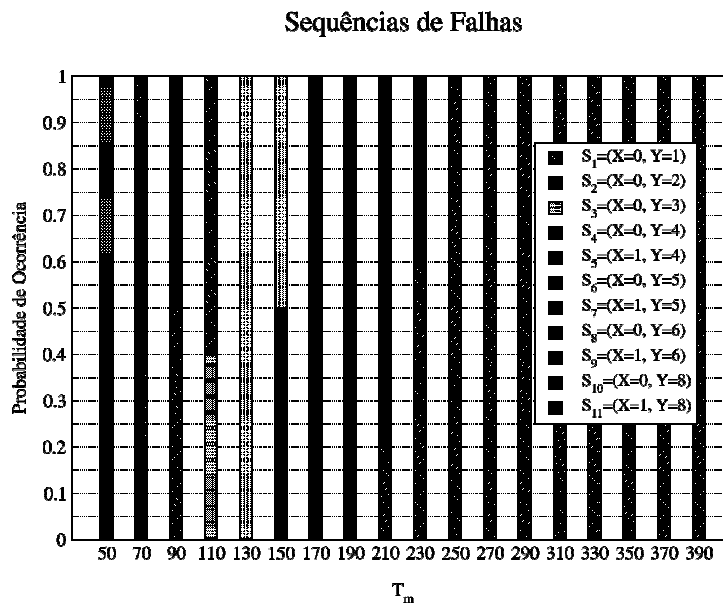


Figura 3.25 - Probabilidades das sequências das falhas.

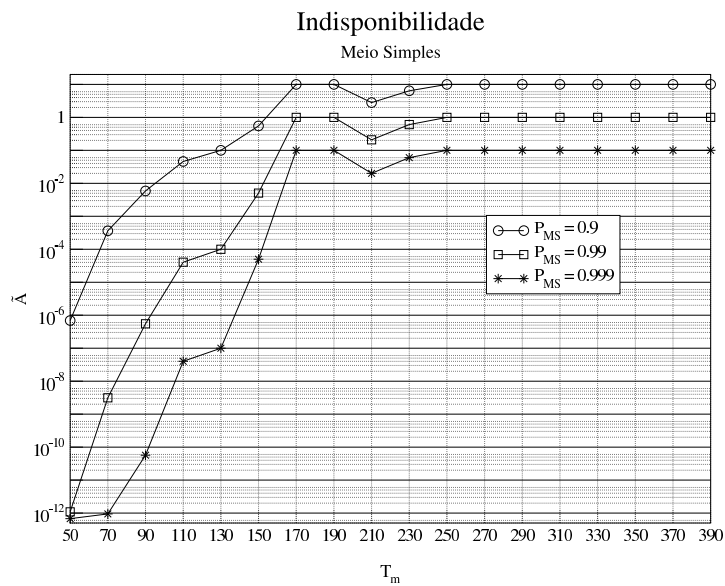


Figura 3.26 - Indisponibilidade: meio simples (não redundante).

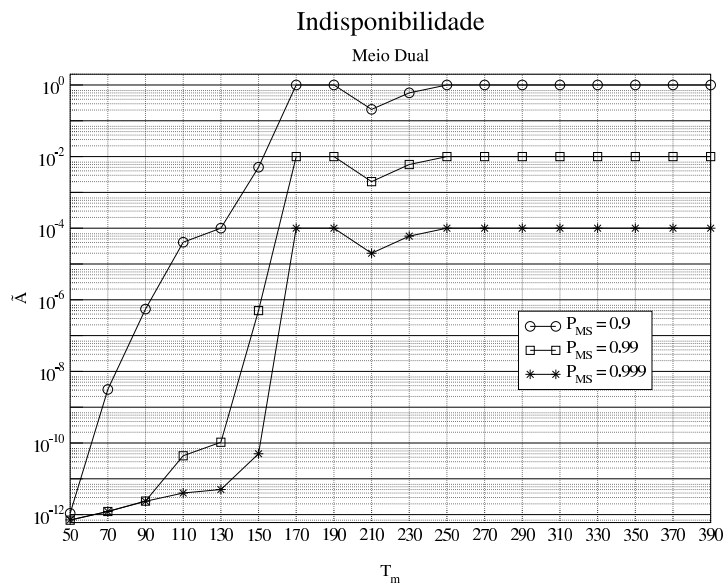


Figura 3.27 - Indisponibilidade: meio dual.

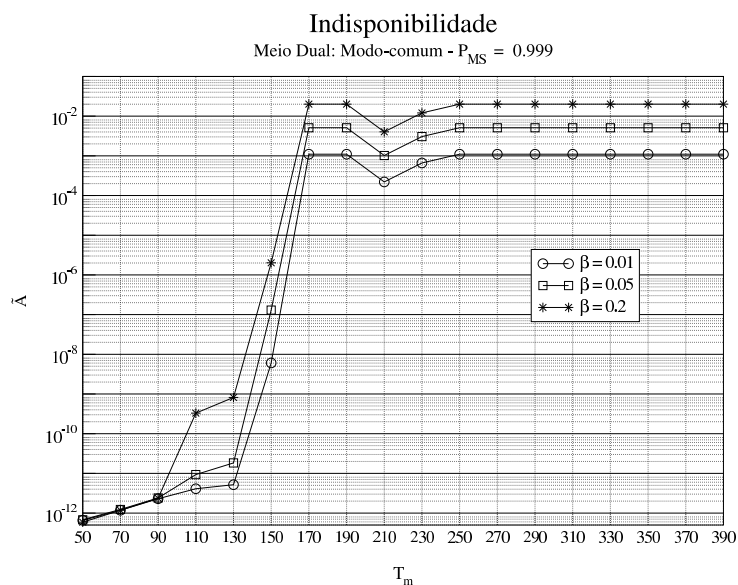


Figura 3.28 - Indisponibilidade: meio dual – falhas em modo-comum.

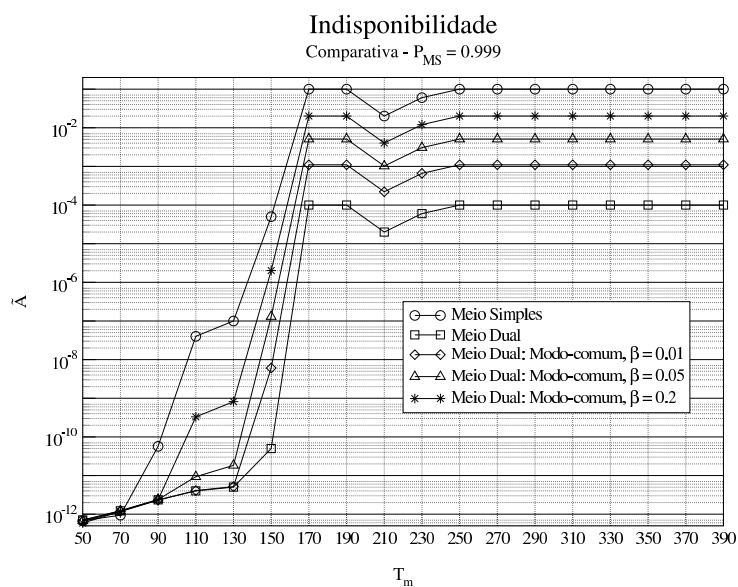


Figura 3.29 - Indisponibilidade comparativa.

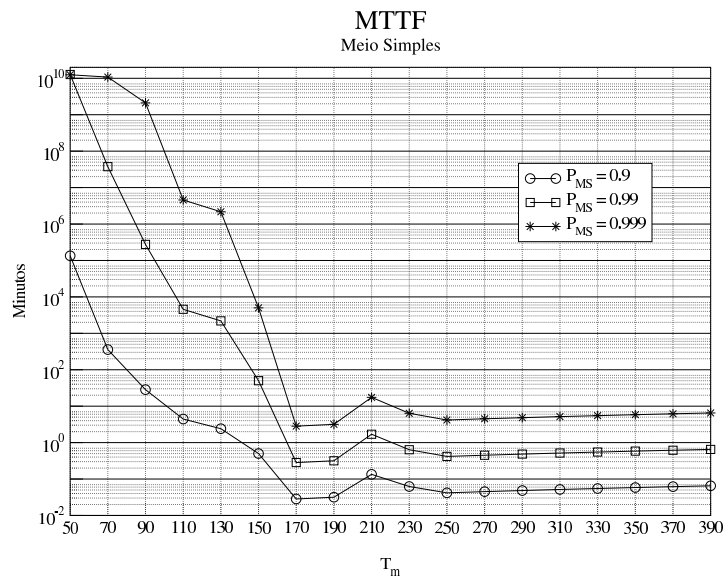


Figura 3.30 - MTTF: meio simples.

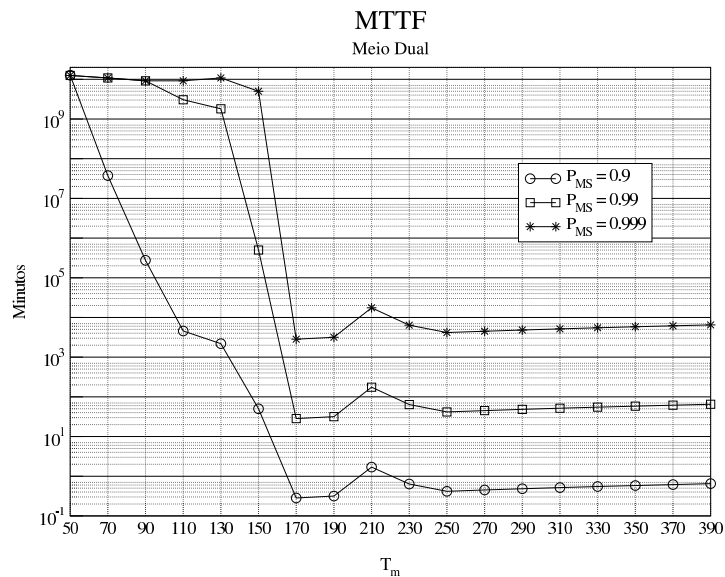


Figura 3.31 - MTTF: meio dual.

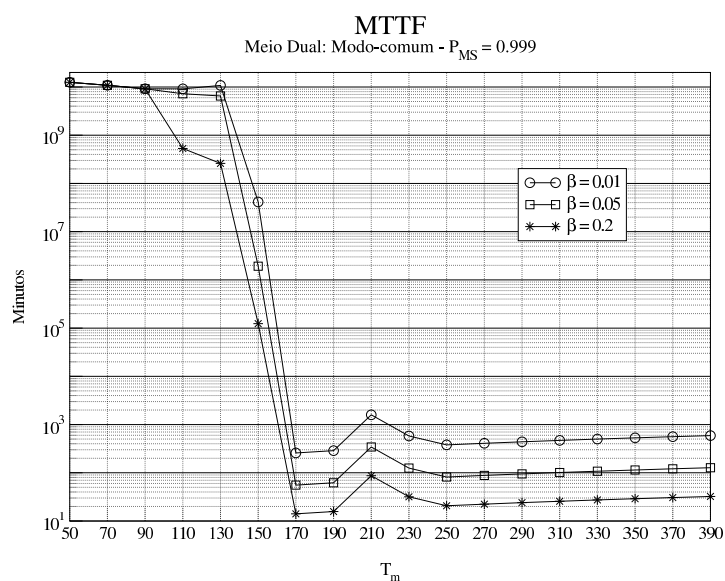


Figura 3.32 - MTTF: meio dual – falhas em modo-comum.

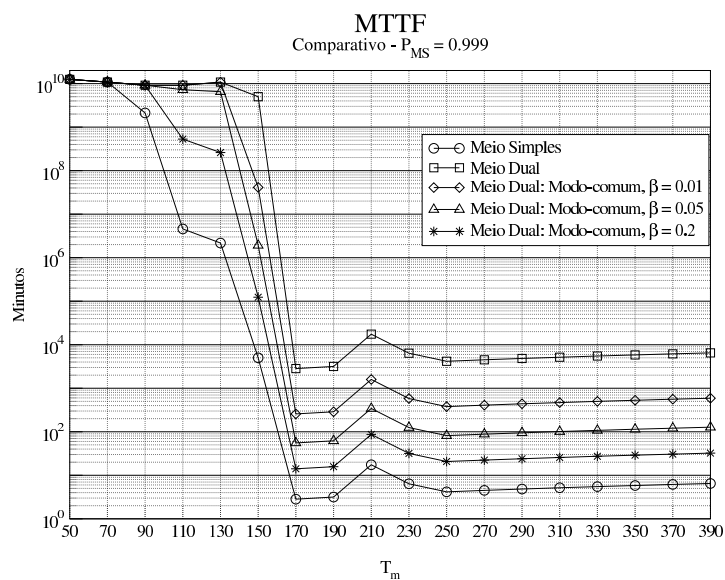


Figura 3.33 - MTTF comparativo.

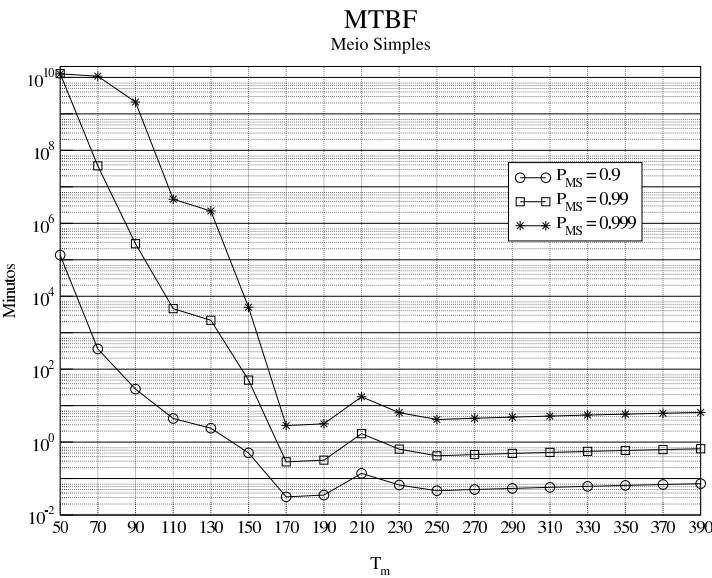


Figura 3.34 - MTBF: meio simples.

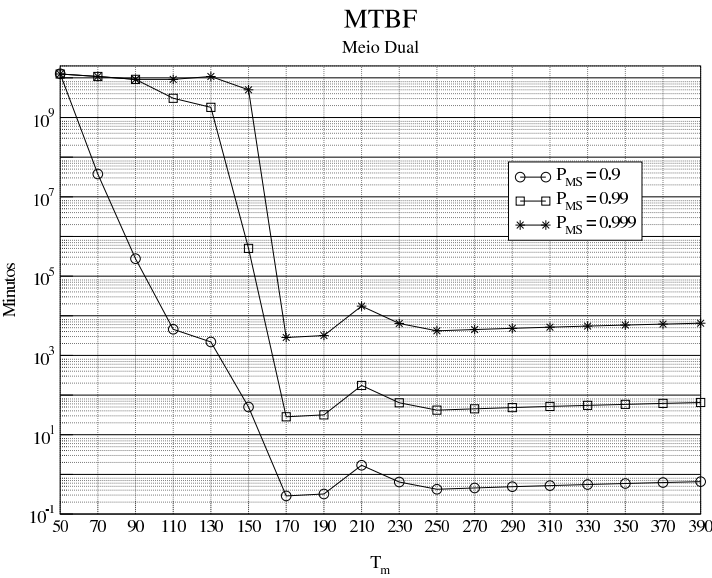


Figura 3.35 - MTBF: meio dual.

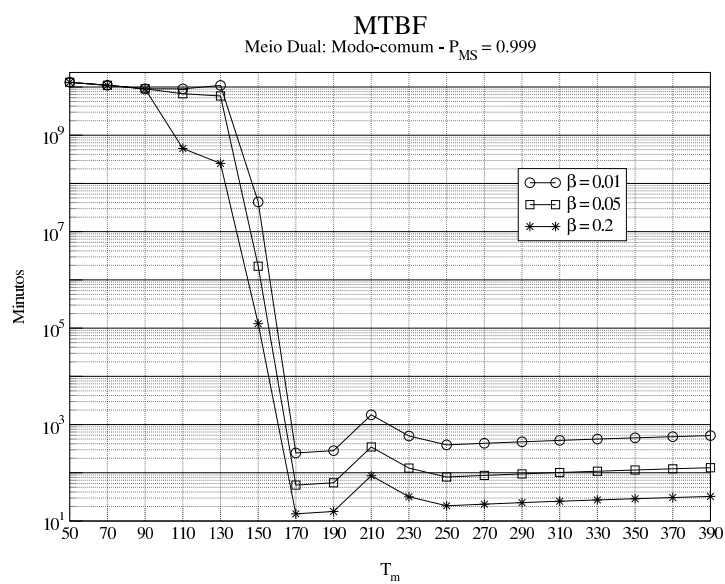


Figura 3.36 - MTBF: meio dual – falhas em modo-comum.

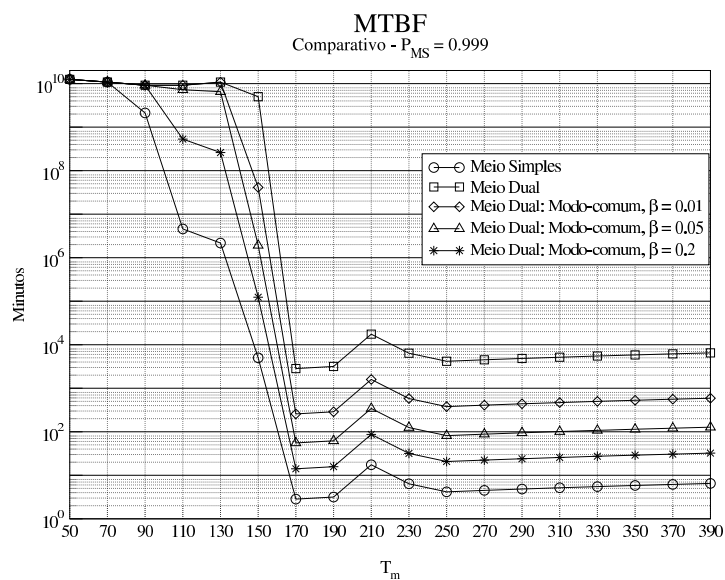


Figura 3.37 - MTBF comparativo.

3.5.4.1 Avaliação de Resultados

A partir dos resultados obtidos é possível estabelecer as seguintes conclusões

- ♦ O parâmetro T_m influi fortemente na definição das sequências que originam a avaria da rede (Fig. 3.25), condicionando de forma directa as medidas obtidas;
- ♦ Embora os valores assumidos para a probabilidade de sucesso fossem bastante conservativos, a indisponibilidade é na maioria das situações muito reduzida (Figs. 3.26, 3.27, 3.28, 3.29). A indisponibilidade depende, como seria de esperar, do valor de T_m dado que este último define as sequências que levam à avaria da rede. Esta dependência é especialmente importante nas situações em que uma pequena alteração em T_m origina uma alteração importante na definição das sequências de falhas, como é o caso de $T_m \in [50 \dots 170]$. Quando as alterações das sequências de falhas são ligeiras, ou mesmo nulas, como é caso de $T_m \in [230 \dots 390]$, a indisponibilidade é aproximadamente constante. Desta forma, é possível concluir que a alteração de T_m por si só é suficiente para melhorar o comportamento do sistema;
- ♦ A introdução de uma rede dual melhora sem qualquer dúvida a qualidade de funcionamento da rede, com uma diminuição da indisponibilidade entre uma a três ordens de grandeza (Fig. 3.27, 3.29);
- ♦ A existência de falhas em modo-comum, mesmo em pequenas percentagens, afecta de forma importante a qualidade de funcionamento da rede dual, com uma diminuição da indisponibilidade entre uma a duas ordens de grandeza (Figs. 3.28, 3.29);
- ♦ Da mesma forma já observada para a indisponibilidade, o *MTTF* depende muito fortemente de T_m , estando esta dependência intimamente relacionada com as sequências que levam à avaria da rede (Figs. 3.30, 3.31, 3.32, 3.33). De forma geral, quanto maior o número de mensagens destas sequências maior é o *MTTF*. Nas situações em que existe uma grande variação de sequências, como é caso de $T_m \in [50 \dots 170]$, o comportamento é mais pronunciado. No entanto, nas situações onde as variações das sequências são ligeiras, ou mesmo nulas, como é o caso de $T_m \in [230 \dots 390]$, o *MTTF* tem um comportamento praticamente linear com T_m . Em quaisquer das situações é possível concluir que a alteração de T_m por si só é suficiente para melhorar qualidade de funcionamento da rede (Figs. 3.31, 3.33). Assim, numa rede simples o aumento de T_m de $150ms$ para $250ms$, reduz o *MTTF* de cerca de $67h$ para $4min$! A introdução de uma rede dual permite melhorar de forma muito significativa o comportamento da rede, com um aumento do *MTTF* entre uma a três ordens de grandeza;

- ♦ A existência de falhas em modo-comum, mesmo em pequenas percentagens, afecta de forma importante o comportamento da rede dual, com reduções do *MTTF* entre uma a duas ordens e meio de grandeza (Figs. 3.32, 3.33);
- ♦ O *MTBF* tem um comportamento quase idêntico ao *MTTF* dado que o *MDT* é muito reduzido (Figs. 3.34, 3.35, 3.36, 3.37). Este comportamento pode ser explicado a partir das cadeias de Markov associadas às sequências de falhas. Quando o sistema se encontra no estado de avaria (**A**), a probabilidade de continuar nesse estado é muito reduzida ($1 - Prob_{MS}$), além do que a probabilidade de retornar a esse estado é ainda muitíssimo mais reduzida, implicando assim tempos muito baixos de ocupação desse estado, com a consequente redução do *MDT*.

3.6 Conclusões

Neste capítulo foram desenvolvidos vários modelos com o objectivo de permitir uma avaliação preliminar da confiança no funcionamento de arquitecturas de controlo suportadas por redes de campo, das quais se dispõe de um conhecimento limitado quer do seu funcionamento, quer dos parâmetros requeridos para o desenvolvimento e a solução do modelo.

Os modelos de falhas permanentes propostos resultam da identificação dos componentes da rede cuja avaria, neste contexto, implicam as maiores consequências para o comportamento da rede. Função da sua estrutura genérica, estes modelos permitem através de um processo de composição a obtenção de modelos de avaliação para diferentes arquitecturas de controlo. Garante-se assim uma alargada gama de aplicabilidade destes modelos. Em paralelo, a possibilidade de extrair expressões analíticas a partir dos modelos utilizando o método dos estados de funcionamento crítico, facilita o próprio processo de avaliação. Desta forma, para além de não exigir a utilização de ferramentas de avaliação específicas, permite também identificar os parâmetros mais importantes ao nível dos modelos, e assim contribuir para a sua simplificação. O caso de estudo apresentado, embora focado num exemplo específico, permite concluir que os vários factores de cobertura e as falhas em modo-comum representam os parâmetros que mais influenciam o comportamento da rede.

O modelo de falhas transitórias resulta de um compromisso entre a possibilidade de dispor de um modelo analítico e o rigor com que o comportamento da rede neste contexto é modelado. O seu desenvolvimento assentou num aspecto central dos sistemas de controlo suportados por redes de campo, nomeadamente na existência de requisitos de tempo-real ao nível das mensagens transmitidas. O

modelo proposto, nas suas várias componentes, pode ser encarado segundo duas perspectivas. Ao nível da camada física, os modelos obtidos são rigorosos e bastante úteis na avaliação de vários aspectos, dos quais se salientam a duração das falhas e a redundância dos meios de transmissão, incluindo as avarias em modo-comum dos mesmos. No entanto e ao nível da camada de ligação de dados, os modelos resultam de uma aproximação, nomeadamente no que se refere ao funcionamento temporal do protocolo de comunicações. Contudo e em condições típicas de funcionamento da rede, as aproximações cometidas podem ser consideradas razoáveis tendo em conta o tipo de avaliação pretendida. É assim possível concluir que se estes modelos forem utilizados num contexto de avaliação comparativa entre diferentes soluções alternativas, os resultados obtidos são perfeitamente consistentes. O caso de estudo apresentado, embora também focado num exemplo muito específico, permite concluir que o período de transmissão das mensagens desempenha um papel chave na definição do comportamento da rede.

Face aos resultados obtidos com ambos os modelos, é possível concluir que estes cumprem os objectivos que foram propostos inicialmente.

Capítulo 4

Avaliação Integrada da Confiança no Funcionamento

4.1 Introdução

O modelos propostos no capítulo anterior tinham como objectivo principal permitir a realização de uma avaliação preliminar do comportamento de uma rede de campo em cenários de falhas permanentes e transitórias. Este objectivo traduziu-se pelo desenvolvimento de um conjunto de modelos analíticos que permitem realizar um avaliação rápida, eficiente e com um nível mínimo de rigor.

O carácter desta avaliação preliminar, nomeadamente o contexto em que esta foi realizada, implicaram algumas restrições na forma como os modelos foram desenvolvidos. Assim, a utilização de uma abstracção de alto nível resultou que os modelos possuísem um baixo nível de detalhe, enquanto que as limitações existentes ao nível dos dados sobre o funcionamento da rede e a ocorrência de falhas, obrigaram à utilização de pressupostos baseados em comportamentos simples. Além disso, a avaliação foi realizada como se ambos os cenários (falhas permanentes e transitórias) fossem independentes, quando na prática não o são. Ambos os tipos de falhas implicam relações de dependência ao nível das suas consequências, que necessitam de ser avaliadas em conjunto para se obter uma

perspectiva global dos seus efeitos. Embora os modelos propostos possuam um inegável interesse no contexto em que foram desenvolvidos, a sua utilização em cenários mais complexos fica imediatamente restringida pela natureza dos mesmos.

Torna-se assim evidente que é necessário dispor de modelos de avaliação com a capacidade de representar situações mais abrangentes, nomeadamente através da introdução no mesmo de diferentes perspectivas ao nível do comportamento da aplicação de controlo distribuída. Os modelos devem assim dispor um maior rigor na descrição funcional do comportamento da rede, incluindo uma caracterização mais próxima da realidade de ocorrência das falhas e das respectivas consequências ao nível das avarias. Além disso, devem também possuir um carácter genérico e adaptável, sem contudo comprometer o seu rigor e a eficácia da sua resolução, permitindo assim a sua utilização na avaliação da confiança no funcionamento de uma gama alargada de situações.

Este capítulo representa uma versão extendida dos trabalhos publicados pelo autor em [PA04a] [PA04b].

4.2 Definição do Problema

As redes de campo são utilizadas presentemente num grande número de domínios de aplicação, com características muito diversas e com diferentes níveis de exigência e de complexidade. Os seus componentes possuem também diferentes características, estão situados em diferentes níveis funcionais e possuem graus de interacção variados entre si. Além disso, o seu comportamento na presença de falhas é complexo com consequências de várias ordens, incluindo diferentes níveis de gravidade.

Perante este conjunto de factos, é fácil perceber que qualquer tipo de avaliação da confiança no funcionamento que procure incorporar de forma exaustiva todos estes aspectos num único modelo, torna-se praticamente impraticável, quer ao nível do processo de modelação, quer ao nível da resolução do modelo. No entanto, e como discutido no capítulo 1, o domínio de aplicações destes tipos de redes obriga, muitas das vezes, à necessidade de se dispor de um modelo de avaliação que tem de incluir muitos destes aspectos.

A complexidade deste problema levou a que os trabalhos que versam a avaliação da confiança no funcionamento se concentrassem apenas em aspectos específicos do sistema, na esperança de que a “soma” das várias contribuições permitisse obter uma avaliação completa. Embora esta seja uma alternativa possível, é fácil de

perceber que só é válida se for assumido um grau elevado de independência entre os vários subsistemas avaliados, o que na maioria dos casos não se verifica. Assim é frequente, que a utilização combinada dos resultados obtidos em cada uma destas avaliações independentes, dê origem a uma avaliação global que implique resultados quer optimistas, quer pessimistas, de acordo com a forma como os resultados parcelares são incorporados. Neste contexto, a situação ideal seria dispor de um modelo único que permitisse avaliar de forma global o comportamento da rede, garantindo assim resultados mais consistentes com a realidade.

A solução proposta neste capítulo para este problema, passa por desenvolver um modelo de avaliação integrado, que inclua de forma adequada diferentes perspectivas sobre o funcionamento da rede. Desta forma, são identificadas as características mais importantes no funcionamento da rede, que embora modeladas de forma detalhada e rigorosa, são incorporadas de forma especial num único modelo por forma a não limitar a exequibilidade da sua solução. Pretende-se assim combinar, ao nível do modelo, um elevado grau de detalhe e de rigor, com a capacidade de obter de forma eficaz a respectiva solução, nomeadamente em relação às medidas de interesse.

Face à complexidade deste problema foi necessário adoptar uma abordagem que garanta que as diferentes perspectivas sobre o funcionamento da rede são tomadas em conta, e que o modelo obtido seja rigoroso e com o nível adequado de detalhe, bem como resolúvel. A abordagem adoptada pode ser sistematizada nas seguintes fases:

1. Análise das características típicas dos sistemas de controlo suportados por redes de campo, incluindo a identificação de requisitos mínimos aos nível da medidas da confiança no funcionamento pretendidas;
2. Estudo dos vários trabalhos desenvolvidos em domínios de conhecimento associados à avaliação da confiança no funcionamento deste tipo de sistemas, com especial relevância para os que visam directamente a modelação;
3. Avaliação das capacidades e das restrições impostas pelo formalismo de modelação adoptado. Análise das técnicas de minimização dos efeitos das restrições;
4. Definição de uma metodologia para o desenvolvimento dos modelos, concretizada através da identificação das características e funcionalidades mais importantes a incorporar no modelo e na utilização de técnicas de modelação específicas;
5. Desenvolvimento do modelo pretendido.

Embora estas fases induzam, aparentemente, uma abordagem sequencial, na prática verificou-se que esta foi fortemente iterativa, nomeadamente entre as fases 3 e 5. Muitos dos objectivos definidos inicialmente necessitaram de ser posteriormente modificados devido às muitas restrições existentes, especialmente ao nível da modelação. Só após várias iterações é que foi possível definir quais as características do modelo de avaliação pretendido, qual técnica de modelação mais adequada e qual a metodologia mais correcta para o desenvolvimento do modelo.

Nas secções seguintes serão discutidas de forma pormenorizada as várias fases da abordagem utilizada.

4.2.1 Características dos Sistemas de Controlo Suportados por Redes de Campo

O conhecimento das condições típicas de funcionamento e de operação das redes⁴⁹ de campo, permite definir, através de diferentes perspectivas, quais as suas características estruturais e funcionais mais importantes:

- ♦ A rede é constituída por vários nós⁵⁰ dispostos geograficamente, que estão interligados através de uma infra-estrutura de comunicações que permite a troca de informações entre estes (Fig. 4.1);

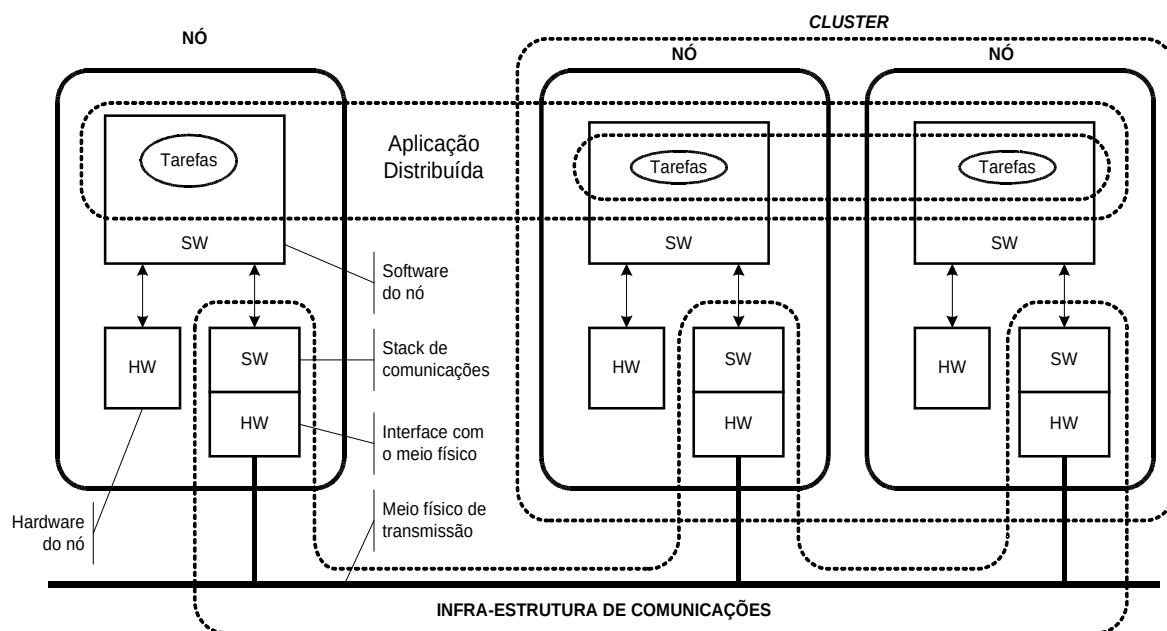


Figura 4.1 - Características estruturais e funcionais da rede campo.

⁴⁹ Tal como no capítulo anterior utilizar-se-à o termo *rede* para designar este tipo de sistemas de controlo.

⁵⁰ Assume-se a interpretação definida no capítulo anterior.

- ♦ A infra-estrutura de comunicações pode ser decomposta em três componentes: o meio físico de transmissão, as interfaces entre os nós e o meio de transmissão e a *stack* de comunicações. Esta última está localizada nos nós da rede e pode ser implementada por *hardware*, *software* ou ambos [HHK+03];
- ♦ A estrutura de cada nó assenta na interacção de dois conjuntos de componentes: *hardware* e *software*. Os componentes de *hardware* podem ser internos (ex. CPU, memória, etc.) ou externos (ex. sensores e actuadores). Ao nível do *software*, cada nó executa um conjunto de tarefas responsáveis pelo controlo do processo. Dada a natureza do tipo de processos controlados, é comum definirem-se requisitos de tempo-real ao nível da sua execução [AB90] [SK92] [LH94] [Kop97];
- ♦ O controlo global do processo é assegurado por intermédio de uma aplicação distribuída, constituída pelas várias tarefas que são executadas nos nós. Estas estão organizadas de acordo com uma estrutura funcional específica e adequada ao controlo do processo em causa. A troca de dados entre tarefas é assegurada pelo envio e recepção de mensagens utilizando a infra-estrutura de comunicações existente. Dada a natureza do tipo de processos controlados, é extremamente comum definirem-se requisitos de tempo-real ao nível das mensagens transmitidas [PD88] [KS94] [Tho98] [Tho99];
- ♦ O funcionamento da rede pode ser afectado pela ocorrência de falhas com características diversas e que afectam componentes de diferentes tipos. A rede incorpora mecanismos ao nível topológico, estrutural e funcional que lhe permitem tolerar algumas dessas falhas. Estes vão desde a simples redundância ao nível do *hardware*, até ao agrupamento de nós em *clusters*⁵¹ com o objectivo de formar grupos de nós tolerantes a falhas, passando pela inclusão ao nível do *software* de mecanismos que permitam, num cenário de falhas, tolerar e reconfigurar a rede a diferentes níveis;
- ♦ A ocorrência de avarias na rede pode manifestar-se de diferentes modos e com diferentes níveis de gravidade. Em relação a este último aspecto, as avarias podem colocar em causa a segurança (*safety*) de pessoas e bens;
- ♦ Após a avaria de componentes da rede ou da própria rede, é possível proceder a trabalhos de reparação dos mesmos por forma a coloca-los no seu estado operacional. As reparações podem variar em tempo, forma e número de componentes reparados;
- ♦ As medidas de confiança de funcionamento que interessa obter estão associadas à fiabilidade, disponibilidade e segurança. Estas medidas podem ser obtidas quer em regime estacionário, quer transitório.

⁵¹ Optou-se por utilizar a terminologia em Inglês por se julgar mais adequada ao contexto.

4.2.2 Discussão dos Trabalhos Relevantes Previamente Desenvolvidos

A importância com que se reveste a avaliação da confiança no funcionamento deste tipo de sistemas, motivou ao longo das últimas décadas a comunidade científica para a criação de várias linhas de investigação com o objectivo de fornecer respostas adequadas aos mesmos. Contudo, e devido à complexidade dos problemas, a estratégia usualmente utilizada foi do tipo “dividir para conquistar”, ou seja, as soluções propostas focaram-se quase exclusivamente apenas numa parte do problema.

As redes de campo não foram uma excepção, verificando-se um grande número de trabalhos desenvolvidos em domínios muito específicos, mas relativamente poucos ao nível dos que pretendem obter modelos de avaliação integrados (que incluam de forma adequada diferentes perspectivas).

No entanto existe um conjunto vastíssimo de trabalhos que embora não versem directamente a avaliação de redes de campo, focam áreas de aplicação muito próximas ou, que pelas suas características, estão implicitamente associadas às redes de campo. Neste sentido, procurou-se alargar a discussão dos trabalhos desenvolvidos incluindo também esta perspectiva. A necessidade de obter um modelo de avaliação integrado, justifica esta abordagem à discussão. Como neste último caso o número de trabalhos desenvolvidos é muito grande, não é possível realizar uma apresentação, mesmo que sumária, de todos eles. No entanto, é possível identificar e classificar as principais linhas de investigação consideradas, e tecer alguns comentários sobre as suas características mais importantes. A classificação que foi utilizada não deve ser interpretada de forma mutuamente exclusiva, dado que muitos dos trabalhos citados podem englobar-se em várias das categorias definidas, mas assim como um forma de definir qual a principal linha de investigação associada ao trabalho desenvolvido. Assim e de acordo com esta perspectiva, estabeleceu-se a seguinte classificação:

♦ Arquitecturas distribuídas tolerantes a falhas

A importância dos sistemas de segurança crítica (*safety-critical systems*), aliada à ausência de soluções específicas, motivou desde à várias décadas o desenvolvimento de um conjunto de soluções específicas ao nível das arquitecturas distribuídas e tolerantes a falhas [LH94].

Estas arquitecturas, de forma geral, procuram definir e incorporar um conjunto muito vasto de aspectos, tais como:

- Arquitectura do *hardware*: estrutura funcional, tolerância a falhas, redundância, regiões de confinamento de falhas, etc.;

- Arquitectura de comunicações: protocolos de comunicação em grupo tolerância a falhas, tempo-real, sincronização de relógios, etc.;
- Arquitectura do *software*: aplicações distribuídas, aplicações de tempo-real, tolerância a falhas, recuperação de erros, gestão da redundância, etc.

Entre as arquitecturas mais importantes, e de aplicação genérica, salientam-se: o MAFT (*Multicomputer Architecture for Fault-Tolerance*) [KWF+88], o MARS (*Maintainable Real-Time System*) [KDK+93], o DELTA-4 [Pow94], o TTA (*Time Triggered Architecture*) [SHS+97] e o GUARDS (*Generic Upgradable Architecture for Real-Time Dependable Systems*) [PAD+99].

A importância assumida presentemente pelas aplicações de segurança crítica em veículos automóveis, fez também surgir um conjunto de arquitecturas adaptadas a este domínio de aplicação [ZPD94], nomeadamente o DACAPO (*Dependable Architecture for Control Applications with Periodic Operation*) [RLB+95], o BASEMENT [HLB+97], o X-By_Wire [XBW98] e o PALBUS [EEH+01].

◆ Escalonamento de tarefas em sistemas de tempo-real

Os modelos clássicos de escalonamento de tarefas em sistemas de tempo-real [ABD+95] foram extendidos por forma a incluírem comportamentos estocásticos. As abordagens vão deste o estabelecimento de uma garantia probabilística para o escalonamento das tarefas em cenários de falhas [BPS+99] [LB01], até ao pressuposto de considerar que os respectivos tempos de execução são probabilísticos [DGK+02] [BCP02], passando pela noção de sistemas que permitem a violação de um número máximo de *deadlines* numa determinada janela temporal (*weakly hard real-time systems*) [BBL01];

◆ Arquitecturas e protocolos de comunicação

Na sequência dos requisitos impostos pelos sistemas de segurança crítica, os trabalhos desenvolvidos vão deste a proposta de novos protocolos de comunicação com o objectivo de satisfazer os requisitos específicos destes tipos de sistemas (ex. TTP/C (*Time Triggered Protocol*) [TTP03], *FlexRay* [Fle04] e *Byteflight* [PBG99]), até à análise das condições de escalonamento de mensagens em redes de campo (ex. WorldFIP (*Factory Instrumentation Protocol*) [ATF+02], Profibus (*PROcess FIeld BUS*) [TV99a] e CAN (*Controller Area Network*) [BBN02]), passando por uma sistematização dos requisitos das arquitecturas de comunicação utilizadas nestes tipos de sistemas [ZPD94] [Rus01] [Rus03];

- ◆ **Estruturas topológicas tolerantes a falhas**

Através da definição e análise de estruturas topológicas⁵² tolerantes a falhas ao nível do *hardware* (ex. TMR (*Triple Modular Redundancy*) [CJP97]), do *software* (ex. NVP (*N-Version Programming*), RB (*Recovery Blocks*)) [AKL90] [Pom00] ou de ambos [JLK92] [LAB+95] [CJP97] [Nic98] [KB00].

- ◆ **Características dos sistemas de controlo**

Estudo dos requisitos ao nível do controlo de processos em sistemas suportados por rede de campo [Lia01]. Análise das condições de ocorrência de falhas em sistemas de controlo e respectivas consequências para o processo controlado [KS94] [KWS00]. Estudo da tolerância a falhas que é intrínseca ao próprio processo físico controlado [AGH02] [CRS02] [JNL03]. Definição de mecanismos baseados em asserções, suportados no conhecimento das características particulares do processo, para obter modos de avaria seguros (*fail safe*) [Hil00] [CMR+01] [JHC+02];

- ◆ **Modelação de sistemas de segurança crítica e de tempo-real**

Aplicação de técnicas de modelação estocástica no desenvolvimento, avaliação e validação de sistemas de segurança crítica e de tempo-real [Con92] [Con94] [TMG+94] [CT95] [KWS98] [KS00] [BNS+01] [ZFH01] [TRF03] [JNL03] [BFS+03];

- ◆ **Técnicas de injeção de falhas**

Utilização de técnicas de injeção de falhas [ACK+03] [CMS98] para avaliação das características de arquiteturas tolerantes a falhas [ACC+93] [Fol99] [MGM+99], ou de protocolos de comunicação associados [AAC+90] [ASB+03];

- ◆ **Caracterização da ocorrência de falhas**

Caracterização dos ambientes sujeitos a interferências electromagnéticas [Sho94] [KWS00]. Modelação dos efeitos de avarias coincidentes (*coincident failures*) [KBJ00] e de avarias simultâneas em múltiplos componentes do sistema (*common-mode failures*) [CC86] [FT97] [Vau98] [Mau00] [Zio02] [Xin03].

Caracterização dos mecanismos de diagnóstico e de discriminação do tipo de falhas em sistemas de controlo [PSB+98] [GCB98] [CB03].

- ◆ **Caracterização dos modos de avarias**

Definição e caracterização dos modos de avaria de um sistema [Cris91] e respectivas implicações ao nível da representatividade dos pressupostos utilizados na avaliação da confiança no funcionamento (*assumption coverage*)

⁵² O termo normalmente utilizado é *arquitecturas*, no entanto optou-se por esta terminologia para evitar confusão com o termo equivalente utilizado anteriormente.

[Pow95]. Caracterização e implementação de modos de avaria-silencioso (*fail silent*) em sistemas de controlo distribuídos [BES+96] [Tem98] [CP01] [BB03].

Deixando os trabalhos de carácter geral e focando a discussão ao nível das redes de campo, verifica-se que o número de trabalhos desenvolvidos com objectivo de obter de modelos de avaliação integrados são muito reduzidos.

Em [CRS+97] são analisadas várias arquitecturas distribuídas e genéricas compostas de nós que internamente possuem alguma redundância. São propostos vários modelos baseados em cadeias de Markov Contínuas, que incorporam os efeitos de falhas transitórias e permanentes nos nós e ao nível do seu *hardware*. A eficácia dos mecanismos de tolerância a falhas são modelados através de diferentes factores de cobertura, adaptados às características quer das falhas, quer da rede. O trabalho foca-se unicamente nos nós e não no funcionamento da rede na sua globalidade. As medidas derivadas centram-se na obtenção da fiabilidade e da segurança (*safety*) associada a cada nó.

Em [CYR+99a] [CYR+99b] o trabalho anterior é estendido e reformulado, através da utilização de rede de Petri Estocásticas. O tipo de arquitecturas avaliadas são semelhantes, adicionando-se no entanto os efeitos de falhas permanentes no meio de transmissão e a possibilidade de existirem agrupamentos (*clusters*) de nós redundantes. Ao contrário do anterior, o trabalho foca-se agora na avaliação do comportamento global da rede. Embora em termos de conceitos apresentados seja interessante, os modelos propostos são relativamente básicos e não suficientemente sistematizados.

[PS02] [Pim03] é proposto um modelo baseado em redes de Petri Estocásticas para avaliar o comportamento de várias arquitecturas baseada em redes de campo, em que os nós estão agrupados em FTUs (*Fault Tolerant Units*). As falhas consideradas são apenas do tipo permanente e afectando unicamente os nós (que desta perspectiva são considerados atómicos). As medidas obtidas estão centradas na fiabilidade dos FTUs. De forma estranha, tendo em conta os pressupostos assumidos, é utilizada simulação para obter a solução do modelo. Os modelos propostos são muito básicos e em alguns pontos mal formalizados.

Em [WCS+03] são analisados os mecanismos de tolerância a falhas em redes utilizadas em veículos automóveis. Os aspectos analisados estão relacionados com a avaliação de pressupostos assumidos durante o desenvolvimento da rede, nomeadamente os modos de avaria-silencioso (*fail silent*) dos nós e da arquitectura de comunicações. Baseados num conjunto de pressupostos simples, são desenvolvidas expressões que permitem quantificar probabilisticamente estes cenários. Em [WSL+03] é proposta uma abordagem complementar, centrada na determinação da qualidade de serviço ao nível das comunicações. Concretamente,

são quantificados, através de uma métrica da qualidade de serviço das aplicações de controlo, os efeitos dos atrasos das mensagens em diferentes cenários de falhas. Embora seja um trabalho interessante e bem formalizado ao nível dos conceitos discutidos, os modelos de avaliação propostos são muito básicos.

Ainda no contexto dos modelos integrados e próximos dos domínios de aplicação das redes de campo, salientam-se os trabalhos desenvolvidos para o MARS [KT91], o DELTA-4 [KP91] o GUARDS [AJK+00] e o TTA [ASB+03]. Embora sejam trabalhos de elevada qualidade, o seu desenvolvimento assenta muito nas características particulares dos sistemas avaliados tornando assim difícil uma transposição imediata dos modelos propostos para a avaliação das redes de campo.

Embora ao nível dos modelos integrados os trabalhos desenvolvidos para as redes de campo sejam reduzidos, o mesmo já não se pode dizer sobre trabalhos que se focam sobre domínios mais específicos das mesmas. A área onde se verificaram mais contribuições está associada ao comportamento da rede em cenários de falhas transitórias que ocorrem durante as comunicações. Nos capítulos 5 e 6 far-se-á uma discussão pormenorizada sobre estes tipos de trabalhos para as redes Profibus e CAN respectivamente. No que diz respeito às restantes redes, salientam-se os seguintes trabalhos.

Em [CMM99] são discutidos e analisados os efeitos de falhas transitórias na rede decorrente do *standard* IEC/ISA, nomeadamente ao nível dos seus mecanismos de recuperação. O comportamento da rede é analisado utilizando um ambiente de simulação em C++ desenvolvido especificamente para este caso. Em [CM00] os mesmos autores adaptam o trabalho anterior ao IEC 61158 tipo 1. São propostos modelos baseados em rede de Petri Estocásticas e os resultados são obtidos por simulação. Os modelos propostos nestes trabalhos têm como objectivo apenas analisar condições muito particulares de operação.

Num contexto não directamente relacionado com a ocorrência de falhas durante a comunicação, em [JA92] [JAB94] é modelado o comportamento da rede WorldFIP em função de diferentes tipos de tráfego, utilizando para isso uma abordagem baseada em rede de Petri Estocásticas. Embora os modelos obtidos sejam aparentemente analíticos, a formalização da solução através da utilização de uma ferramenta de modelação específica (RTPS [Ata93]), sugere algumas dúvidas ao nível conceptual.

Da discussão realizada nesta secção é possível concluir que embora existam muitos trabalhos sobre domínios específicos, são reduzidos e relativamente pouco abrangentes os que procuram obter uma perspectiva integrada do problema. Justifica-se assim a necessidade de propor uma nova abordagem que tenha como objectivo o desenvolvimento de um modelo de avaliação integrado.

4.2.3 Modelação Utilizando Redes de Petri Estocásticas

Ficou patente na discussão dos modelos apresentados no capítulo 3, que a modelação utilizando cadeias de Markov só pode ser realizada de forma eficaz quando o sistema a modelar não é complexo e quando se verificam pressupostos Markovianos ao nível dos processos estocásticos subjacentes.

A hipótese Markoviana limita muito o tipo de comportamentos que se podem modelar e analisar. Só em casos muito específicos é que o sistema em causa possui um comportamento deste tipo, resultando desta forma que os modelos obtidos são de facto uma aproximação do comportamento real. Esta aproximação é frequentemente assumida de forma consciente, com o objectivo de tirar partido das vantagens disponibilizadas pela utilização de um modelo analítico. É assim encarada como um compromisso entre o rigor do modelo e facilidade de obter a respectiva solução.

Quando os sistemas são complexos, a utilização de cadeias de Markov dão origem a problemas relacionados com a construção efectiva do próprio modelo. Estes problemas não resultam de quaisquer questões computacionais associadas à respectiva solução (que também existem e serão discutidas mais adiante), mas sim do processo, digamos manual, da própria construção do modelo. Estas dificuldades surgem a dois níveis. Quanto mais complexo for o sistema a modelar (e com maiores interacções entre os seus componentes) maior é a dimensão do seu espaço de estados. A existência de um número elevado de estados, dificulta o respectivo processo de enumeração e da definição das características das transições entre os mesmos. Por fim, como ferramenta gráfica (ou até mesmo considerando uma interface textual), o processo de construção do modelo torna-se penoso e sujeito a erros frequentes. Torna-se assim necessário adoptar técnicas de modelação que permitam adoptar níveis de abstracção maiores, mas sem contudo perderem o poder e a capacidade de modelação.

Da discussão realizada no capítulo 2, ficou patente que as redes de Petri Estocásticas⁵³ são um formalismo de modelação que possui todas as características necessárias para resolver este problema [Bal95]. As suas principais vantagens residem no elevado poder de modelação, na capacidade de representarem de forma compacta sistemas complexos, na flexibilidade de utilização e na possibilidade de se obterem soluções analíticas ou de simulação de forma directa. Acresce ainda o facto da sua crescente utilização com sucesso para o desenvolvimento de modelos focados na avaliação da confiança no funcionamento [AKL90] [Con92] [MCT94] [CT95] [MT95] [Lin98] [MBZ+99] [SGB00] [BNS+01] [AH02] [Triv02] [BFS+03] [TRF03].

⁵³ Por questões de simplicidade de linguagem utilizar-se-á deste ponto em diante o termo *rede de Petri* como sinónimo de rede de Petri Estocástica.

4.2.3.1 Aspectos Prévios ao Desenvolvimento de Modelos Baseados em Redes de Petri

O conjunto de vantagens disponibilizadas pelas redes de Petri, leva muitas das vezes o utilizador inexperiente a uma abordagem simplista do processo de modelação. É frequente considerar-se que após tomadas em conta as restrições impostas pelo processo estocástico subjacente, as redes de Petri podem ser utilizadas para analisar e modelar qualquer sistema, sem quaisquer limitações que não decorram das restrições impostas pelo seu poder de modelação ou pelo tipo de solução pretendida.

É assim comum nestas situações, o utilizador adoptar uma estratégia em que concentra todo o seu esforço no desenvolvimento do modelo, nomeadamente no grau de detalhe pretendido. A solução do mesmo é considerada um problema que envolve apenas cálculo e que pode ser obtida de forma transparente e exacta pela ferramenta de modelação utilizada.

Contudo em muitos dos casos esta estratégia falha, porque os recursos computacionais exigidos para a resolução do modelo são de tal forma elevados, que a obtenção da respectiva solução se torna impraticável. Note-se que este problema não surge das limitações inerentes à própria capacidade de modelação do formalismo (ex. tipo de processos estocásticos suportados), mas sim dos aspectos directamente relacionados com a forma de obter a solução. Dito de outra forma, o modelo pode teoricamente ser resolúvel, mas em termos práticos verifica-se que a obtenção da solução não é exequível.

Torna-se assim óbvio que existe uma relação entre as características do modelo desenvolvido e a exequibilidade da sua solução. É assim necessário analisar que tipos de restrições existem e de que forma influenciam as características do modelo.

Nas secções seguintes far-se-á uma descrição dos principais problemas que devem ser equacionados durante o desenvolvimento de um modelo baseado em redes de Petri. Assume-se previamente a esta discussão que o objectivo passa obter uma solução analítica, e que o modelo em causa foi desenvolvido de acordo com as restrições estruturais, discutidas no capítulo 2, que asseguram a existência dessa mesma solução.

4.2.3.1.1 Dimensão do Espaço de Estados

Como qualquer modelo do tipo espaço de estados (*state space models*) [MFT00], as redes de Petri obrigam, para obter a respectiva solução, à enumeração de todos os estados do sistema modelado.

Nos sistemas simples, o espaço de estados é normalmente reduzido não dando origem quaisquer tipo de problemas. Contudo, à medida que a complexidade do sistema aumenta, isto geralmente implica também o aumento do respectivo espaço de estados. Como a resolução do modelo utilizando meios computacionais, implica a necessidade de dispor de mecanismos que permitam armazenar esse mesmo espaço de estados (tipicamente sob a forma de uma matriz), resulta daqui um problema designado como *explosão do espaço de estados* [MFT00]. Ou seja, o espaço de estado atinge uma dimensão tal que não é exequível obter uma solução sem se dispor de recursos computacionais muito grandes ou mesmo ilimitados. Note-se, que durante o desenvolvimento do modelo este problema nem sempre é óbvio devida à elevada capacidade de síntese das redes de Petri. Só durante a fase da geração do espaço de estados (pela ferramenta de modelação) é que este problema muitas das vezes se torna claro.

Como este é o principal problema que se depara durante o desenvolvimento de modelos baseados em rede de Petri, a comunidade científica dedicou um considerável esforço ao desenvolvimento de um conjunto de técnicas que permitam minimizar este problema (*largeness avoidance*) [ML96], das quais se destacam-se as seguintes :

- ◆ **Truncatura do espaço de estados**

A partir de uma análise prévia do comportamento do sistema, são identificados os estados aos quais correspondem as mais baixas probabilidades de ocorrência ou de ocupação, sendo estes posteriormente eliminados, geralmente de forma manual [MSG89]. Trata-se assim de um método baseado numa aproximação. Esta técnica foi inicialmente adoptada para as cadeias de Markov, onde em princípio é mais simples identificar os estados em causa a partir de uma inspecção da estrutura da própria cadeia [LS90]. Nas redes de Petri como o acesso aos estados é realizado de forma indirecta a partir da marcação da rede, torna-se mais difícil realizar a respectiva identificação e eliminação. Esta tarefa passa tipicamente por uma análise da estrutura da rede e da evolução da marcação através do grafo das marcações acessíveis [KT91];

- ◆ **Agregação de estados**

Através da utilização de argumentos probabilísticos (*lumping theorems*) [Kao97] é possível agregar vários estados num único. Esta é uma técnica particularmente utilizada nas cadeias de Markov onde é possível aceder directamente aos estados do sistema [SL87] [LYT03]. Quando se pretende realizar uma análise em regime estacionário esta técnica permite obter resultados exactos, embora com alguma perda de informação específica sobre os estados agregados [Buc94]. No caso de uma análise transitória a solução obtida geralmente resulta numa aproximação [BT86]. Ao nível das redes de

Petri, a utilização directa desta técnica torna-se mais difícil devido ao acesso indirecto ao espaço de estados. As aplicações reduzem-se tipicamente a modelos que exibam uma estrutura particular [Til95] [Kor99]. No entanto, quando existem simetrias entre modelos, esta técnica revela-se extremamente eficaz [SM91a];

- ◆ **Composição de modelos**

A estrutura funcional do sistema permite que o modelo global seja obtido através da composição de vários modelos, cujas soluções são obtidas de forma independente e posteriormente combinadas através da troca de resultados entre modelos [DTS+86] [MT93]. A utilização desta técnica assume implicitamente que os vários modelos são independentes entre si;

- ◆ **Decomposição estrutural**

A partir da estrutura funcional do modelo, é realizada uma decomposição em vários submodelos cujas soluções são obtidas de forma independente e posteriormente processadas de forma hierárquica [MT93] [LYT03]. A utilização desta técnica assume que os modelos são independentes ou que o grau de dependência entre estes é muito reduzido [CT93];

- ◆ **Decomposição temporal**

A existência de diferentes escalas temporais no modelo, permite realizar uma decomposição hierárquica do modelo numa sequência de submodelos, em que cada um é válido durante uma determinada escala temporal [AI89]. Esta decomposição pode ser realizada com base nas características funcionais do sistema a avaliar [TG83] [Con92] [Con94] ou na estrutura do próprio modelo obtido [AHL87] [Kor99]. Os resultados obtidos por cada modelo são posteriormente integrados em modelos hierarquicamente superiores;

- ◆ **Resolução iterativa**

O modelo é decomposto em vários submodelos, mas em que se assume a existência de uma certa interacção entre estes. Assim, alguns dos parâmetros de um modelo são definidos como dependentes da solução de outros submodelos. Desta forma e utilizando uma estimativa inicial para os parâmetros, os modelos são iterativamente resolvidos até ser atingida a convergência da solução [CT92] [Hav99]. A aplicação deste método está dependente da existência de condições específicas que garantam a respectiva convergência [MT96];

Por oposição às técnicas anteriores, foram também desenvolvidas técnicas que evitam o armazenamento explícito do espaço de estados (*largeness tolerance*), sem contudo o limitar. Um exemplo deste tipo de técnicas é utilização de operadores de Kronecker, em que o espaço de estados é descrito através de uma soma de

operadores deste tipo, o que possibilita a sua decomposição em matrizes de menor dimensão [CT96] [Buc04].

4.2.3.1.2 Ordem de Grandeza dos Parâmetros

A existência no modelo de parâmetros cujas ordens de grandeza diferem significativamente (*stiffness*), resulta geralmente na existência de problemas de eficiência ao nível dos métodos numéricos utilizados para resolver o modelo [MFT00]. Estes problemas traduzem-se tipicamente na instabilidade durante o processo de convergência dos métodos numéricos. Isto reflecte-se no aumento do tempo necessário para se obter uma solução ou até na impossibilidade de a obter devido à ausência de convergência.

Este problema, em princípio, só se coloca quando é necessário resolver sistemas de equações diferenciais (tipicamente no cálculo de matizes exponenciais [Ger00]), estando portanto presente na análise transitória de sistemas Markovianos e nas análises transitórias e estacionárias de sistemas Não-Markovianos.

A resolução deste problema passa por utilizar uma das seguintes estratégias:

- ♦ Impedir que o modelo contenha parâmetros deste tipo (*stiffness avoidance*). Uma das soluções possíveis passa por utilizar algumas das técnicas de agregação de estados [BT86] ou de decomposição temporal [DT89] descritas previamente;
- ♦ Utilizar técnicas de solução dos modelos que sejam numericamente estáveis em cenários deste tipo (*stiffness tolerance*) [MMT94]. Este aspecto está fortemente associado às características da ferramenta utilizada para obter a solução do modelo.

4.2.4 Metodologia Utilizada no Desenvolvimento dos Modelos

A metodologia utilizada para o desenvolvimento dos modelos propostos neste capítulo, passou pela definição prévia de um conjunto de requisitos que os modelos devem respeitar:

- ♦ **Redução do espaço de estados**

Este problema foi abordado segundo duas perspectivas. Em primeiro lugar utilizar técnicas para a construção dos modelos que minimizem a marcação da rede de Petri [Ben94]. A utilização destas técnicas passa numa primeira fase pela identificação das partes do estado do sistema dos quais não é necessário manter registo e posteriormente por uma construção do modelo por forma que a marcação da rede de Petri reflecta esta situação.

A segunda perspectiva, passou por analisar ao nível das ferramentas de modelação analítica de redes de Petri a possibilidade destas incluírem técnicas que permitam a redução do espaço de estados do modelo. Após a análise das características das várias ferramentas existentes [JM88] [TM93] [HN96], verificou-se que a ferramenta **Möbius** [CCD+01] (anteriormente denominada UltraSAN) implementa uma técnica deste tipo. Dadas as vantagens proporcionadas por esta ferramenta (descritas nos parágrafos seguintes), optou-se por utilizá-la como ambiente para o desenvolvimento dos modelos propostos.

A redução do espaço de estados nesta ferramenta baseia-se na reformulação na forma como este é definido em modelos que exibam simetrias estruturais [SM91a]. Mais concretamente, se o modelo que se pretende desenvolver for possível de decompor em várias réplicas estruturalmente iguais, o espaço de estados do sistema pode ser obtido de forma muito mais compacta. Grosso modo, a ideia por detrás desta abordagem está na definição do novo espaço de estados como o número de modelos que está numa determinada marcação, e não como o produto cartesiano das marcações dos modelos individuais.

Embora numa primeira análise a necessidade do modelo ter que ser decomposto em réplicas possa parecer uma limitação, no contexto do problema definido neste capítulo acaba por ser uma enorme vantagem. De facto, como a rede de campo é constituída por vários nós com características funcionais semelhantes, o conceito de replicação aplica-se perfeitamente. Assim, em vez de se desenvolver um modelo específico para cada nó, basta ter um único modelo que representa o funcionamento de um nó em geral, e utilizar o conceito de replicação da ferramenta para construir o modelo dos restantes nós, e assim obter o modelo completo da rede de campo. É possível assim desenvolver modelos mais detalhados, sem o problema (dentro de certos limites) da dimensão do espaço de estados.

O Möbius implementa um formalismo equivalente às redes de Petri Estocásticas (Regenerativas) denominado SAN (*Stochastic Activity Networks*) [SM01]. As SANs são compostas por posições, actividades e portas (*gates*) de entrada e saída. As posições e as actividades são equivalentes respectivamente aos conceitos de posição e de transição das redes de Petri clássicas, tendo também uma representação gráfica equivalente (Fig. 4.2).

As actividades são de dois tipos: temporizadas e imediatas, podendo para o primeiro caso especificarem-se alguns tipos de políticas de memória (PRD e PC). Associadas a cada actividade podem existir *selecções* (*cases*). Estas são utilizadas para modelar o comportamento probabilístico após o disparo da actividade (política de pré-selecção, ver capítulo 2). Assim, quando a

actividade dispara é escolhida apenas uma das selecções possíveis para modificar a marcação da rede. Esta escolha é realizada através de uma distribuição discreta de probabilidade definida pelo utilizador. Um dos aspectos relacionados com esta distribuição é o de permitir de forma mais simples o desenvolvimento modelos bem especificados (*well specified*, ver capítulo 2) ao nível do processo estocástico subjacente [DS99]. Graficamente, as selecções são representadas por pequenos círculos adjacentes às actividades.

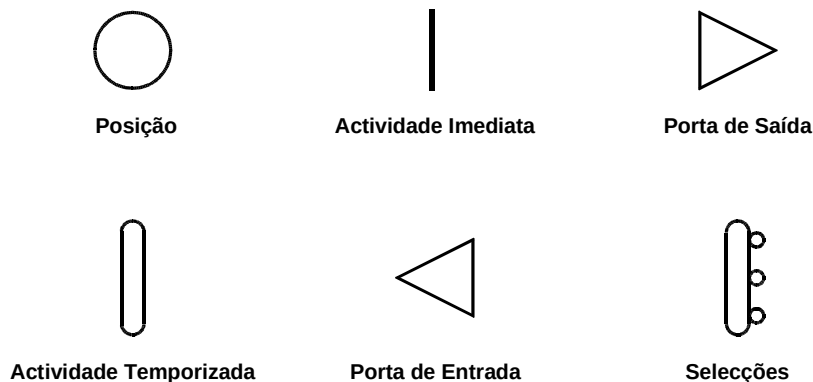


Figura 4.2 - Representação gráfica das SAN.

As regras de evolução são iguais às das rede de Petri. As actividades estão habilitadas quando as posições anteriores possuem testemunhos ou quando as portas de entrada as habilitam. As portas são utilizadas para interligar as posições às actividades. As portas de entrada permitem definir de forma muito flexível quais as condições em que as actividades estão habilitadas. Para isso dispõem de um predicado e de uma função. O predicado é utilizado para definir a habilitação das actividades (guardas). A função é executada quando a actividade dispara, permitindo assim modificar a marcação da rede. As portas de saída especificam a forma como a marcação da rede é alterada após o disparo das actividades, dispondo para isso também de uma função. Em ambos os casos as funções e os predicados são definidos através de código em C++. Ambas as portas são representadas graficamente por triângulos orientados.

A ferramenta permite também a utilização de técnicas de modelação hierárquica. Para isso dispõe de duas primitivas: *Join* e *Rep*, que permitem a construção de um modelo composto. A primitiva *Join* permite compor modelos através de posições comuns que são partilhadas entre estes. A primitiva *Rep* permite replicar os modelos e também definir eventuais posições comuns entre eles.

As medidas são obtidas através da utilização de modelos de recompensa do tipo taxa (*rate reward*) ou impulso (*impulse reward*). As taxas podem ser definidas como uma função que depende da marcação da rede, enquanto que

os impulsos são definidos como uma função associada ao disparo das actividades. O Möbius dispõe de uma interface gráfica muito flexível, que lhe permite realizar múltiplos tipos de estudos. Além disso encontra-se desenvolvido em Java, sendo assim suportado pela maioria dos sistemas operativos;

♦ Modelação hierárquica

A utilização de técnicas de modelação hierárquica permitem decompor um modelo complexo em vários modelos mais simples e de menor dimensão. É garantido assim ao utilizador uma maior flexibilidade no desenvolvimento do modelo, e também uma maior eficácia no processo da sua validação [MT93] [CYR+99a] [BNS+01].

O problema em análise neste capítulo tem uma estrutura nitidamente hierárquica. Isto pode se observado quer ao nível do *hardware* (ex. decomposição do nó nos seus componentes constituintes), quer ao nível do *software* (ex. diferentes níveis funcionais). É assim possível decompor o modelo pretendido de acordo com uma estrutura hierárquica de submodelos de menor dimensão. Isto permite refinar o modelo até a um nível que se julgue adequado, ou que as limitações computacionais o restrinjam.

Se os submodelos forem de alguma forma independentes, é possível obter a sua solução em separado e posteriormente incorpora-la no modelo de alto nível. Note-se também que esta abordagem também contribui para a redução do espaço de estados. No caso dos modelos não serem independentes, é necessário dispor de uma ferramenta (como é o caso) que permita realizar a composição hierárquica dos mesmos;

♦ Modularidade

A opção pela utilização de uma abordagem modular, partiu do facto de que o problema proposto ser totalmente genérico (não se foca numa rede em particular) e incluir muitos aspectos distintos. O desenvolvimento de um modelo monolítico, para além de ser geralmente mais complexo, implica também uma menor flexibilidade, dado que a introdução ou a modificação de alguns pressupostos leva, muitas das vezes, a uma reconstrução de raiz da totalidade do modelo.

A opção por um desenvolvimento modular permite uma maior flexibilidade, uma reutilização de modelos já desenvolvidos e uma melhor validação dos mesmos [FKK99] [AK02]. No entanto, existe um custo ao utilizar este tipo de abordagem. As interfaces dos modelos têm que estar muito bem definidas para que a sua composição seja possível [KB00]. Isto obriga a um maior esforço por parte de quem desenvolve os modelos, dado que é necessário

analisar as possíveis interações entre os vários modelos e representa-las de forma adequada;

♦ Decomposição temporal

A existência no problema em causa de subsistemas cujas escalas temporais diferem em várias ordens de grandeza e com um elevado grau de independência, permite realizar uma decomposição temporal do mesmo. Contribui-se assim também para a redução do espaço de estados.

A técnica adoptada neste trabalho foi a *decomposição comportamental* (*behavioral decomposition*) [TG83] [DT89] [Gei91] [Con92]. Segundo esta técnica, o comportamento funcional de um sistema com estas características, permite decompor o respectivo modelo em vários submodelos de acordo com a sua escala temporal, geralmente denominados *rápidos* e *lentos*. Cada um dos modelos rápidos é resolvido de forma independente (assume-se assim implicitamente a existência de um grau de independência elevado), e posteriormente os seus resultados são incorporados nos modelos lentos de acordo com a sua estrutura funcional.

Por detrás desta técnica, está o pressuposto de que os modelos rápidos chegam ao seu regime estacionário (estocástico) num intervalo de tempo que é várias ordens de grandeza inferior ao dos modelos lentos, e assim, na perspectiva destes últimos, os modelos rápidos podem ser observados (e modelados) como se já estivessem em regime estacionário. Note-se, que esta técnica só pode ser considerada válida se os modelos forem independentes ou com interações muito ligeiras;

♦ Decomposição estrutural

Embora as redes de campo possuam características muito diversas entre si, é possível, dentro desta diversidade, identificar ao nível estrutural (e de certa forma também funcional) um conjunto de subsistemas que são comuns a todas elas. Assim, e com base nas características típicas dos sistemas de controlo baseados em rede de campo [CYR+99a] [PB99], é possível realizar uma decomposição estrutural da seguinte forma:

- **Nó.** Cada nó pode ainda ser decomposto nos seus componentes de *hardware* e de *software*;
- **Cluster.** Um *cluster* representa um conjunto de nós que estão organizados de acordo com uma arquitectura tolerante a falhas, e que do ponto de vista lógico operam como se se tratasse de um único nó;
- **Infra-estrutura de comunicações.** Esta infra-estrutura pode também ser decomposta em componentes de *hardware* e de *software*.

Esta decomposição incorpora implicitamente dois aspectos. Os subsistemas definidos, e a respectiva decomposição interna, vão corresponder aos componentes da rede onde os efeitos das falhas vão implicar as maiores consequências ao nível do seu funcionamento. Desta forma, a decomposição reflecte a importância destes componentes ao nível da sua contribuição para o comportamento global da rede. O segundo aspecto, está relacionado com o facto de se pretender que o modelo seja o mais genérico possível. O objectivo é assim dispor de modelos facilmente adaptáveis que possam ser utilizados em diferentes situações.

No entanto, este carácter genérico não implica qualquer ausência de rigor ou de minimização de detalhes nos modelos desenvolvidos, antes pelo contrário. Neste sentido, o desenvolvimento dos modelos foi efectuado segundo duas perspectivas. Por um lado, utilizar um nível de abstracção que permita incorporar o máximo número de funcionalidades possíveis sem comprometer o carácter genérico do mesmo. Por outro, deixar em aberto a possibilidade de introduzir novos elementos nos modelos, com o objectivo de descrever com maior detalhe o comportamento de uma rede de campo em particular;

♦ Características da falhas consideradas

Sem perda de generalidade, considerou-se que todas as falhas têm origem apenas nos componentes de *hardware* do sistema, e que ocorrem de acordo com um processo de Poisson. Este pressuposto é baseado na abordagem clássica de ocorrência das falhas em componentes de eléctricos ou electrónicos [MIL91].

O pressuposto anterior assume de forma implícita que o *software* não contribui de forma directa para o processo de ocorrência das falhas. Ou seja, o processo de desenvolvimento do *software* não introduziu qualquer tipo de falhas (dormentes) no mesmo.

No entanto é um facto conhecido que não é possível garantir que num sistema com este grau de complexidade o *software* não contenha falhas dormentes, o que parece, numa primeira abordagem, que assumir este pressuposto resulta numa limitação importante quando se pretende desenvolver um modelo integrado de avaliação. Neste sentido é preciso justificar os motivos que estiveram por detrás do sua consideração.

Conforme se verificará durante a apresentação dos modelos, o seu carácter genérico e modular permite, com extrema simplicidade, a sua utilização ou adaptação para representar o comportamento de falhas com origem quer no *hardware* quer no *software*. A particularização que foi realizada para o primeiro caso (falhas com origem no *hardware*), deveu-se apenas ao facto de se conhecer com maior detalhe o tipo de processos estocásticos que estão na origem das falhas, enquanto que no segundo caso a sua caracterização é mais complexa

[Goe85] [Yam00]. Desta forma, esta particularização deve ser encarada apenas como uma forma de simplificar a apresentação dos vários conceitos que estão subjacentes ao modelo de avaliação proposto, e não como uma limitação do mesmo.

De uma perspectiva associada à persistência temporal, as falhas podem ser de dois tipos: permanentes ou transitórias. Ao contrário dos modelos discutidos no capítulo anterior, neste caso assume-se que as falhas transitórias podem afectar qualquer componente da rede. Este pressuposto resulta do facto de vários estudos indicarem que este tipo de falhas representarem a maioria das falhas que afectam os sistemas [GCB98] [PSB+98].

A ocorrência de uma falha num componente de *hardware*, caso não seja tolerada de forma adequada, possibilita a sua propagação aos componentes de *software* que com ele interagem. Assume-se a existência de mecanismos de tolerância a falhas quer ao nível do *hardware*, quer ao nível do *software*. Além disso, considerou-se também a possibilidade de existirem avarias em modo-comum;

♦ Modos de avaria considerados

De acordo com as abordagens recomendadas para a avaliação de sistemas de segurança crítica [Gob98] [IEC04a], os modos de avaria da rede foram classificados de acordo com a sua gravidade em *seguros* e *não-seguros*. O primeiro, corresponde a uma avaria cujas consequências não coloquem em causa a integridade de pessoas e bens. O segundo modo de avaria contempla os casos diametralmente opostos.

Por forma a incorporar o comportamento referente a estes modos de avarias, utilizou-se uma abordagem semelhante à proposta na FMECA (*Failure Modes, Effects and Criticality Analysis*) [Wei91] e também recomendada ao nível dos standards associados ao desenvolvimento de sistemas de segurança crítica [IEC04a]. Neste sentido, as falhas podem classificar-se de acordo com a sua *gravidade potencial* em duas categorias: *seguras* e *não-seguras*. No primeiro caso, se a sua ocorrência se não for convenientemente tolerada (não-coberta) pelos mecanismos de tolerância a falhas existentes no sistema, ocorrerá uma avaria segura da rede. No segundo caso e nas mesmas condições, ocorrerá uma avaria não-segura. No entanto e neste último caso, se a falha for tolerada (coberta) no pior cenário apenas poderá dar origem a uma avaria segura da rede. Esta interpretação assume assim, que desde que os mecanismos de tolerância a falhas funcionem correctamente, as avarias da rede serão sempre seguras. Refira-se de novo, que esta classificação é baseada num conceito que considera quais as consequências *potenciais* da falha na gravidade da avaria do sistema se a falha não for convenientemente tolerada.

Após a ocorrência de uma falha permanente, assume-se que o componente só pode repor o seu serviço após uma reparação. Podem existir reparações de componentes específicos ou da rede na sua globalidade. Também sem perda de generalidade, assume-se que as reparações ocorrem de acordo com uma distribuição exponencial, ou seja, com uma taxa constante. Como posteriormente se verificará, esta restrição pode ser facilmente modificada.

♦ Tipo de solução e medidas pretendidas

Pretende-se que a solução do modelo seja obtida através de técnicas analíticas e/ou numéricas, dado o conjunto de vantagens, nomeadamente computacionais, disponibilizadas por este tipo de solução.

As medidas que se pretendem obter do modelo estão associadas à fiabilidade, disponibilidade e segurança (*safety*). Estas podem ser obtidas quer em regime estacionário, quer em regime transitório.

4.3 Modelos Desenvolvidos

De acordo os requisitos de modularidade definidos, foram desenvolvidos um conjunto de modelos que reflectem a decomposição estrutural do sistema que se pretende avaliar. A partir da composição do conjunto destes modelos, obtêm-se o modelo de avaliação integrado da rede de campo.

Nas secções seguintes é realizada uma descrição pormenorizada de cada um dos modelos, bem como a uma discussão dos pressupostos que serviram de base para o seu desenvolvimento.

4.3.1 Modelo HW

O modelo **HW** foi desenvolvido com o objectivo de representar o comportamento de um componente de *hardware* do nó⁵⁴ (Fig. 4.3). De acordo com a perspectiva de modelação hierárquica discutida previamente, este modelo pode representar desde todo o *hardware* de um nó, até a um subsistema específico do mesmo (ex. CPU, memória, sensores, actuadores, etc.).

Para representar a eventual existência de redundância estrutural ao nível do componente (ex. 2 CPUs redundantes), optou-se por incluí-la directamente no modelo do componente, ao invés de utilizar vários modelos (1 por cada

⁵⁴ Como será descrito adiante, este modelo vai também ser utilizado (com adaptações) para representar outros componentes de *hardware* da rede.

componente). Neste sentido, o número de componentes operacionais do tipo representado pelo modelo é indicado pelo número de testemunhos na posição *hw*.

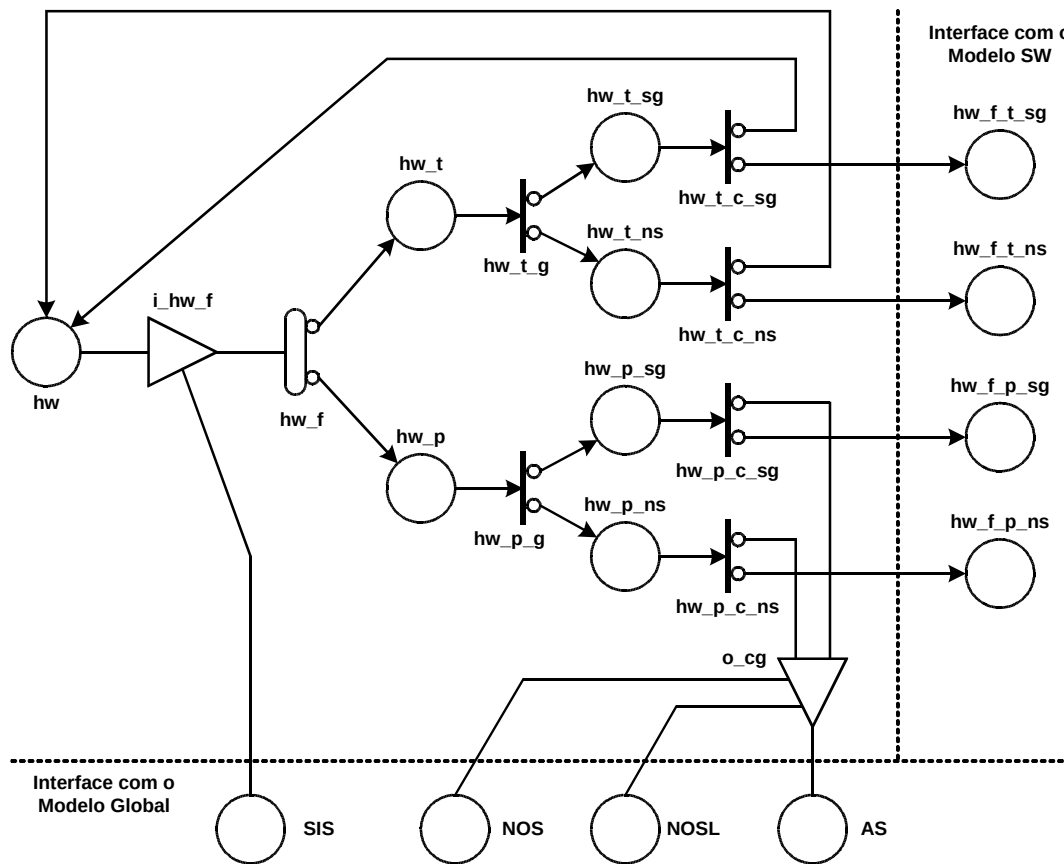


Figura 4.3 - Modelo HW.

A falha⁵⁵ do componente só pode ocorrer se estiverem reunidas um conjunto de condições para tal. As condições aqui referidas apenas têm a ver com o modelo em si, e não directamente com as condições reais (desde que o componente esteja operacional pode avariar em qualquer momento independentemente do estado do sistema). Ao nível do modelo, estas condições são representadas através da porta de entrada *i_hw_f*. Estas dependem do estado global da rede, representado pela posição *SIS*, e das condições locais, que por sua vez são uma função das características funcionais consideradas para o sistema. A posição *SIS* é global⁵⁶ e indica, caso contenha um testemunho, que a rede está num estado de avaria (segura ou não). Note-se que a definição destas condições resulta apenas da especificação existente para o sistema, e que desta forma são deixadas em aberto ao nível do modelo.

55 De facto trata-se da sua avaria. No entanto, e no sentido da discussão realizada no capítulo 3, utilizar-se-à o termo falha.

56 As posições globais representam posições que são comuns a vários modelos. Em termos de simbologia utilizada são representadas pela sua denominação em maiúsculas, por oposição às posições locais que são representadas por minúsculas.

Com o objectivo de concretizar estas condições, e sem perda de generalidade, a Tabela 4.1 representa um exemplo típico do código associado a esta porta⁵⁷. Neste caso o exemplo é muito simples, pois considera que enquanto o sistema estiver operacional (*SIS* não marcada) e existir redundância entre componentes (*hw* marcada), os componentes operacionais podem avariar (remoção de um testemunho de *hw*).

	i_hw_f
Predicado	/* Componentes operacionais e sistema operacional */ (hw->Mark() > 0) && (SIS->Mark() == 0)
Função	/* Componente avariado */ hw->Mark()--;

Tabela 4.1 - Exemplo do código da porta i_hw_f.

Se as condições definidas anteriormente estiverem reunidas, o componente pode avariar. Esta ocorrência é representada pela transição exponencial *hw_f*. Para representar de forma correcta a taxa de avarias real, a taxa associada a esta transição (sob a forma do inverso da sua duração) deverá ser função do número de componentes operacionais existentes, da seguinte forma

$$Taxa_{hw\_f} = hw \rightarrow Mark() \cdot \lambda_{HW} \quad (4.1)$$

em que λ_{HW} representa a taxa de avaria do componente em causa. Este aspecto decorre directamente das propriedades do processo de Poisson.

Conforme já discutido, de uma perspectiva de persistência temporal a falha pode ser de dois tipos: transitória ou permanente. A distinção entre ambas é realizada através da escolha das selecções associadas à transição *hw_f*. A primeira escolha (sentido de cima para baixo), indica a ocorrência de uma falha transitória. Esta é seleccionada com probabilidade p_{hw_t} e a posição *hw_t* é marcada. A segunda escolha, indica a ocorrência de uma falha permanente, sendo seleccionada com probabilidade $(1 - p_{hw\_t})$, implicando a marcação da posição *hw_p*. Ao nível da modelação, esta abordagem é comum à maioria dos trabalhos existentes [DT89] [KKG+90] [CT95] [CYR+99a] [KB00] [Con01].

Esta separação probabilística, resulta directamente das propriedades do processo de Poisson, podendo ser formalizada da seguinte forma

$$\begin{aligned} \lambda_{HW}^P &= \lambda_{HW} \cdot (1 - p_{hw\_t}) \\ \lambda_{HW}^T &= \lambda_{HW} \cdot p_{hw\_t} \end{aligned} \quad (4.2)$$

⁵⁷ O método (C++) **XX->Mark()** indica o número de testemunhos na posição **XX**.

em que λ_{HW}^P e λ_{HW}^T são respectivamente as taxas de avaria permanentes e transitórias do componente.

As consequências potenciais da ocorrência de uma falha são de dois tipos: seguras e não-seguras. Este comportamento foi modelado através de transições imediatas com duas selecções, associadas respectivamente a cada uma das consequências. A selecção é definida através de uma probabilidade. Ao nível da modelação, esta abordagem é também a usualmente utilizada nos sistemas de segurança crítica [Gob98] [ISA02] [IEC04a].

Assim e se a falha for transitória, a posição hw_t está marcada permitindo que a transição imediata hw_t_g fique habilitada. Esta possui duas selecções associadas à gravidade potencial da falha. A escolha da primeira selecção corresponde à ocorrência de uma falha segura, a qual é seleccionada probabilidade $p\_hw\_t\_sg$, resultando na marcação da posição hw_t_sg . A escolha da segunda selecção corresponde à ocorrência de uma falha não-segura, a qual é seleccionada com probabilidade $(1-p\_hw\_t\_sg)$, resultando na marcação da posição hw_t_ns .

No caso das falhas permanentes o comportamento é semelhante. Assim partindo da marcação da posição hw_p , as posições hw_p_sg e hw_p_ns podem ser marcadas com uma probabilidade de escolha $p\_hw\_p\_sg$. Estas posições correspondem respectivamente a falhas seguras e não-seguras.

Tal como discutido previamente a rede possui vários mecanismos de tolerância a falhas, incluindo naturalmente ao nível do seu *hardware*. Estes mecanismos podem ser de muitos tipos e com características muito diversas [KS00]. Ao nível da modelação, podem existir duas abordagens para a respectiva representação. A mais imediata, é introduzir directamente no modelo o funcionamento destes mecanismos. Contudo e devido à complexidade natural dos mesmos, isto levará seguramente a um aumento do espaço de estados, pelo que deve ser evitada.

A segunda abordagem, passa por utilizar técnica da decomposição comportamental (*behaviorial decomposition*) descrita anteriormente. Em termos funcionais, os mecanismos de tolerância a falhas actuam no período imediatamente a seguir à ocorrência da mesma. Além disso, é um facto conhecido que sua execução decorre numa escala temporal que é muitas ordens de grandeza inferior às taxas de avaria e reparação do componentes [KS00]. Se se admitir que durante este período de tempo as interacções ao nível estocástico (falhas, reparações, etc.) com outras partes do modelo são mínimas ou mesmo nulas, é possível decompor o modelo em duas partes: uma referente aos mecanismos de tolerância a falhas (modelo *rápido*), e outra referente ao comportamento funcional do sistema durante a ocorrência de falhas e reparações (modelo *lento*). Se exceptuarmos o caso de falhas quase coincidentes (*near coincident faults*) [DT89] [Gei91] [PCA+96], cujo interesse só é

relevante para sistemas ultra-fiáveis (*ultra-reliable systems*), o pressuposto assumido é perfeitamente razoável.

Desta forma, o modelo referente aos mecanismos de tolerância a falhas (detecção, diagnóstico e recuperação de erros) é modelado e resolvido em separado. Como a técnica descrita considera que o modelo rápido já está em regime estacionário, a resolução do modelo rápido implica a obtenção das probabilidades estacionárias associadas ao funcionamento com sucesso, ao não, dos mecanismos. Estes valores são posteriormente incorporados no modelo funcional do sistema, através de um factor de cobertura (*coverage factor*, ver capítulo 3) (Fig. 4.4) [DT89] [Gei91] [Con92]. A utilização desta técnica garante assim uma enorme redução no espaço de estados do modelo.

O factor de cobertura [BCS69] é um parâmetro extremamente crítico na avaliação da confiança no funcionamento de um sistema (conforme se discutiu e observou no capítulo 3), tendo sido alvo de muitos estudos [DT89] [ACC+93] [PMA+03] [PMA+95] [PCA+96] [Fol99] [Hil00] [Con03]. Note-se contudo, que a obtenção deste parâmetro não tem que ser necessariamente realizada por intermédio de um modelo de avaliação, podendo ser obtida através da utilização de outras técnicas. Devido à necessidade de se obter valores rigorosos para o mesmo, a situação mais comum é recorrer a técnicas de injeção de falhas [Fol99].

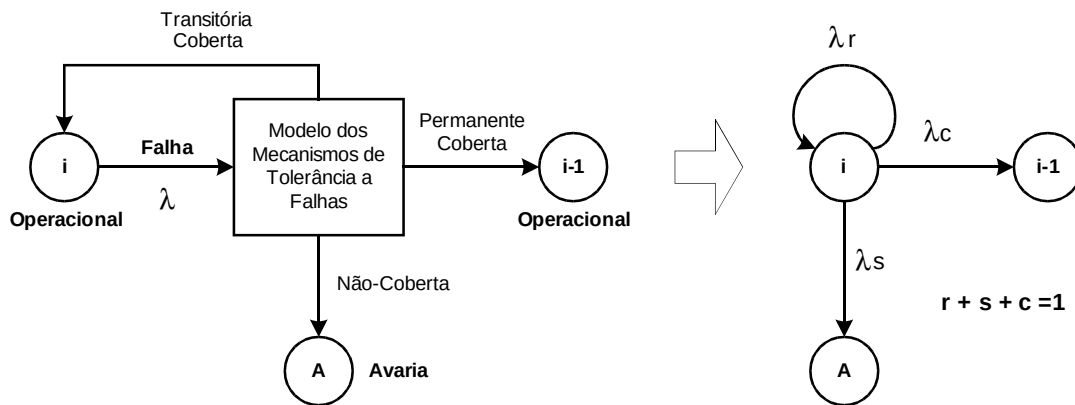


Figura 4.4 - Aplicação da técnica de decomposição comportamental.

Voltando de novo ao modelo, após a ocorrência de uma falha os mecanismos de tolerância a falhas entram em funcionamento, sendo a sua eficácia modelada seguindo o princípio da decomposição comportamental através de factores de cobertura. Este comportamento é modelado através de transições imediatas com duas selecções associadas, cada uma representando respectivamente uma falha coberta ou não-coberta⁵⁸. Se a falha for coberta significa que os mecanismos de tolerância a falhas funcionaram com sucesso (ex. reconfiguraram os componentes

⁵⁸ Note-se que é necessário conhecer os factores de cobertura referentes às várias combinações de falhas: transitórias, permanentes, seguras e não-seguras.

redundantes ou isolaram o componente avariado) e o serviço fornecido, ao nível do componente ou da rede, pode continuar a ser fornecido. As falhas não-cobertas representam o caso oposto.

No caso de uma falha transitória está-se a assumir que a duração da falha tem uma ordem de grandeza que não é superior à execução dos mecanismos de tolerância a falhas, o que é um pressuposto credível⁵⁹ [PCA+96] [KS00]. Assim se a falha for coberta, independentemente de ser potencialmente segura ou não, significa que os mecanismos de tolerância a falhas funcionaram correctamente e que a falha deixou de estar activa. Desta forma o componente retorna ao seu estado operacional. Este comportamento é representado através da escolha da primeira selecção das transições imediatas $hw\_t\_c\_sg$ e $hw\_t\_c\_ns$, que levam à marcação da posição hw . Associadas a estas escolhas estão os factores de cobertura, respectivamente $p\_hw\_t\_c\_sg$ e $p\_hw\_t\_c\_ns$.

No caso da falha transitória ser não-coberta, assume-se que os seus efeitos (erros) se propagam aos componentes hierarquicamente superiores com que o componente de *hardware* interage, que neste caso concreto são os componentes de *software* (este assunto será discutido com maior detalhe durante a apresentação do modelo dos componentes de *software*). Este comportamento é representado através da escolha da segunda selecção das transições imediatas $hw\_t\_c\_sg$ e $hw\_t\_c\_ns$, levando à marcação das posições $hw\_f\_t\_sg$ e $hw\_f\_t\_ns$ respectivamente. Estas representam a interface com o modelo do *software* (SW).

É importante notar que não se assumiu que as falhas nos componentes de *hardware* se propagam a outros componentes de *hardware*. Embora isto fosse perfeitamente possível de modelar, iria originar um modelo demasiado complexo. Além disso, nem sempre é fácil conhecer com o detalhe necessário este tipo de interacção. Note-se no entanto, que este comportamento pode ser modelado de forma indirecta através dos seus efeitos sobre os componentes de *software*.

No caso de falhas permanentes, o comportamento é diferente devido à falha originar a não-operacionalidade do componente. Assim se a falha for segura e coberta, são analisadas previamente um conjunto de condições para determinar se a falha do componente implica, ou não, algum tipo de avaria (segura) da rede. A especificação destas condições depende inteiramente dos requisitos do problema em causa, estando portanto assim em aberto. Tipicamente, estas condições estão relacionadas com a existência de número mínimo de componentes operacionais, para os quais se considera que a rede não está num estado de avaria. Isto implica assim, analisar as condições locais (do nó) e globais (da rede).

⁵⁹ Caso se verificasse a situação oposta, a falha poderia ser classificada como permanente e modelada de forma equivalente. Note-se que a classificação é também função da forma como operam os mecanismos de tolerância a falhas.

Ao nível do modelo, o comportamento anterior é representado através da escolha da primeira selecção das transições imediatas $hw\_p\_c\_sg$ e $hw\_p\_c\_ns$, às quais estão associados os factores de cobertura $p\_hw\_p\_c\_sg$ e $p\_hw\_p\_c\_ns$ respectivamente. A verificação das condições de avaria da rede (locais e globais) é realizada por intermédio da porta o_cg .

Com o objectivo de concretizar estas condições, e sem perda de generalidade, a Tabela 4.2 apresenta um exemplo típico do código (apenas parte) associado a esta porta, quando se considera que o modelo de **HW** representa a totalidade do *hardware* de um nó da rede. Considerando assim este exemplo, inicialmente são verificadas condições locais relacionadas com a existência de eventual redundância local. Neste caso é assumido que enquanto existir redundância local (número de testemunhos em $hw > 0$), o nó continua operacional. Caso estas condições não sejam satisfeitas, são verificadas as condições globais da rede. Neste caso são muito simples, e estão apenas relacionadas com a existência de número mínimo de nós operacionais. Se estas também não forem satisfeitas, ocorre uma avaria segura da rede.

	o_cg
Função	<pre> /* Condições para uma falha coberta segura */ /* Existe redundância local ? */ if (hw->Mark() > 0) { /* Sim */ /* Eventual reparação local */ } else { /* Não */ /* Redundância local esgotada */ /* Actualiza condições globais */ NOS->Mark()++; /* Nós reais */ NOSL->Mark()++; /* Nós lógicos */ /* Verifica as condições globais */ if (NOSL->Mark() > MAXIMO_NOS_AVARIADOS) { AS->Mark() = 1; /* Avaria segura da rede */ } } </pre>

Tabela 4.2 -Exemplo do código da porta o_cg .

A execução do código da porta o_cg pode assim levar à marcação das posições globais *AS*, *NOS* e *NOSL*. A posição *AS* indica que existiu uma avaria segura da rede, a posição *NOS*, representa o número real de nós num estado de avaria, e a posição *NOSL*, representa o número lógico de nós num estado de avaria. A diferença de interpretação entre as duas posições, vem do facto de se considerar que um *cluster* com m nós representar um termos funcionais um único nó (este

aspecto será discutido com maior nível de detalhe quando for apresentado o modelo de reparação da rede de campo).

No caso de falhas permanentes não cobertas (seguras ou não), o comportamento é semelhante aos das falhas transitórias não-cobertas. Assim, assume-se que o seus efeitos (erros) se propagam aos componentes hierarquicamente superiores. Este comportamento é representado através da escolha da segunda selecção das transições imediatas $hw\_p\_c\_sg$ e $hw\_p\_c\_sg$, levando à marcação respectiva das posições $hw\_f\_p\_sg$ e $hw\_f\_p\_ns$.

Como nota final refira-se que todos os parâmetros deste modelo associados às escolhas (probabilidades e factores de cobertura) podem ser definidos como uma função do número de componentes operacionais. Isto permite assim caracterizar de forma mais realista o comportamento do componente.

4.3.2 Modelo SW

O modelo **SW** foi desenvolvido com o objectivo de representar o comportamento de um componente de *software* do nó⁶⁰.

Este modelo deve representar o comportamento do *software* na perspectiva da ocorrência das falhas, e não da componente funcional associada ao controlo do processo em causa. Ou seja, a modelação incide apenas sobre o funcionamento dos mecanismos (de tolerância a falhas) existentes nestes componentes quando ocorrem falhas, e quais as respectivas consequências ao nível do comportamento da rede, e não sob a forma como são executadas as acções de controlo do processo. Neste sentido, o desenvolvimento do modelo reflecte este aspecto.

Ao nível do *software* existem mecanismos de tolerância a falhas de vários tipos. Enquanto que uns actuam localmente no nó, outros estendem a sua influência aos vários nós da rede. Assim e de acordo com esta perspectiva, considerou-se que o componente de *software* pode ser decomposto em 2 níveis hierárquicos: *software local* e *global* (Fig. 4.5).

O *software local* representa o comportamento dos mecanismos de tolerância a falhas cuja actuação se restringue apenas ao nó. Estes dependem das características da locais da aplicação, e estão associados a aspectos tais como: a arquitectura de tolerância a falhas das tarefas que são executadas no nó (*N-Modular Redundancy*, *N-Version Programming*, *Recovery Blocks*, etc.) [AKL90] [DL94], aos mecanismos de detecção de erros utilizados [Pom00] [SBB04], às asserções (*assertions*) sobre o comportamento do sistema [Hil00] [JHC+02], ao sistema operativo de suporte,

⁶⁰ Como será descrito adiante, este modelo vai também ser utilizado (com adaptações) para representar outros componentes de *software* da rede.

etc. O *software global*, representa o comportamento dos mecanismos globais de tolerância a falhas. Estes estão associados aos mecanismos implementados ao nível da aplicação distribuída, ou seja em todos os nós da rede, como por exemplo a replicação de tarefas, a gestão da redundância, a comunicação em grupo, etc. [Mul93] [Jal94] [GS97].

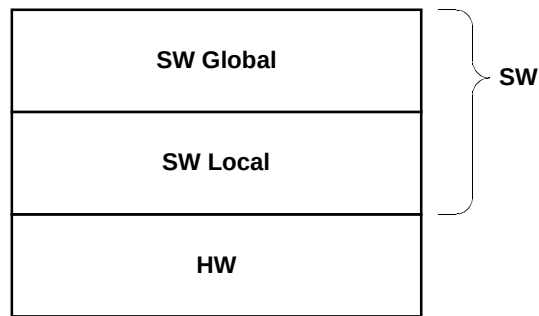


Figura 4.5 - Arquitectura dos modelos HW e SW.

Como os mecanismos de tolerância a falhas existentes ao nível do *software* são específicos de cada aplicação, não é possível dispor de um modelo que descreva todas as situações. Neste sentido, recorreu-se a uma estratégia dupla. Por um lado, utilizar o conceito de modularidade para decompor o *software* nos seus vários componentes constituintes. Por outro lado, modelar o comportamento dos mecanismos de tolerância da falhas existentes nestes componentes utilizando factores de cobertura que são obtidos através da técnica da decomposição comportamental. Em relação a este último aspecto, os motivos para a sua utilização são da mesma natureza dos já discutidos para o modelo **HW**, pelo que se optou por não os referir de novo. Refira-se que tal como anteriormente, estes factores de cobertura podem ser obtidos através da resolução de modelos de avaliação [AKL90] [DL94] [LAB+95], ou através de técnicas de injeção de falhas [Hil00] [CMR+01] [Con03].

A interacção entre os modelos **HW** e **SW** resulta da existência de falhas ao nível do *hardware* que não foram cobertas pelos mecanismos existentes localmente, e cujos efeitos (erros) se vão propagar ao *software*. Estes efeitos, que se vão traduzir também por falhas, vão activar os mecanismos de tolerância a falhas existentes a este nível que por sua vez vão agir em conformidade.

Esta abordagem hierárquica para a interacção entre o *hardware* e o *software*, permite realizar uma decomposição lógica baseada no funcionamento típico deste tipo de componentes. Os mecanismos de tolerância a falhas actuam em cada nível, e no caso de não terem sucesso, a tarefa de tolerar a falha é entregue (em termos lógicos) ao nível imediatamente superior e assim sucessivamente. Este conceito é análogo ao do *componente tolerante a falhas ideal* (*idealised fault tolerant component*) proposto por [LA90] (Fig. 4.6).

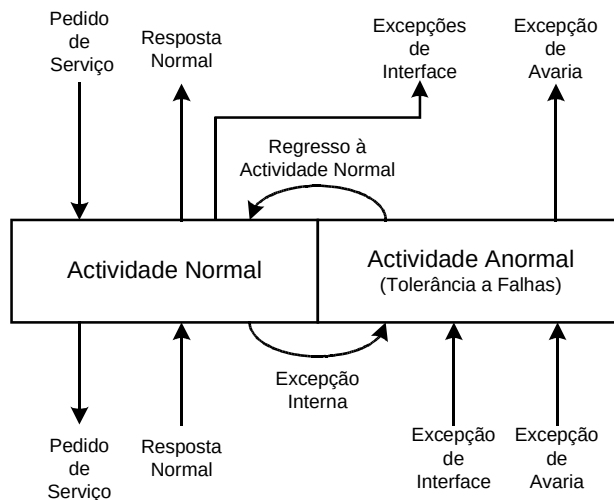


Figura 4.6 - Componente tolerante a falhas ideal.

É importante notar que a execução destes mecanismos podem envolver vários componentes de diferentes tipos e situados a diferentes níveis (ex. os mecanismos implementados ao nível do *software* podem envolver componentes de *hardware* de vários tipos), embora em termos de modelo interesse apenas o seu comportamento conjunto. Daí que o modelo reflecta apenas este aspecto.

Embora o modelo **SW** representado indique graficamente apenas uma interface com o modelo **HW**, na prática o número de interfaces é igual ao número de componentes (modelos) de *hardware* com que o componente de *software* interage. Permite-se assim que o modelo **SW** possa ser adaptado às características específicas de cada componente **HW** com que interage, nomeadamente ao nível dos respectivos factores de cobertura. Desta forma, o modelo completo do *software* irá corresponder à composição de vários modelos com a estrutura apresentada na Figura 4.7.

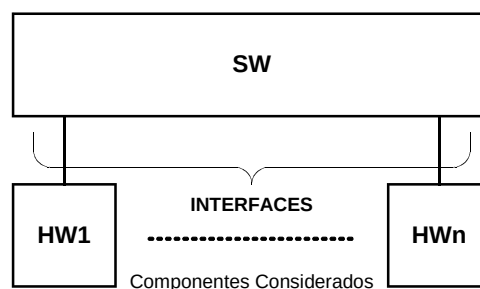
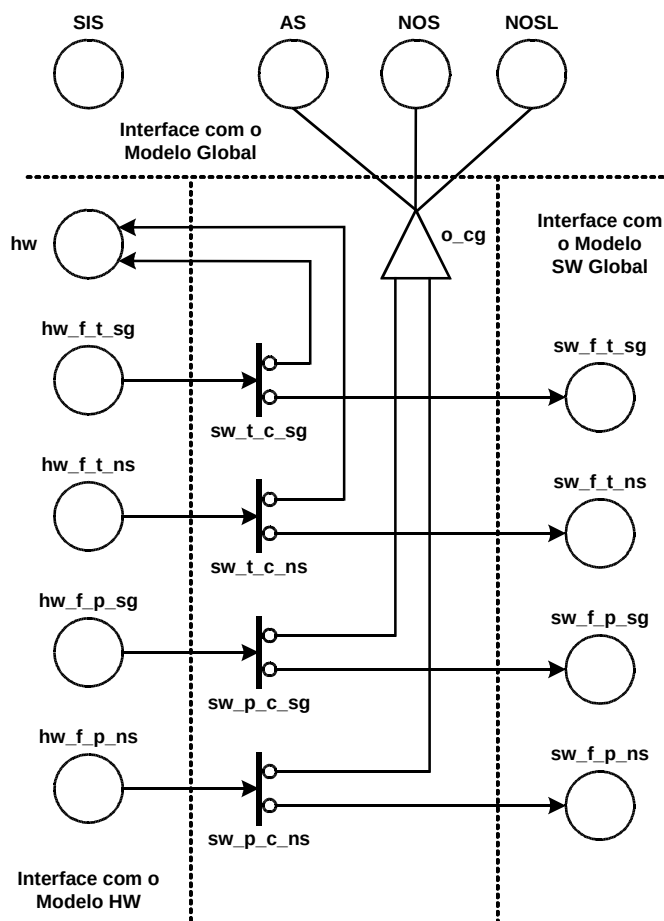


Figura 4.7 - Composição de modelos HW e SW.

Embora não descrita formalmente, esta abordagem pode-se também aplicar internamente ao modelo **SW**, nomeadamente à interacção entre os modelos **SW Local** e **Global**. A definição destas interfaces é apenas função das características específicas da aplicação, sendo desta forma deixada em aberto. Neste sentido, a descrição dos modelos **SW Local** e **Global** pode ser interpretada de duas



No caso de uma falha transitória coberta (segura ou não) o componente retorna ao seu estado operacional. Este comportamento é representado através da escolha da primeira selecção das transições imediatas $sw\_t\_c\_sg$ e $sw\_t\_c\_ns$, que levam à marcação da posição hw . No caso contrário, a falha propaga-se para o componente que representa *software global*. Este comportamento é representado através da escolha da segunda selecção das transições imediatas $sw\_t\_c\_sg$ e $sw\_t\_c\_ns$, levando à marcação respectiva das posições $sw\_f\_t\_sg$ e $sw\_f\_t\_ns$, que representam a interface com o modelo **SW Global**. Associadas a estas transições estão os factores de cobertura $p\_sw\_t\_c\_sg$ e $p\_sw\_t\_c\_ns$ respectivamente.

No caso de falhas permanentes, o comportamento é representado através das transições imediatas $sw\_p\_c\_sg$ e $sw\_p\_c\_ns$, cuja lógica de funcionamento é igual à descrita para o modelo **HW**. Dado o carácter repetitivo da descrição optou-se por não a realizar. Refira-se apenas que os factores de cobertura associados a estas transições são $p\_sw\_p\_c\_sg$ e $p\_sw\_p\_c\_ns$ respectivamente.

A verificação das condições de avaria da rede é realizada por intermédio da porta o_cg , cuja funcionamento foi já discutido no modelo **HW**.

4.3.2.2 Modelo SW Global

O modelo **SW Global** (Fig. 4.9) situa-se hierarquicamente a seguir ao modelo do **SW Local** com o qual partilha uma interface, juntamente com o modelo **HW**.

Os mecanismos globais de tolerância a falhas, estão representados por intermédio das transições imediatas $swg\_t\_c\_sg$, $swg\_t\_c\_ns$, $swg\_p\_c\_sg$ e $swg\_p\_c\_ns$. A estas últimas estão associadas os factores de cobertura $p\_swg\_t\_c\_sg$, $p\_swg\_t\_c\_ns$, $p\_swg\_p\_c\_sg$ e $p\_swg\_p\_c\_ns$ respectivamente. Em termos de comportamento, este é semelhante ao descrito para o modelo **SW Local**, pelo que se optou por não o referir de novo.

As portas de saída o_cg_X , implementam a verificação das condições globais que dão origem a uma avaria da rede (só estão representados na figura os arcos que levam à avaria da rede). Estas condições têm as mesmas características das definidas no modelo **HW**, com uma excepção para o caso das falhas não-seguras e não-cobertas. Como estas não se podem propagar mais (em termos lógicos), vão dar origem a uma avaria não-segura da rede. Este aspecto é modelado através da introdução da posição ANS , que representa esta situação.

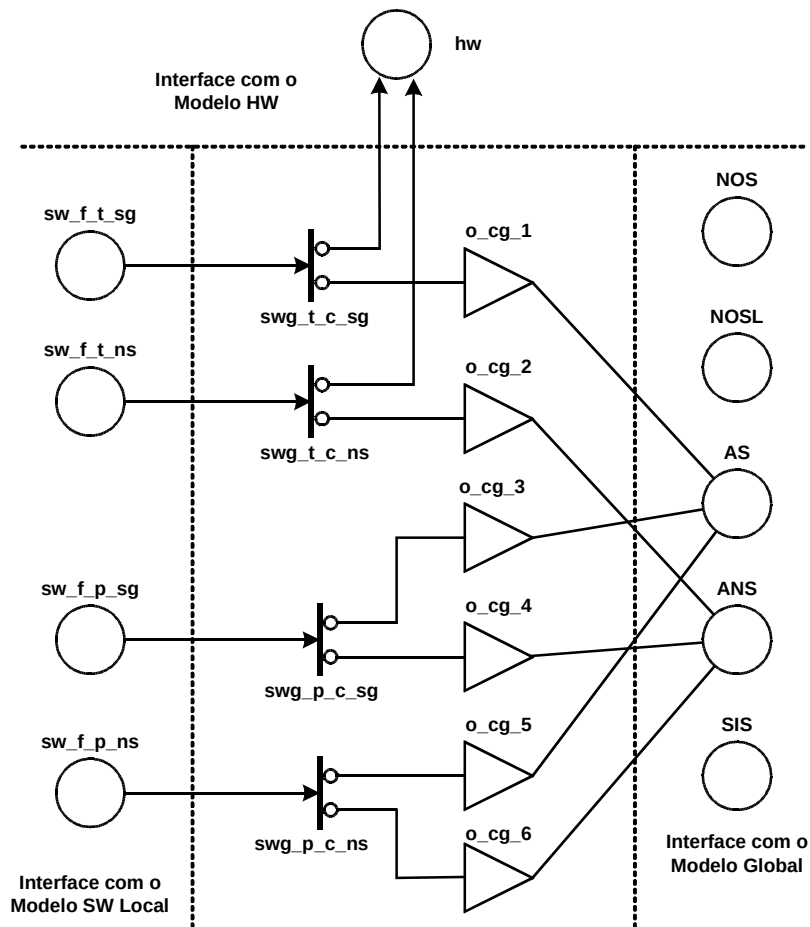


Figura 4.9 - Modelo SW Global.

Para além destas funções, estas portas também realizam também uma “limpeza” da marcação do modelo. Assim, e no caso de ocorrerem avarias da rede, a marcação do modelo é removida (todas as posições ficam sem testemunhos) com o objectivo minimizar o número de marcações intermédias dos modelos, e assim contribuir para a redução do espaço de estados. Um exemplo do código (parte) associado a uma destas portas é apresentado na Tabela 4.3.

o_cg_2	
Função	<pre>/*Falha não-segura e não-coberta */ NOS->Mark()++; NOSL->Mark()++; hw->Mark() = 0; ANS->Mark() = 1;</pre>

Tabela 4.3 -Exemplo do código da porta o_cg_2.

Tendo-se descrito os vários modelos que compõem o modelo do nó, é possível concretizar com maior pormenor as condições que dão origem a uma avaria da rede. Neste sentido, e sem perda de generalidade e utilizando o exemplo definido

para o modelo **HW**, a Figura 4.10 representa o fluxograma que descreve a forma como estas condições foram implementadas nas portas o_cg_X ⁶¹.

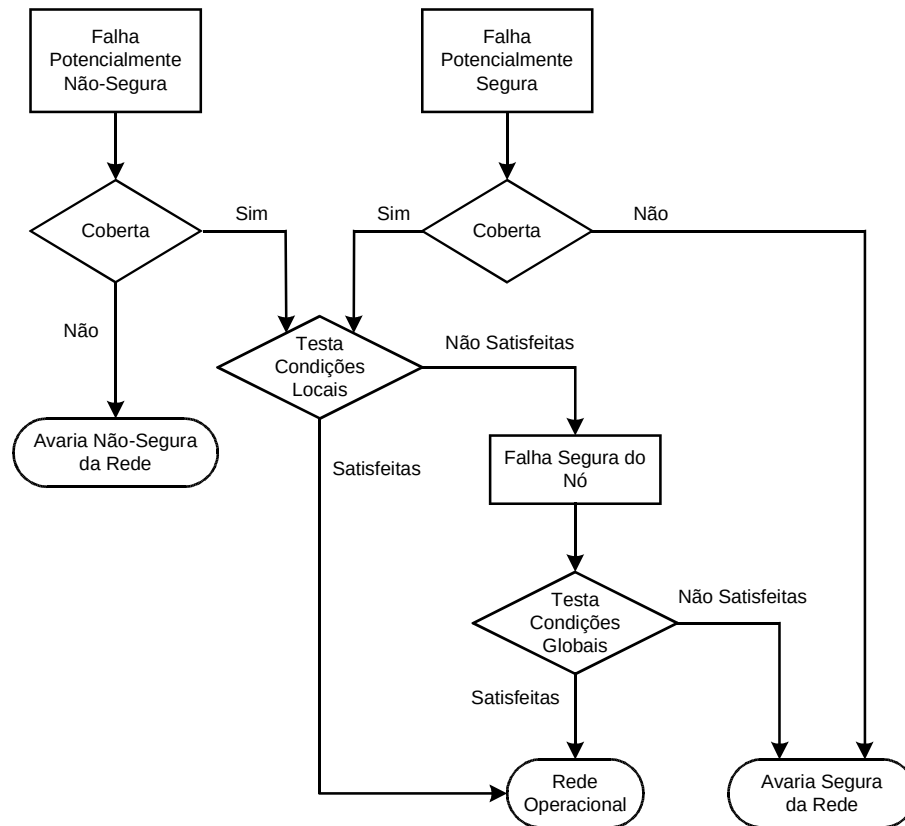


Figura 4.10 - Exemplo das condições de avaria da rede.

Por fim importa fazer algumas considerações sobre o modelo **SW** desenvolvido. Ao contrário do modelo **HW**, este não modela qualquer tipo de recursos locais ao nível do *software*, com por exemplo o comportamento das tarefas que são executadas nos nós. Esta foi um opção deliberada para manter o modelo genérico, dado que o comportamento das mesmas é muito específico da implementação realizada. Neste caso, é necessário conhecer com detalhe as relações funcionais entre as várias tarefas que são executadas nos vários nós para obter um modelo minimamente realista [KB00] [AK02]. A opção utilizada neste modelo, foi a de “condensar” este comportamento através dos factores de cobertura existentes em cada nível. Note-se contudo, que desde que se disponha da informação necessária este comportamento pode ser facilmente incorporado.

⁶¹ Este fluxograma pode também ser adaptado para as portas o_cg definidas nos restantes modelos, com a ressalva de que as falhas não-cobertas se propagam aos níveis superiores, e que portanto não dão logo origem a uma avaria da rede.

4.3.3 Modelo CL

O modelo **CL** foi desenvolvido com o objectivo de representar o comportamento de um *cluster*⁶² de nós da rede de campo.

Um *cluster* é aqui considerado como um agrupamento redundante de nós, que do ponto de vista lógico funciona como um único nó da rede. Quando um dos nós entra num estado de avaria, os serviços do *cluster* são assegurados pelos restantes nós (do *cluster*) operacionais. O objectivo é assim mascarar a avaria de um nó. O número de nós do *cluster* é definido ao nível da especificação do sistema, sendo tipicamente função do número de avarias que sequenciais que se pretende tolerar [LAB+95], dos modos de avaria assumidos para cada nó (*fail-silent*, *fail-stop*, *fail-controlled*, *fail-uncontrolled*, etc.) e do grau de representatividade com que a avaria da rede é coberta pelos modos de avaria assumidos (*assumption coverage*) [Pow95].

O desenvolvimento deste modelo foi baseado na definição de uma arquitectura genérica para o *cluster*, que reflecte a hierarquia dos mecanismos de tolerância a falhas existentes na rede (Fig. 4.11). Ao nível de cada nó, os modelos possuem a estrutura dos modelos **HW** e **SW** apresentados previamente. No entanto, o modelo **SW** foi separado nas suas componentes locais e globais, para permitir a introdução de uma camada adicional, **SW Cluster**, referente aos mecanismos de tolerância a falhas que existem ao nível do *cluster*. Estes mecanismos têm como tarefa agir após a ocorrência de uma falha num dos nós do *cluster* e cujas consequências se façam sentir ao nível do *cluster*, podendo executar funções de reconfiguração dos nós, gestão da redundância (ex. *replica determinism*), etc.

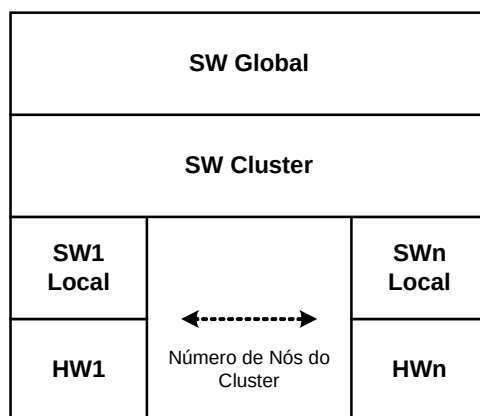


Figura 4.11 - Arquitectura do modelo CL.

Como o modelo resulta da composição dos modelos **HW** e **SW**, cuja apresentação foi já realizada, a sua descrição vai restringir-se apenas à componente referente ao **SW Cluster** (Fig. 4.12).

⁶² Também por vezes denominados FTU (*Fault Tolerant Units*).

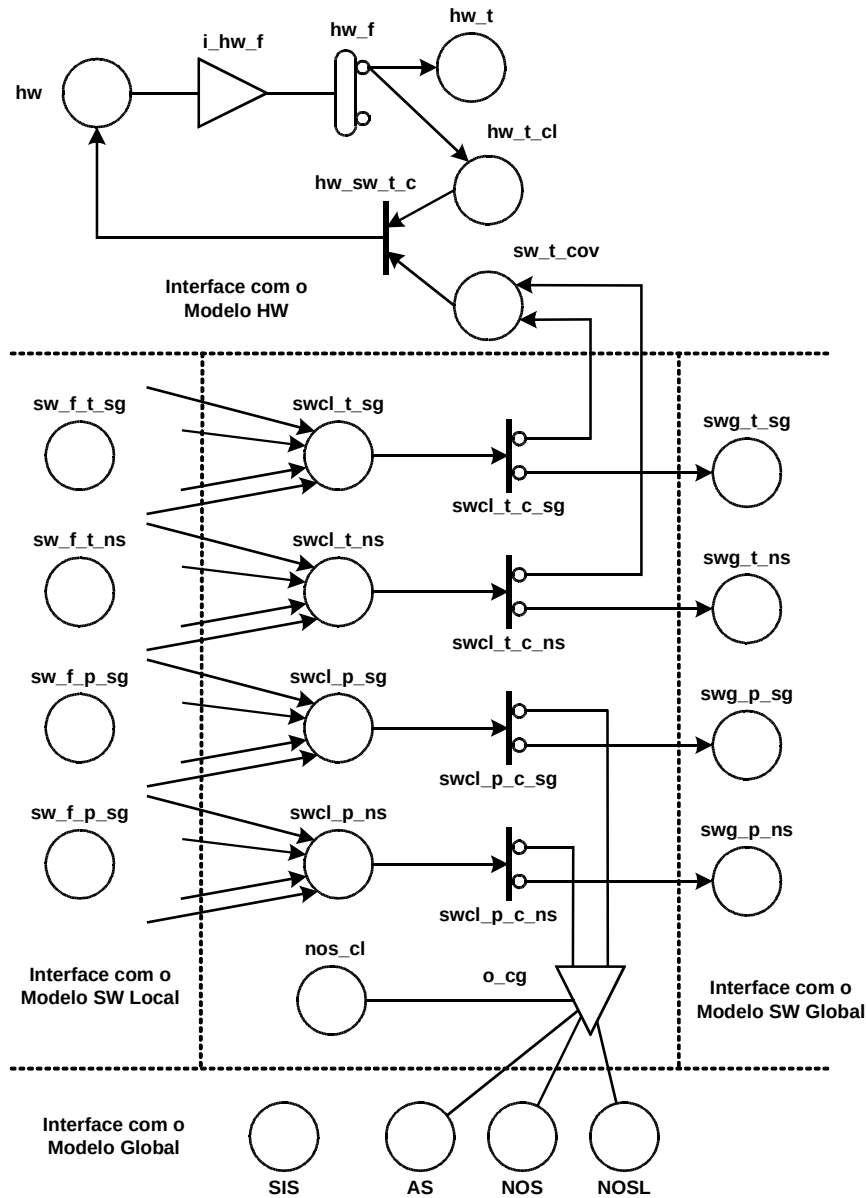


Figura 4.12 - Modelo CL.

As posições *swcl_t_sg*, *swcl_t_ns*, *swcl_p_sg* e *swcl_p_ns* representam a interface com os vários modelos **SW Local**. Cada uma destas posições resulta da sobreposição das várias posições equivalentes referentes aos modelos **SW Local**.

Os mecanismos de tolerância a falhas do *cluster*, estão representados por intermédio das transições imediatas *swcl_t_c_sg*, *swcl_t_c_ns*, *swcl_p_c_sg* e *swcl_p_c_ns*. A estas últimas estão associadas os factores de cobertura *p_swcl_t_c_sg*, *p_swcl_t_c_ns*, *p_swcl_p_c_sg* e *p_swcl_p_c_ns* respectivamente. Em termos de funcionamento, este é semelhante ao descrito para os modelos **SW Local** e **Global**, pelo que se optou por não os referir de novo.

A interface com o modelo **SW Global**, é realizada por intermédio das posições *swg_t_sg*, *swg_t_ns*, *swg_p_sg* e *swg_p_ns*. Note-se no entanto, que no modelo **SW Global** estas posições aparecem com um nome diferente (*sw_f_X_Y*), mas cujo significado é o mesmo.

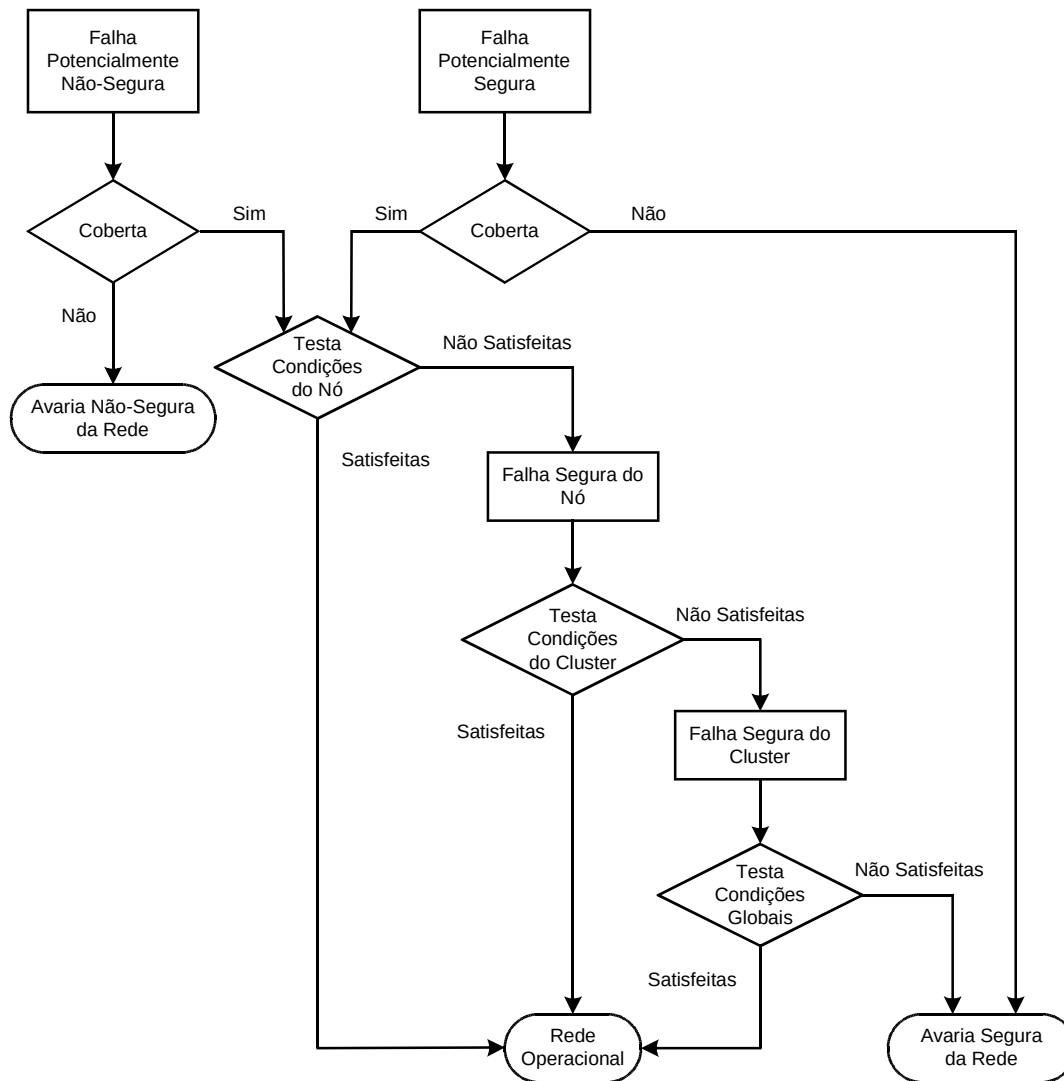


Figura 4.13 - Exemplo das condições de avaria da rede incluindo o cluster.

Como se trata de um *cluster*, as condições de avaria da rede têm que ser ligeiramente reformuladas para incluir os efeitos das falhas dos nós que constituem o *cluster*. Assim, foram incluídas várias posições adicionais. A posição local *nos_cl* é utilizada para indicar dentro do *cluster* quantos nós estão operacionais. Isto é necessário para definir as condições de avaria ao nível do mesmo. A especificação destas condições depende inteiramente dos requisitos do problema em causa, estando portanto assim em aberto. Tipicamente, estas condições estão relacionadas com a existência de número mínimo de nós (do *cluster*) considerados operacionais.

As condições de avaria da rede têm agora que ser equacionadas a três níveis: nó, *cluster* e global. Sem perda de generalidade, e utilizando o exemplo definido anteriormente, a Figura 4.13 representa o fluxograma que descreve a implementação das condições de avaria da rede⁶³.

Como da perspectiva da operação da rede, um *cluster* funciona como um nó lógico, a posição *NOSL* é utilizada para indicar o número de nós lógicos que estão num estado de avaria devido a uma falha permanente (segura ou não). A posição *NOS* tem uma semântica semelhante, mas representa a totalidade de nós (façam ou não parte de *clusters*) que estão nessa situação. A necessidade de manter estas duas posições está relacionada com a reparação e será justificada durante a discussão dos modelos de reparação.

A necessidade de se conhecer em qual dos modelos **HW** ocorreu a falha transitória, implica uma ligeira modificação dos mesmos através da introdução da posição *hw_t_cl*, que memoriza em qual dos modelos (**HW**) a falha ocorreu. Assim, se a falha transitória for coberta ao nível do *software* (**Local**, **CL** ou **Global**), a posição a *sw_t_cov* é marcada e a transição imediata *hw_sw_t_c* dispara, retornando o componente ao seu estado operacional.

4.3.4 Modelo BUS

O modelo **BUS** foi desenvolvido com o objectivo de representar o comportamento da infra-estrutura de comunicações da rede.

O *hardware* associado a esta infra-estrutura pode ser decomposto em componentes locais, que fazem parte dos nós da rede (ex. processadores de comunicações, *transceivers*, conectores, etc.), e em componentes que são globais (comuns) a todos os elementos da rede (ex. meio físico de transmissão). No entanto, ao nível do *software* (*stack* de comunicações) uma decomposição deste tipo já não é possível, dado que este componente (de *software*) faz parte do nó, dependendo além disso para a sua execução de *hardware* existente no próprio nó.

As falhas que afectam esta infra-estrutura têm também efeitos quer locais (no componente), quer globais. Por forma a concretizar melhor este aspecto, nos parágrafos seguintes são analisados os possíveis efeitos das falhas de acordo com a sua persistência temporal e com tipo de componentes afectados.

Ao nível do *hardware*, uma falha permanente pode afectar tanto um componente local ao nó, como um componente global à rede, ou até ambos. Admita-se por

⁶³ Estas condições estão implementadas nas portas *o_cg_X* do modelo **SW Global** do *cluster* e podem ser transpostas para as portas *o_cg* dos restantes modelos do *cluster* de forma semelhante à referida anteriormente.

exemplo uma falha deste tipo num *transceiver* de um nó. Os seus efeitos vão desde o impedimento do nó comunicar com os restantes, até à impossibilidade de todos os nós comunicarem (ex. uma falha do tipo *stuck-at* ou *stuck-open* ao nível da interligação com o meio físico de comunicação). Uma falha transitória por seu lado, afecta tipicamente os dados que são transmitidos entre nós.

Ao nível dos efeitos no *software*⁶⁴, uma falha permanente como impede a existência de comunicações, impossibilita também qualquer tipo de troca de dados com os restantes nós. São assim afectadas as tarefas que são executadas no nó, bem como as tarefas que são executadas nos restantes nós (admite-se uma relação bi-unívoca ao nível da troca de dados entre as tarefas). Além disso, todos os mecanismos de tolerância a falhas globais (também entre *clusters*) podem ser afectados (embora estes estejam, em princípio, preparados para lidar com este tipo de situações). As falhas transitórias, também causam efeitos semelhantes, sendo contudo mais visíveis quando existem requisitos de tempo-real associados à transmissão de mensagens entre tarefas da aplicação distribuída. Tendo em conta que a maioria dos sistemas de controlo suportados por redes de campo possui requisitos deste tipo [Tho99], a discussão centrar-se-á neste tipo de sistemas. Assim e tal como discutido no capítulo 3, o atraso ou a perda de mensagens de controlo, devido aos efeitos de falhas transitórias, podem implicar a violação das respectivas *deadlines*, que por sua vez podem resultar em consequências graves (avarias) ao nível do processo controlado [KS94] [Kop97].

A questão ao nível da modelação torna-se assim complexa, pois ao nível estrutural existem componentes locais e globais, passando-se o mesmo ao nível dos efeitos das falhas. É assim necessário tomar uma opção no que diz respeito ao que se pretende modelar e qual a forma de o concretizar. O problema, grosso modo, pode colocar-se entre incorporar este comportamento nos modelos já desenvolvidos, ou desenvolver um único modelo que descreva este comportamento.

A opção tomada neste trabalho foi a de desenvolver um único modelo, aqui denominado **BUS**. A justificação para esta opção pode ser compreendida através da abordagem utilizada para o desenvolvimento do mesmo:

- ♦ Em relação às falhas permanentes no *hardware*, assumiu-se que estas podem ser decompostas em locais e globais. Os efeitos das falhas locais podem ser incorporados como um modelo do tipo **HW** (e respectivas interacções como modelo **SW**) na construção do modelo referente a um nó, enquanto os efeitos das falhas globais são incorporados como um modelo do tipo **HW** no modelo do **BUS**. O *hardware* aqui definido como global, diz respeito ao meio físico de

⁶⁴ Está-se aqui a utilizar o pressuposto definido previamente de que as falhas afectam de forma directa o *hardware*, podendo os seus efeitos propagar-se eventualmente ao *software*.

transmissão e à componente local aos nós, mas cujo efeito ao nível das falhas é considerado como global⁶⁵.

O pressuposto assumido implica assim uma decomposição entre efeitos locais e globais, o que na prática nem sempre é verdade nem fácil de determinar. Contudo, é um compromisso que se julga razoável face à complexidade do problema. Desta forma, os parâmetros destes modelos devem reflectir este aspecto;

- ♦ Assumir que as falhas transitórias têm apenas efeitos nos componentes globais de *hardware* (da infra-estrutura de comunicações), e que afectam essencialmente a transmissão de mensagens entre tarefas. Estes efeitos podem assim ser incorporados como um modelo do tipo **HW** (e respectivas interações com o modelo **SW**) no modelo do **BUS**. Este pressuposto assenta na lógica de que são os componentes globais que estão mais expostos a este tipo de falhas.

Nas aplicações típicas das redes de campo, os efeitos deste tipo de falhas produzem as maiores consequências para o funcionamento do sistema quando afectam a transmissão de mensagens utilizadas para o controlo do processo, nomeadamente quando implicam a violação das respectivas *deadlines*. Este pressuposto foi assim assumido com base no grau do impacto sobre o funcionamento da rede.

- ♦ O comportamento do *software* associado à infra-estrutura de comunicações deve modelado a partir de uma perspectiva global à rede, dado que do ponto de vista dos pressupostos assumidos ao nível do *hardware* para o modelo **BUS**, os efeitos das falhas (permanentes e transitórias) vão-se fazer sentir ao nível global da rede.

Face ao exposto, o modelo **BUS** referente à estrutura de comunicações têm uma estrutura de acordo com a Fig. 4.14.

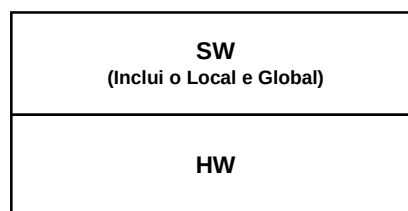


Figura 4.14 - Arquitectura do modelo BUS.

Embora o modelo **BUS** seja a composição dos modelos do tipo **HW** e **SW** apresentados anteriormente, existem várias diferenças ao nível da interpretação, quer das posições de rede de Petri, quer dos parâmetros associados.

⁶⁵ Por questões de simplicidade de apresentação, estes componentes são deste ponto em diante identificados com o meio físico de transmissão.

A componente **HW** do modelo representa as falhas permanentes globais e as falhas transitórias que afectam a infra-estrutura de comunicações. Os factores de cobertura de ambas, devem caracterizar a capacidade dos mecanismos de tolerância a falhas existentes ao nível do *hardware* para lidar com este tipo de situações. Assim e em relação às falhas permanentes, estes mecanismos reflectem por exemplo a gestão da redundância dos meios físicos de transmissão. Concretizando através de um exemplo, admita-se que o meio físico é redundante bem como os *transceivers* nos nós. Quando ocorre uma falha permanente, um dos meios fica não-operacional e é necessário, ao nível do *hardware*, passar a utilizar apenas um meio de transmissão. A capacidade de realizar de forma global e correctamente esta operação de reconfiguração define o factor de cobertura respectivo.

No que diz respeito às falhas transitórias ao nível do *hardware*, estes mecanismos estão relacionados com os efeitos das blindagens e das protecções relativas às interferências electromagnéticas. Em relação à potencial gravidade das falhas (seguras ou não-seguras), só tem sentido considerar este aspecto ao nível das falhas permanentes, dado que nas falhas transitórias a gravidade pode ser definida ao nível do modelo **SW**. É assim necessário remover parte da estrutura do modelo (de **HW**).

A componente **SW** do modelo resulta da compactação dos modelos **SW Local** e **Global** num único modelo (com uma estrutura igual a **SW Global**). Os factores de cobertura associados às falhas permanentes globais, devem reflectir a capacidade global da rede para lidar com este tipo de falhas. Concretizando de novo através do exemplo anterior, quando ocorre uma falha permanente não-coberta no *hardware* é necessário que ao nível global os vários nós sejam capazes de lidar simultaneamente e da mesma forma com esta ocorrência. Embora isto seja assegurado pelas *stacks* de comunicação existentes em cada nó, o que interessa é o seu efeito global e não local (ex. alguns nós conseguem lidar com o problema e outros não).

Os factores de cobertura associados às falhas transitórias no modelo **SW**, resultam directamente da violação das *deadlines* associadas às mensagens de controlo do processo que são transmitidas entre as várias tarefas da aplicação distribuída. A determinação destes parâmetros pode ser realizada também através da utilização uma estratégia baseada na *decomposição comportamental*. Como a escala temporal associada às transmissões das mensagens é várias ordens de grandeza inferior aos restantes parâmetros do modelo (está-se a excluir obviamente a taxa de ocorrência das próprias falhas transitórias), o respectivo comportamento pode ser analisado em separado num outro modelo, e posteriormente incluído como um factor de cobertura⁶⁶.

⁶⁶ Note-se que se considerar que este tipo de falhas podem causar avarias seguras ou não-seguras, é necessário obter para o modelo das falhas transitórias durante as comunicações dois factores de

Entre as hipóteses assumidas nesta abordagem, está o pressuposto que existe um elevado grau de independência face aos restantes subsistemas. Esta é um aspecto que necessita de ser devidamente justificado. Em termos absolutos não é possível afirmar que o comportamento face à ocorrência deste tipo de falhas é independente do estado dos restantes subsistemas da rede. Para confirmar este aspecto, basta considerar um cenário em que a rede está operacional, mas em que alguns dos seus nós estão não-operacionais (ex. avarias seguras dos nós). Neste caso, os nós nesta situação não contribuem para a transmissão de mensagens, o que implica que os resultados obtidos dos modelos das falhas transitórias não reflectam este aspecto (porque estes foram, em princípio, desenvolvidos tendo em conta todas as mensagens existentes), além de que as *deadlines* associadas às mensagens deixam de ser cumpridas, implicando assim uma eventual avaria da rede.

A discussão anterior parece assim indicar que existe um forte grau de dependência entre os modelos, o que inviabiliza qualquer abordagem baseada na decomposição comportamental. No entanto, se o problema for analisado de uma outra perspectiva, é possível concluir que este tipo de abordagem é perfeitamente razoável. Utilizando o exemplo acima citado, os efeitos das avarias dos nós sobre as *deadlines* das mensagens podem ser incluídos directamente nos modelos dos nós (ou *clusters*), através da definição das condições de avaria da rede. Ou seja, quando se dá uma falha de um nó (ou de um *cluster*) são tomados em conta também as respectivas consequências, ao nível das *deadlines* das mensagens, para a definição do estado de avaria da rede. Concretizando com um exemplo: a falha de um nó desde que segura, não implica, no contexto anterior, qualquer avaria da rede. No entanto a falha de dois nós, implica a avaria da rede.

Está assim parte do problema resolvido. A questão que falta discutir diz respeito ao facto do modelo das falhas transitórias durante as comunicações considerar todo o tráfego de mensagens, quando em termos reais a rede pode estar operacional com alguns nós no estado de avaria (segura). Esta parte do problema pode ser resolvida de duas formas. A mais complexa, implica a análise de n modelos de falhas transitórias durante as comunicações, cada um deles reflectindo o caso em que um dos nós (ou *cluster*) está num estado de avaria. Utilizando a soluções de cada um destes modelos, é possível definir o factor de cobertura das falhas transitórias associadas ao modelo **BUS**, como uma função dos factores de cobertura obtidos em cada um dos modelos indicados. Em termos de implementação isto é fácil de realizar, dado que é possível (ao nível do modelo **BUS**) definir este factor de cobertura como função do número de nós operacionais (posições *NOS* e *NOSL*). No entanto, esta abordagem obriga à resolução de n modelos distintos, o que nem sempre é eficaz de realizar, principalmente quando a solução dos modelos é obtida através de simulação.

cobertura, cada um referente ao tipo de avaria considerado.

A segunda forma de resolver este problema é mais pragmática. Se se assumir que os tempos correspondentes à não-operacionalidade dos nós são muito baixos (ex. são reparados rapidamente), é possível concluir que na esmagadora maioria do tempo a rede vai estar com todos os seus nós operacionais. Ou seja, em termos estocásticos isto é equivalente a considerar que em regime estacionário o comportamento da rede (em cenários de falhas transitórias durante as comunicações) pode ser perfeitamente descrito por apenas um único modelo que considera todo o tráfego normal da rede. Embora isto seja uma aproximação, é evidente que o erro cometido é muito pequeno. Nos capítulos 5 e 6 serão analisados respectivamente os comportamentos das redes Profibus e CAN de acordo com esta perspectiva.

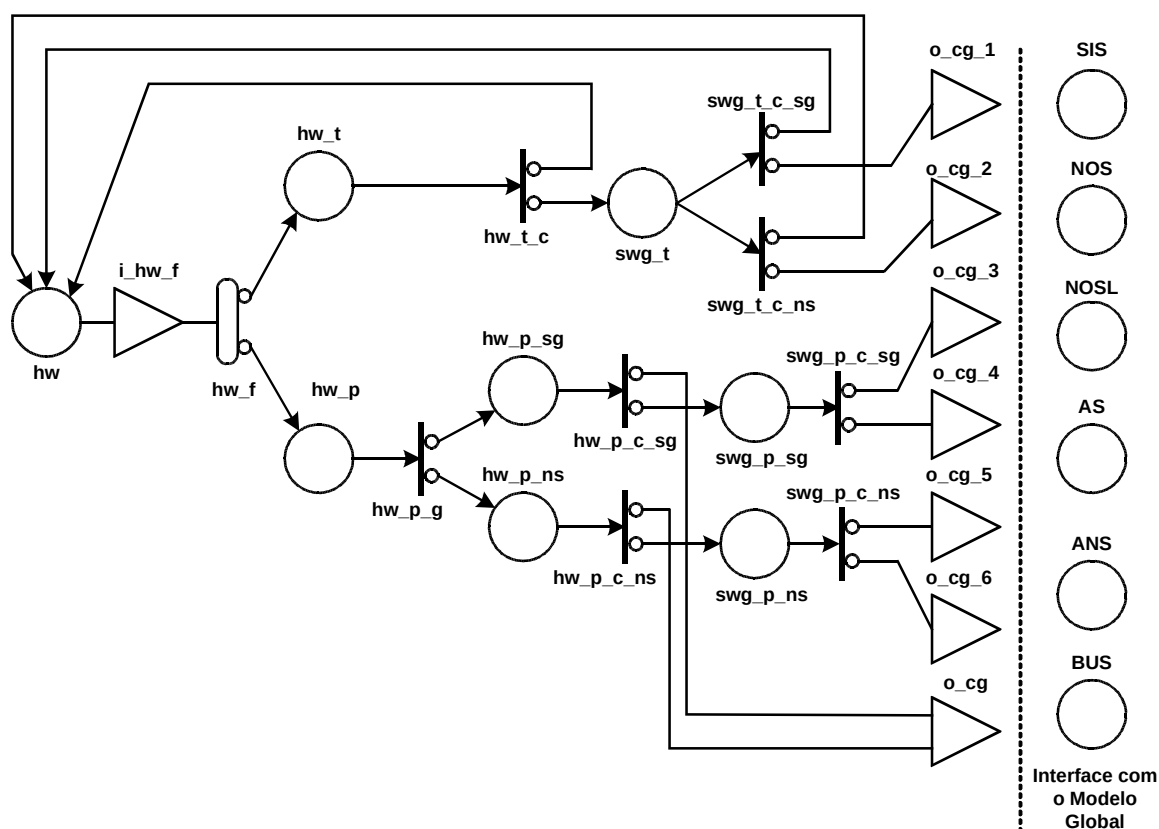


Figura 4.15 - Modelo BUS.

A Fig. 4.15 apresenta o modelo **BUS**. Dado que este resulta da composição dos modelos **HW** e **SW**, optou-se quer por não realizar a sua descrição. Além disso manteve-se a simbologia associada aos mesmos por forma a facilitar a compreensão (foram também eliminados os arcos de ligação ao modelo global). No contexto da discussão sobre a semântica das posições e transições, a posição *hw* indica a redundância ao nível dos meios físicos de transmissão existentes. No mesmo sentido, foi acrescentada a posição global *BUS*⁶⁷ para indicar o número de meios físicos de transmissão que estão num estado não-operacional. A definição das

⁶⁷ Esta posição é redundante e a sua inclusão só se justifica por uma questão de dispor de um modelo com características estruturais semelhantes aos apresentados anteriormente.

condições de avaria da rede neste modelo, são assim função quer da redundância existente (*hw*), quer do número de meios físicos não-operacionais (*BUS*).

4.3.5 Modelos das Reparações

Após uma falha permanente num componente ou uma avaria da própria rede, pode ser necessário uma reparação. Partindo da estrutura da rede, foram identificados dois níveis possíveis para as reparações:

- ◆ **Local**

Esta situação ocorre quando um componente está num estado de avaria (segura), mas a rede continua num estado operacional. Embora fosse possível definir modelos de reparação ao nível dos componentes internos do nó (ou *cluster*) ou da infra-estrutura de comunicações, na prática esta situação não é comum. Assim, assumiu-se que as reparações locais ocorrem sempre ao nível do nó ou da infra-estrutura de comunicações e de forma completa;

- ◆ **Global**

Esta situação ocorre quando existe uma avaria (segura ou não-segura) da rede, e resulta na reparação completa de todos os componentes que se encontram num estado não-operacional.

4.3.5.1 Reparação Local

A reparação local inicia-se após a falha permanente e coberta (potencialmente segura ou não) de um componente⁶⁸, ou seja, o componente avariou de forma segura (Fig. 4.16). Assume-se que a reparação pode ser efectuada durante a operação da rede, e que a posterior reintegração do componente em causa se realiza sempre com sucesso. Refira-se contudo, que este pressuposto pode ser facilmente modificado se forem assumidas outros tipos de condições para a reparação e reintegração do componente (ex. é necessário suspender a operação da rede).

Quando ocorre a situação anterior a transição *hw_p_c_XX* dispara, e a porta *o_cg* é executada. Entre as várias verificações efectuadas por esta porta (e já descritas), estão também as relacionadas com as acções de reparação do componente. Assim, se a falha não originar uma avaria da rede então a reparação do componente é eventualmente possível. O termo eventual, vem do facto de que pode apenas ser desejado (ao nível da especificação da política de reparação) que a reparação se realize só a determinados níveis. Por forma a concretizar este aspecto, admita-se que o modelo **HW** representa um sensor de um nó. Ao nível da especificação pode ser considerado que só após a falha do nó em si (a avaria do sensor não implica a

⁶⁸ Entenda-se aqui por componente o nó ou a infra-estrutura de comunicações, de acordo com o caso.

avaria do nó) é que este se efectua a respectiva reparação (incluindo todos os seus componentes que estão num estado de avaria).

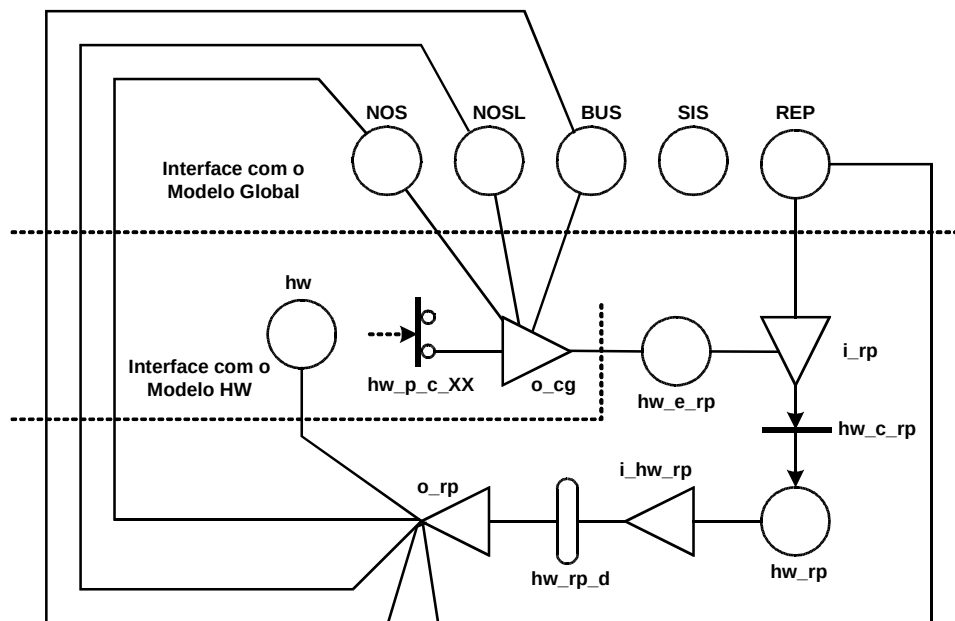


Figura 4.16 - Modelo da reparação local.

Admitindo que estas condições estão satisfeitas, a posição *hw_e_rp* é marcada indicando que o componente está avariado e à espera de reparação. A posição global *REP* indica (através da sua marcação) o número de recursos disponíveis para reparação existentes no momento. Estes recursos são definidos em função das características do sistema em causa. Tipicamente representam o número de pessoas, ou de equipas, que existem para realizar as operações de reparação.

Se o número mínimo de recursos existir (verificada pela porta de entrada *i_rp*) a transição imediata *hw_c_rp* dispara, removendo os testemunhos respectivos de *hw_e_tp* e *REP* (tantos quantos exija esta operação de reparação) e a posição *hw_rp* é marcada. Esta posição representa o início da reparação do componente. A reparação propriamente dita é modelada através da transição exponencial *hw_rp_d*, cuja duração (ou taxa) corresponde ao tempo necessário para se realizar a reparação (que pode ser definida como função do tipo de componente avariado). A porta *i_hw_rp* permite desabilitar esta transição no caso de uma avaria da rede (posição *SIS* marcada).

A utilização de uma distribuição exponencial poderá não ser uma representação muito realista, sendo mais comum utilizar outras distribuições (ex. Weibull). No entanto, este problema pode ser contornado utilizando como por exemplo uma aproximação por fases como descrita no capítulo 2. Note-se no entanto, que neste caso é necessário ter algum cuidado em relação ao crescimento do espaço de estados. Existe contudo um caso particular que permite representar a duração desta

transição através de uma distribuição generalizada. Assim, se o número de recursos de reparação for unitário, só vai existir em cada marcação no máximo uma reparação a ser executada de cada vez. Ou seja, só uma transição do tipo *hw_rp_d* está habilitada (podem estar habilitadas outras transições exponenciais). Obtêm-se assim um processo Não-Markoviano cuja solução é possível de obter por meios analíticos.

Após a reparação terminar, a transição dispara e a porta de saída *o_rp* é executada. Esta porta para além de colocar o componente no seu estado operacional, através da marcação de *hw*, e de repor os recursos utilizados na reparação, através da marcação de *REP*, actualiza também o número de elementos operacionais da rede através da marcação das posições *NOS*, *NOSL* ou *BUS*, de acordo com o tipo de componente em causa.

4.3.5.2 Reparação Global

O modelo de reparações globais (Fig. 4.17) é utilizado com uma dupla funcionalidade. Por um lado modelar o comportamento ao nível da reparação global, e por outro servir como modelo global de interface entre os restantes. Desta forma, onde nos modelos anteriores se faz referência a *modelo global* entenda-se que se está a referir a este modelo.

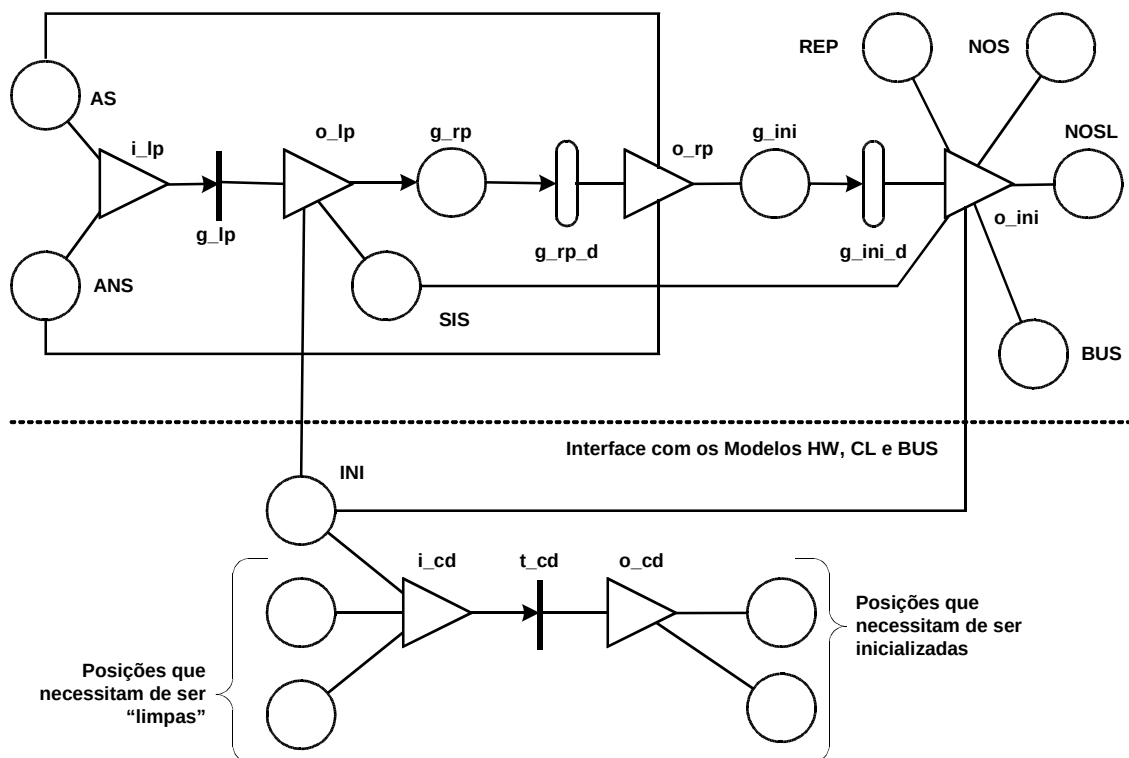


Figura 4.17 - Modelo da reparação global.

A reparação global inicia-se logo após uma avaria da rede (segura ou não), indicada através da marcação das posições *AS* ou *ANS*. O teste destas condições é realizado pela porta de entrada *i_lp*, que por sua vez permite o disparo da transição imediata *g_lp* e da execução da porta de saída *o_lp*. Como resultado desta sequência, as posições *SIS* e *g_rp* são marcadas. A marcação da posição *SIS* vai desabilitar todas as outras transições temporizadas existentes nos vários modelos. Assume-se que a reparação global tem prioridade sobre eventuais reparações locais.

Uma das questões levantadas pela utilização de uma abordagem hierárquica, tem a ver com o caso de uma acção (ex. disparo de uma transição) no modelo de mais alto nível necessitar de modificar a marcação nos modelos de mais baixo nível (entenda-se aqui o termo nível como hierárquico). Enquanto que num modelo monolítico, basta uma única transição imediata para realizar esta modificação, num modelo hierárquico é necessário utilizar uma outra abordagem. Esta, tipicamente, passa por utilizar uma posição comum a todos os modelos e definir, em cada modelo, uma transição imediata associada a essa posição que tem como objectivo realizar a alteração local da marcação do modelo. Embora isto seja simples de implementar, dá origem a dois problemas que têm que ser devidamente acautelados. O primeiro, e mais importante, tem a ver com a possibilidade do processo estocástico subjacente não estar correctamente especificado (*well specified*) [SM91a]. Isto deve-se ao facto das várias transições imediatas estarem habilitadas simultaneamente, dando assim origem a um comportamento, ao nível do seu disparo, não especificado à partida. Este problema é especialmente grave se associado a essas transições imediatas estiverem definidos impulsos de recompensa (*impulse reward*).

O segundo problema, tem a ver com o processo de geração automática do espaço de estados. A existência de muitas transições habilitadas simultaneamente vai obrigar a ferramenta a gerar todos as sequências possíveis referentes aos disparos das transições. Se o número de transições for pequeno, o impacto é reduzido. Contudo, para valores elevados o tempo utilizado para realizar esta tarefa pode ser muito considerável.

Ao nível do modelo proposto temos um problema deste tipo. Quando ocorre uma avaria da rede, os vários modelos que descrevem o comportamento da rede podem estar numa marcação qualquer. Neste contexto, guardar a marcação desses modelos é irrelevante, porque estas em nada vão contribuir para a evolução do processo (note-se que se impede a ocorrência de falhas durante a avaria da rede: posição *SIS* marcada). Além disso, esse conjunto de marcações intermédias vai fazer aumentar o espaço de estados (todas as combinações possíveis). É assim necessário ao nível da modelação impedir que isto aconteça.

A abordagem utilizada para resolver este problema foi semelhante à anteriormente descrita, mas com algumas diferenças. Numa primeira fase admita-se que o tempo necessário para gerar os espaço de estados não é problemático. Neste sentido, é definida uma posição global comum a todos os modelos, à qual está associada, em cada modelo hierarquicamente inferior, uma transição imediata. Posteriormente à marcação desta posição global, é necessário definir um mecanismo que garanta que as várias transições imediatas disparam apenas uma única vez. A implementação deste mecanismo, baseou-se no conhecimento das condições locais de cada modelo. Assim, em cada modelo vai ser verificado se existem posições marcadas, e em caso afirmativo a marcação é removida. Posteriormente à remoção da marcação, a transição imediata associada a esse modelo deixa de estar habilitada. No fim do disparo de todas as transições imediatas, todos os modelos ficarão com uma marcação nula. Repare-se que no fim desta operação de “limpeza” dos modelos, a marcação global do modelo está perfeitamente definida (todos os modelos estão com uma marcação nula excepto o global), funcionado assim como um “salto” instantâneo para uma nova marcação. Como não faz sentido definir quaisquer impulsos de recompensa associados a estas transições imediatas e como a marcação final está bem definida, o processo estocástico subjacente está bem especificado.

Voltando ao modelo da Fig. 4.17, quando a porta de saída o_{lp} for executada as posições g_{rp} e INI vão ser marcadas, sendo esta última associada à “limpeza” dos modelos. Como as transições imediatas têm prioridade sobre as temporizadas, as transições t_{cd} existentes nos restantes modelos (**HW**, **CL** e **BUS**) vão disparar primeiro, e vão remover a marcação desses mesmos modelos (note-se que a ordem pela qual estas operações são realizadas é irrelevante). A habilitação da transição é realizada pela porta i_{cd} . Note-se também, que o modelo de actualização das marcações não faz parte do modelo global, mas sim dos restantes modelos.

No final desta sequência de disparo a transição temporizada g_{rp}_d fica habilitada. Esta transição vai representar o tempo associado à reparação global da rede. Ao contrário do que pode acontecer com as reparações locais, é possível utilizar uma distribuição generalizada para representar a duração desta reparação, porque esta é a única transição temporizada habilitada nesta marcação. Passa-se assim a obter um processo estocástico Não-Markoviano. Por forma a introduzir maior realismo na modelação, também é possível definir a duração da reparação como uma função do número de componentes avariados, dada pelas marcações de *NOS* e *BUS*, e do número de recursos de reparação disponíveis, dada pela marcação de *REP*. Daqui se justifica a necessidade de utilizar as duas posições *NOS* e *NOSL* associadas aos nós não-operacionais. A utilização de apenas *NOSL* embora permitisse conhecer o número lógico de nós não-operacionais, não permitia conhecer o número de nós efectivamente não-operacionais. O raciocínio inverso aplica-se à posição *NOS*.

Terminada a reparação a porta de saída o_{rp} é executada. As posições AS e ANS passam a ter a sua marcação removida (a avaria da rede foi reparada), e a posição g_{ini} é marcada. Esta posição vai representar o início da fase de inicialização da rede após a reparação. O tempo associado a esta inicialização é definido pela transição temporizada $g_{ini}_d^{69}$ (que pelos motivos anteriormente citados pode também ser modelizada por uma distribuição generalizada). Após o seu disparo a porta o_{ini} é executada e a marcação inicial da rede é reposta. Esta passa por inicializar REP com o número de recursos de reparação existentes, remover as marcações de NOS , $NOSL$, BUS e SIS (habilitando assim as restantes transições temporizadas do modelos) e inicializar a marcação dos vários modelos hierarquicamente inferiores. Para este último caso, utilizou-se uma abordagem semelhante à “limpeza” dos modelos, utilizando para isso um modelo com as mesmas características⁷⁰. A posição INI é marcada e a transição t_{cd} dispara, inicializando as posições locais, através da execução da porta de saída o_{cd} . Para evitar disparos consecutivos destas transições dentro do mesmo modelo, a transição é desabilitada logo após o modelo ser inicializado, utilizando para isso informação local. Como agora não é necessário retornar ao modelo global, a posição INI é marcada com o número de modelos a actualizar (isto é conhecido à partida), cada vez que uma transição t_{cd} dispara a marcação da posição INI é decrementada de 1 testemunho, permitindo assim que no disparo da última transição a respectiva marcação seja nula.

Quando o tempo necessário para gerar o espaço de estados se torna significativo devido à análise de todas as sequências possíveis de disparo das transições imediatas, torna-se necessário adoptar uma outra estratégia. Como o problema está no número de transições imediatas simultaneamente habilitadas, poder-se-ia pensar que bastaria ligar em série (utilizando posições globais) os vários modelos de actualização de marcações. A ideia seria actualizar (“limpar” ou inicializar) de forma sequencial cada modelo, cujo último retornaria ao modelo global. Embora isto fosse possível de implementar, originaria problemas ao nível do espaço de estados. Como foi discutido previamente, a técnica de minimização do espaço de estados implementada pela ferramenta baseia-se no facto de existirem simetrias entre modelos, concretizadas através de réplicas dos mesmos. Como a utilização de réplicas, não permite distinguir os modelos entre si (não é possível distinguir posições comuns entre réplicas, ou seja, a posição é comum a todas as réplicas), não é possível realizar a modelação desta forma sem prejuízo do espaço de estados.

Face a este problema, adoptou-se uma abordagem mais simples que passou por não realizar as operações de “limpeza” dos vários modelos. Embora isto resulte num aumento do espaço de estados, verificou-se na prática que devido à simetria dos

⁶⁹ Se necessário, esta actividade pode ser incluída como fazendo parte da reparação, obtendo-se assim um modelo mais simples.

⁷⁰ Como estruturalmente são iguais optou-se por representar apenas um dos modelos.

modelos, acaba por resultar apenas num pequeno acréscimo. Subjacente a esta abordagem, está o pressuposto que o número de marcações distintas dos vários modelos após a ocorrência de uma avaria da rede é reduzido. De facto, se a redundância existente nos modelos for pequena, este pressuposto acaba por se verificar. Como o espaço de estados resulta do número de modelos que estão numa determinada marcação, o impacto é pequeno.

As modificações em relação ao modelo proposto (Fig. 4.18) passam por remover completamente o modelo global apresentado, passando apenas este a ser constituído por apenas as posições globais. Ao nível dos restantes modelos, a actualização das marcações passa a ser realizado após a reparação local.

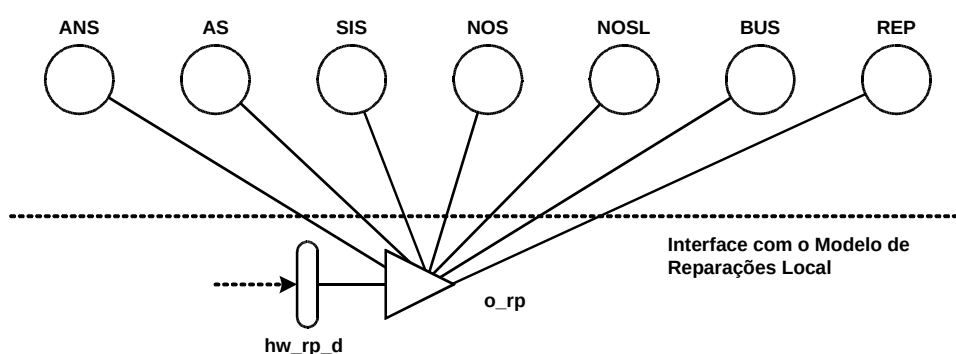


Figura 4.18 - Versão alternativa do modelo global.

Desta forma, quando ocorre uma avaria da rede (posição *SIS* marcada⁷¹) vai-se permitir (ao contrário do modelo anterior) que as várias reparações locais decorram em simultâneo (pode ocorrer uma avaria da rede durante a reparação local de um componente). Quando uma reparação termina, a porta de saída *o_rp* é executada e para além da inicialização local do modelo, são também actualizadas e verificadas as condições globais. Quando ocorrer a última reparação (*NOS*, *NOSL* e *BUS* têm marcação nula) a marcada da posição *SIS* é removida, permitindo assim a re-inicialização de todos os modelos. Neste caso são também inicializadas as marcações das posições *ANS*, *AS* e *REP*. Como é possível distinguir entre uma reparação local ou global (através do posição *SIS*) a taxa da transição *hw_rp_d* pode ser ajustada por forma a reflectir estas diferenças.

Em termos de funcionamento, isto corresponde a partir de um estado de avaria da rede até atingir o estado de total operacionalidade, passando por um conjunto de estados intermédios referentes às reparações locais de cada um dos componentes. Isto permite até que em alguns casos o aumento do espaço de estados seja reduzido. Suponha-se que se define que a rede é considerada operacional se existirem menos do que *m* nós num estado de avaria (segura). Nesta situação, no momento em que num estado intermédio de reparação seja atingido este valor, a

⁷¹ Ao contrário do modelo anterior, a posição *SIS* passa agora a ser marcada em cada modelo através das portas *o_cg*.

rede é colocada imediatamente num estado operacional e as restantes reparações continuam como se fossem locais, fazendo parte assim do espaço de estados natural do modelo.

A utilização desta abordagem implica contudo que as transições hw_rp_d sejam do tipo exponencial. Além disso tem subjacente a hipótese Markoviana, de ausência de memória em relação às eventuais reparações locais em curso. É também necessário ter em conta que nos estados de reparação intermédios, acima referidos, a taxa efectiva de reparação é a soma das existentes. Através da utilização da marcação das posições *NOS* e *BUS* é possível ajustar dinamicamente esta taxa, permitindo assim que a reparação global seja equivalente a uma distribuição generalizada de Erlang. Esta distribuição é bastante flexível, e pode ser utilizada para realizar uma aproximação a outras distribuições.

Como nota final, é importante referir que através de um pequeno modelo auxiliar inserido no modelo global, é possível representar comportamentos relacionados com a manutenção da rede. Manipulando a marcação da posição *REP*, é possível implementar esquemas de manutenção de vários tipos, incluindo o periódico (de forma aproximada).

4.3.6 Avarias em Modo-Comum

As avarias em modo-comum caracterizam-se pela avaria simultânea de vários componentes do mesmo tipo (Fig. 4.19). Neste modelo assumiu-se que este tipo de falhas⁷² têm características e efeitos semelhantes às falhas permanentes.

A modelação deste comportamento pode ser realizada a diferentes níveis:

- ♦ **Local**, afectando apenas um grupo local de componentes;
- ♦ **Nó**, afectando um nó completo;
- ♦ **Cluster**, afectando um *cluster* completo;
- ♦ **Comunicações**, afectando a infra-estrutura de comunicações por completo;
- ♦ **Global**, afectando quer nós (lógicos), quer a estrutura de comunicações da rede.

Por uma questão de espaço de apresentação, optou-se apenas por descrever o comportamento local, embora como poderá ser observado da descrição, o modelo pode ser facilmente transposto para outros níveis.

⁷² Está-se a adoptar a terminologia adoptada no capítulo 3.

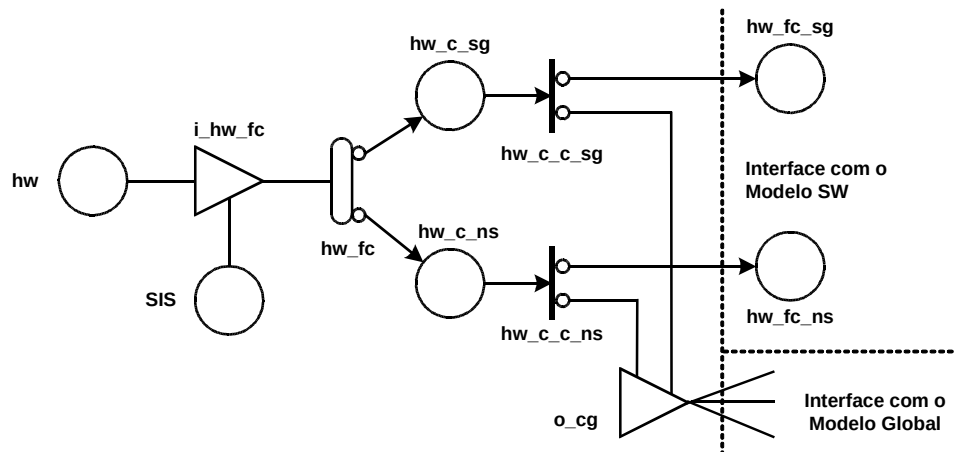


Figura 4.19 - Avarias em modo-comum.

As condições para a ocorrência das falhas em modo-comum são do mesmo tipo das descritas para os restantes tipos de falhas, sendo definidas pela porta de entrada i_hw_fc . Esta porta vai habilitar a transição exponencial hw_fc , que representa a ocorrência de uma falha em modo-comum. Desta forma, a respectiva taxa deverá reflectir este aspecto. Quando esta transição dispara, a falha ocorre e todos os testemunhos da posição hw são removidos, reflectindo assim o seus efeitos (a avaria simultânea de todos os componentes). Tal como nos restantes tipos de falhas, assumiu-se que estas são potencialmente seguras ou não.

A distinção entre ambas é realizada através da escolha das selecções associadas à transição hw_fc . A primeira escolha indica a ocorrência de uma falha potencialmente segura. Esta é seleccionada com probabilidade $p\_hw\_fc\_sg$ e a posição hw_c_sg é marcada. A segunda escolha, indica a ocorrência de uma falha não-segura, sendo seleccionada com probabilidade $(1-p\_hw\_fc\_ns)$, implicando a marcação da posição hw_c_ns . Deste ponto em diante, o comportamento do modelo é semelhante ao apresentado no modelo **HW**, considerando a existência de falhas de modo comum cobertas (execução da porta o_cg), ou não-cobertas (marcação das posições hw_fc_sg e hw_fc_ns) correspondendo à respectiva propagação ao modelo **SW**.

Por fim importa referir que as condições de avaria da rede são definidas de forma análoga, contando contudo agora com a falha simultânea de vários componentes a diferentes níveis.

4.3.7 Notas Finais Sobre os Modelos

Da análise dos modelos propostos, torna-se evidente que estes possuem um elevado número de parâmetros. Este facto é uma consequência de se ter assumido um carácter genérico para os mesmos. Numa primeira abordagem isto pode ser encarado como um problema, porque para obter a solução do modelo é necessário

conhecer todos estes valores, o que em termos práticos pode ser uma tarefa muito difícil. No entanto e numa outra perspectiva, este aspecto pode ser visto de forma positiva. Pela forma como os modelos foram desenvolvidos é simples proceder à sua simplificação. Os parâmetros que não são conhecidos podem ser removidos, ou o comportamento de vários mecanismos pode ser condensado num único parâmetro. Como a quase totalidade dos parâmetros estão associados a transições imediatas, torna-se óbvio a remoção ou a condensação podem ser facilmente implementadas, bastando para isso remover transições ou condensar várias transições numa única.

Numa perspectiva oposta, também é possível acrescentar novas funcionalidades aos modelos. Esta tarefa torna-se relativamente simples devido à decomposição estrutural realizada e à forma modular como os modelos foram desenvolvidos, o que permite identificar rapidamente os componentes abrangidos por estas funcionalidades e também localizar os seus efeitos ao nível da modelação.

A componente de *software* dos modelos está deliberadamente simplificada. Como foi discutido previamente, esta opção baseou-se no facto de que o comportamento respectivo é muito dependente da aplicação em causa, e que portanto não é possível de modelar antecipadamente. Note-se no entanto, que estes modelos possuem já os elementos estruturais necessários que lhes permitem ser facilmente adaptados às características das aplicações. Neste contexto, um dos aspectos que seria importante modelar diz respeito à estrutura funcional das tarefas que executam em diferentes nós, nomeadamente aos efeitos da propagação de erros entre estas.

Por fim importa referir que o comportamento em relação às falhas não-cobertas pode ser ligeiramente modificado em alguns casos, nomeadamente ao nível da sua propagação. Se for assumido que existem situações em que os mecanismos de tolerância a falhas dos níveis superiores (*software*) nunca vão funcionar (devido ao tipo de falha que ocorreu), é possível tomar uma decisão sobre a avaria da rede logo no modelo **HW** (ou num outro modelo de acordo com o caso). Ao nível do modelo, isto é implementado através da introdução de portas de saída adicionais que “impeçam” a propagação das falhas e definam as condições de avaria da rede nestas situações. Se além disso se assumir que existem falhas que se propagam ou não, é necessário que as transições imediatas associadas aos respectivos mecanismos de tolerância a falhas incluam 3 selecções, correspondendo respectivamente a falhas cobertas, não-cobertas com propagação e não-cobertas sem propagação.

4.4 Caso de Estudo

No sentido de validar a abordagem proposta é apresentado um caso de estudo.

De acordo com a discussão realizada no início deste capítulo, a dimensão do espaço de estados é o principal problema com que se tem de lidar quando se pretende obter uma solução analítica para modelos baseados em redes de Petri. Desta forma, em vez de apresentar um caso de estudo focado nas medidas de confiança no funcionamento, optou-se por caracterizar a qualidade dos modelos propostos face ao problema do espaço de estados. Esta medida de qualidade, é definida sob a forma de uma métrica correspondente à dimensão do espaço de estados gerado.

Esta métrica permite assim inferir o grau de exigência dos modelos ao nível dos recursos computacionais necessários para obter a respectiva solução analítica. Desta perspectiva, os parâmetros do modelo são apenas relevantes para a obtenção das medidas de confiança no funcionamento desejadas, dado que não influenciam a dimensão do espaço de estados gerado. Como a determinação das medidas não é o objectivo primário deste caso de estudo, utilizaram-se para os parâmetros do modelo valores totalmente genéricos.

O caso de estudo proposto, é baseado numa rede de campo composta por vários nós e vários *clusters*, em que todos os nós possuem características semelhantes. Ao nível da modelação considerou-se que o seu *hardware* pode ser redundante, sem contudo realizar qualquer decomposição específica da sua estrutura interna. Ou seja, o modelo HW representa todo o *hardware* do nó. Embora fosse possível considerar um maior grau de detalhe (ex. CPU, memória, sensores, actuadores, etc.), optou-se por não o modelar, dado que isto traduzir-se-ia numa maior complexidade ao nível da caracterização dos resultados obtidos. Refira-se contudo, que os efeitos desta decomposição podem ser grosso modo avaliadas fazendo variar o nível de redundância interna de cada nó. Um nó é considerado não-operacional se esgotar toda a sua redundância interna.

Ao nível dos *clusters* foi avaliado o efeito do seu número e respectiva dimensão face aos restantes nós da rede. Um *cluster* é considerado não-operacional se todos os nós que o constituem ficarem não-operacionais.

Em relação à infra-estrutura de comunicações, considerou-se que esta pode conter um meio físico de transmissão simples ou dual. Neste último caso, a infra-estrutura considera-se operacional enquanto existir um meio físico de transmissão operacional.

As condições de avaria segura da rede de campo são definidas pelo número máximo de nós (lógicos) não-operacionais (avariação segura) e/ou pela não-operacionalidade da estrutura de comunicações (avariação segura). Qualquer falha potencialmente não-segura e não-coberta destes elementos dá origem a uma avariação não-segura da rede.

Foram também incluídas as reparações locais e as falhas em modo comum ao nível dos nós. Note-se, que nas condições deste caso de estudo estes efeitos não contribuem para aumentar o espaço de estados. As reparações locais fazem evoluir a rede de campo para um estado previamente existente, e as falhas em modo comum são equivalentes à exaustão da redundância do nó. Ao nível das reparações foi considerado que existe apenas um recurso disponível, e não foi assumido qualquer tempo para a inicialização da rede após a reparação global.

4.4.1 Modelo de Avaliação Completo

O modelo de avaliação completo da rede de campo foi construído utilizando as primitivas *Join* e *Rep* do Möbius, cuja descrição foi previamente efectuada. A Figura 4.20 demonstra a aplicação destas primitivas para a construção do modelo.

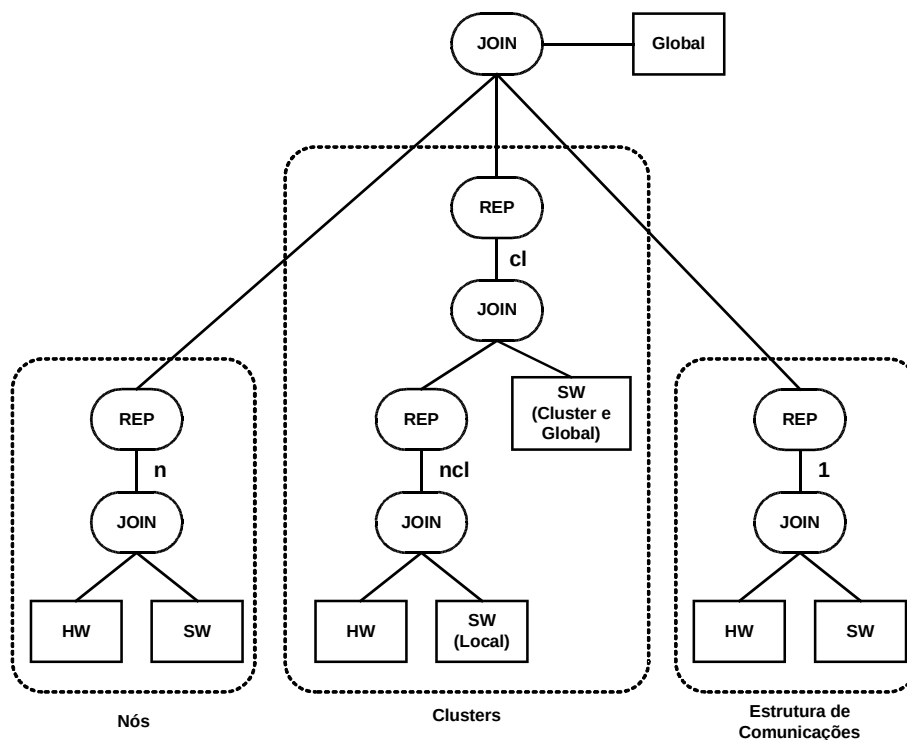


Figura 4.20 - Modelo de avaliação integrado da rede de campo.

O modelo do nó é obtido pela composição dos modelos **HW** e **SW** respectivos utilizando a primitiva *Join*. O modelo referente aos n nós da rede é obtido através da replicação n vezes deste último utilizando a primitiva *Rep*⁷³. Para obter o

⁷³ Neste caso é assumido que os nós possuem características semelhantes. Contudo, se fosse

modelo do *cluster* o procedimento é ligeiramente diferente. O modelo dos nós do *cluster* é obtido pela composição dos modelos **HW** e **SW**, mas neste último contém apenas a sua componente **Local**. Este modelo é replicado *ncl* vezes para representar os *ncl* nós do *cluster*. Posteriormente o modelo é composto com as componentes **Cluster** e **Global** do modelo **SW**, e por fim replicado *cl* vezes para representar o número de *clusters* da rede. O modelo da estrutura de comunicações (modelo *BUS*), é obtido de forma semelhante ao modelo do nó, contudo não é replicado dado que a sua estrutura interna já define a respectiva redundância. Embora a figura referente a este modelo indique uma primitiva *Rep* com uma replicação unitária, isto é apenas necessário porque a ferramenta não permite conectar directamente duas primitivas *Join*.

Por fim, o modelo completo da rede foi obtido através da composição dos modelos anteriores como o modelo **Global**. As ligações entre os modelos são realizadas por intermédio de posições globais. Posições com o mesmo nome em modelos diferentes representam uma posição única. Refira-se por fim que esta forma de construir o modelo é função das características do mesmo, ou seja, outros tipo de especificações podem levar (por vários motivos) a construir o modelo de forma diferente (ex. os modelos **HW** e **SW** do nó podem ser construídos como um único modelo).

4.4.2 Definição das Medidas de Interesse

Embora as medidas de confiança no funcionamento não sejam directamente focadas neste estudo, é fundamental indicar a forma de as obter, caso contrário o desenvolvimento dos modelos ficaria incompleto.

No caso de medidas associadas à disponibilidade, a definição de uma taxa de recompensa unitária (*reward rate*) na posição *SIS* e cálculo da recompensa esperada (instantânea) em regime estacionário, permite obter a disponibilidade da rede. Utilizando a mesma abordagem, mas agora nas posições *AS* e *ANS* é possível obter a disponibilidade relativa às avarias seguras e não-seguras respectivamente.

Para as medidas associadas à fiabilidade, é necessário proceder a algumas alterações nos modelos, dado que por definição não existem reparações. É assim necessário remover as partes dos modelos referentes às reparações⁷⁴. Note-se que o tipo de reparações a remover (locais ou globais) só depende das especificações do problema. A fiabilidade, pode ser obtida definindo uma taxa de recompensa

assumido que existem diferenças entre grupos de nós, em vez de uma replicação passar-se-ia a ter um conjunto de replicações, tantas quantos os grupos.

⁷⁴ Na prática não é necessário realizar qualquer modificação nos modelos, basta utilizar uma variável global (definida pelo Möbius) que permita inibir as partes do modelo que não são necessárias.

unitária na posição *SIS*, e calculando o complementar da recompensa (instantânea) em regime transitório. Definindo uma taxa de recompensa unitária para todas as marcações que não incluam as posições *SIS*, *AS* e *ANS* é possível calcular o MTTF da rede através da recompensa acumulada quando $t \rightarrow \infty$. Utilizando a metodologia proposta em [CWT98], é possível calcular o MTTF referente às falhas seguras e não-seguras. Este último procedimento é necessário, porque o Möbius não suporta modelos com mais dos que um estado absorvente.

As medidas associadas à segurança (*safety*) podem ser obtidas de forma análoga à disponibilidade e fiabilidade, mas agora definindo uma taxa de recompensa unitária associada à posição *AS*.

Para além das medidas relacionadas com a confiança no funcionamento, é também possível obter medidas de capacidade de desempenho (*performability*), tais como:

- ♦ Número médio de nós avariados: definir uma taxa de recompensa igual à marcação da posição *NOS*, e calcular o valor esperado da recompensa (instantânea) em regime estacionário;
- ♦ Tempo médio entre reparações: definir um impulso unitário associado à transição g_rp_d (modelo de reparação), e calcular o inverso do valor esperado da recompensa (instantânea) obtida em regime estacionário.

4.4.3 Resultados Obtidos

As tabelas 4.4, 4.5, 4.6, 4.7, 4.8, 4.9, 4.10 e 4.11 apresentam os resultados obtidos para a métrica definida (dimensão do espaço de estados) para diferentes configurações da rede. As configurações foram definidas de acordo com situações típicas de operação. Os resultados foram obtidos utilizando a versão alternativa do modelo global.

Para cada configuração é indicado o número mínimo de nós (lógicos) num estado de avaria segura que a rede pode tolerar, sem entrar a própria rede num estado de avaria segura. Além disso, cada configuração foi analisada para um meio de transmissão simples (não redundante) e dual. Assumiu-se que cada *cluster* é constituído por 2 nós.

Avaliação Integrada da Confiança no Funcionamento

Espaço de Estados	Meio simples - Condição de avaria segura: > 1 nó			
Redundância interna do nó (#hw) ⁷⁵	Número de nós da rede			
	3	5	7	10
1	11	11	11	11
2	44	74	104	149
3	108	273	514	1018

Tabela 4.4 - Meio simples - Condição de avaria segura: > 1 nó.

Espaço de Estados	Meio simples - Condição de avaria segura: > 2 nós			
Redundância interna do nó (#hw)	Número de nós da rede			
	3	5	7	10
1	17	17	17	17
2	56	102	148	217
3	126	349	688	1414

Tabela 4.5 - Meio simples - Condição de avaria segura: > 2 nós.

Espaço de Estados	Meio dual - Condição de avaria segura : > 1 nó			
Redundância interna do nó (#hw)	Número de nós da rede			
	3	5	7	10
1	30	30	30	30
2	119	203	287	413
3	292	752	1428	2847

Tabela 4.6 - Meio dual - Condição de avaria segura : > 1 nó.

Espaço de Estados	Meio dual - Condição de avaria segura: > 2 nós			
Redundância interna do nó (#hw)	Número de nós da rede			
	3	5	7	10
1	46	46	46	46
2	149	277	405	597
3	336	952	1896	3927

Tabela 4.7 - Meio dual - Condição de avaria segura: > 2 nós.

⁷⁵ #hw representa o número de testemunhos na posição hw do modelo HW.

Avaliação Integrada da Confiança no Funcionamento

Espaço de Estados	Meio simples - Condição de avaria segura: > 1 nó 2 Clusters c/ 2 nós		
Redundância interna do nó (#hw)	Número de nós da rede		
	2	4	6
1	96	96	96
2	1300	2488	3676
3	7005	21882	45171

Tabela 4.8 - Meio simples - Condição de avaria segura: > 1 nó – 2 Clusters c/ 2 nós.

Espaço de Estados	Meio simples - Condição de avaria segura: > 2 nós 2 Clusters c/ 2 nós		
Redundância interna do nó (#hw)	Número de nós da rede		
	2	4	6
1	203	209	209
2	2011	4463	6915
3	9225	34347	76565

Tabela 4.9 - Meio simples - Condição de avaria segura: > 2 nós - 2 Clusters c/ 2 nós.

Espaço de Estados	Meio dual - Condição de avaria segura: > 1 nó 2 Clusters c/ 2 nós		
Redundância interna do nó (#hw)	Número de nós da rede		
	2	4	6
1	254	254	254
2	3528	6800	10072
3	19227	60652	125705

Tabela 4.10 - Meio dual - Condição de avaria segura: > 1 nó - 2 Clusters c/ 2 nós.

Espaço de Estados	Meio dual - Condição de avaria segura: > 2 nós 2 Clusters c/ 2 nós		
Redundância interna do nó (#hw)	Número de nós da rede		
	2	4	6
1	536	554	554
2	5430	12198	18966
3	25196	95085	213066

Tabela 4.11 - Meio dual - Condição de avaria segura: > 2 nós – 2 Clusters c/ 2 nós.

A partir dos resultados obtidos, é possível verificar que a dimensão do espaço de estados gerado para cada configuração da rede toma valores perfeitamente compatíveis com os recursos computacionais (particularmente memória) existentes actualmente. Este aspecto estende-se mesmo aos casos limites de configuração da rede (com maior número de nós). É possível atribuir este comportamento a dois factores. Por um lado, a forma como os modelos foram desenvolvidos garantiu à partida que o espaço de estados gerado seria de pequena dimensão. Por outro lado, a técnica de redução do espaço de estados implementada pela ferramenta de modelação. Entre estes dois factores, não existe qualquer dúvida que o último é de facto o mais importante.

É assim possível concluir que o comportamento dos modelos, face à dimensão do espaço de estados gerado, é de facto muito bom, e que a sua resolução pode ser obtida sem quaisquer tipo de exigências especiais ao nível dos recursos computacionais necessários.

Por forma de concretizar este último aspecto, admita-se que se pretende avaliar uma rede constituída por n nós. Para isso dispõe-se de um modelo para o nó que possui m estados. Se admitirmos que o modelo global é obtido à custa da replicação (manual) do modelo dos nós, e que o comportamento destes modelos pode ser considerado como quase independente (com a excepção das avarias da rede), o espaço de estados M gerado tem aproximadamente seguinte dimensão

$$M \sim m^n \quad (4.3)$$

A Tabela 4.12 apresenta a dimensão do espaço de estados obtido de acordo com a expressão anterior para diferentes configurações da rede. Para que se possa realizar uma comparação com os resultados obtidos previamente, assumiu-se que a redundância interna do nó as características definidas anteriormente, e que os elementos que compõem esta estrutura podem estar num de dois estados possíveis: operacional e avaria, de forma independente dos restantes ($m=2^{\text{Redundância}}$).

Redundância interna do nó (#hw)	Número de nós da rede			
	3	5	7	10
1	27	32	128	1024
2	64	1024	16384	1 048 567
3	512	32768	2 097 152	1 073 741 824

Tabela 4.12 - Comparação de resultados.

A comparação destes resultados (Tab. 4.12) com os obtidos para o modelo proposto, permitem confirmar a existência de uma enorme redução do espaço de estados (para as configurações com maior número de nós a redução é muito

elevada), o que permite concluir, sem quaisquer dúvidas, da qualidade em relação a este aspecto dos modelos propostos.

Uma outra conclusão importante, diz respeito à possibilidade de obter modelos com um nível razoável de detalhe sem incorrer num espaço de estados de grande dimensão. Se a rede a avaliar tiver um dimensão pequena (digamos menor ou igual a 5 nós), os resultados obtidos permitem antever que é possível aumentar o nível de detalhe dos modelos (considerar um maior número de componentes) e ainda obter um espaço de estados perfeitamente razoável. É assim possível por exemplo incluir maior número de detalhes no modelo **SW**, nomeadamente ao nível do seu comportamento funcional, sem comprometer a capacidade de resolução do modelo completo.

4.5 Conclusões

Neste capítulo foram desenvolvidos vários modelos, que no seu conjunto permitem realizar uma avaliação integrada da confiança no funcionamento de sistemas de controlo suportadas por redes de campo.

A natureza distinta dos componentes da rede, com diferentes funcionalidades e níveis de interacção variados e complexos, aliada à necessidade de incluir várias perspectivas sobre o seu funcionamento, obrigou à utilização de um formalismo com elevado poder de modelação para o desenvolvimento dos modelos. Este foi concretizado através da utilização de redes de Petri Estocásticas. No entanto a sua utilização neste contexto, implicou também alguns problemas importantes ao nível da dimensão do espaço de estados do modelo. Desta forma, foi adoptada uma metodologia para o desenvolvimento dos modelos que minimizasse este aspecto. Esta assentou na utilização das vantagens oferecidas pela ferramenta Möbius neste contexto, e na utilização da técnica da decomposição comportamental.

Os modelos obtidos resultam de uma decomposição estrutural e funcional da rede, e incluem de forma integrada aspectos distintos. Estes vão desde o tipo de falhas consideradas até aos diferentes mecanismos de tolerância a falhas envolvidos, passando por modos de avaria da rede que incluem diferentes níveis de gravidade. Os modelos possuem assim um elevado nível de rigor e de detalhe, o que permite garantir a sua utilização com sucesso na avaliação de arquitecturas de controlo complexas.

Tendo em conta que o principal problema da abordagem proposta neste capítulo resulta da dimensão do espaço de estado do modelo, o caso de estudo proposto

focou-se unicamente sobre este aspecto. A partir deste estudo é possível concluir que os modelos desenvolvidos possuem em relação a este aspecto uma dimensão totalmente compatível com os recursos computacionais existentes actualmente. Além disso, os resultados também demonstram que é ainda possível introduzir maiores níveis de detalhe nos modelos sem comprometer a respectiva solução.

Face a este conjunto de aspectos, é possível concluir que a metodologia proposta para o desenvolvimento dos modelos juntamente com as características destes últimos, cumprem no seu conjunto os objectivos propostos inicialmente.

Esta página foi intencionalmente deixada em branco

Capítulo 5

Avaliação da Rede Profibus-DP em Cenários de Falhas Transitórias

5.1 Introdução

O Profibus é presentemente a rede de campo mais divulgada e com maior número de aplicações a nível mundial [Pro04]. A sua utilização estende-se por vários domínios de aplicação, incluindo, naturalmente, também aqueles em que a confiança no funcionamento é um dos requisitos necessários para a operação do sistema de controlo suportado pela rede.

Como foi discutido nos capítulos anteriores, as falhas transitórias que ocorrem durante as comunicações têm a capacidade de perturbar a dinâmica de controlo do processo. Este aspecto torna-se particularmente crítico em sistemas que possuem requisitos de tempo-real associados às comunicações. Nestes casos, as consequências das falhas traduzem-se pela violação das *deadlines* associadas às mensagens trocadas entre as várias tarefas que constituem a aplicação de controlo, que por sua vez podem dar origem a comportamentos indesejados ao nível do

processo controlado. Estes traduzem-se em avarias do mesmo, das quais podem resultar consequências de vários tipos e níveis de gravidade. Justifica-se assim neste contexto a necessidade de analisar o comportamento das redes de campo em cenários com estas características.

No capítulo 3, este assunto foi já alvo de uma discussão e de um conjunto de propostas ao nível dos modelos de avaliação. No entanto, o contexto em que estes modelos foram desenvolvidos permite apenas a sua utilização em avaliações preliminares ou comparativas, onde não é necessário dispor de um nível de rigor elevado. Quando se pretende uma avaliação rigorosa e absoluta estes modelos deixam de ser aplicáveis, sendo necessário desenvolver modelos específicos para a rede de campo que se pretende avaliar.

Neste sentido, neste capítulo é desenvolvido um modelo para avaliar a confiança no funcionamento de uma rede Profibus-DP em cenários de falhas transitórias que ocorrem durante as comunicações. Para além deste objectivo principal, este modelo enquadra-se também na abordagem definida no capítulo anterior, nomeadamente através da utilização da decomposição comportamental para integrar os resultados obtidos.

Este capítulo representa uma versão extendida dos trabalhos publicados pelo autor em [PA04b] e [PA04c].

5.2 O Profibus

O Profibus⁷⁶ (*PROcess FIeld BUS*) [IEC03a] [IEC03b] foi desenvolvido nos finais dos anos 80 por um conjunto de empresas Alemãs, lideradas pela Siemens, para ser utilizado na interligação de dispositivos de baixa complexidade localizados nos níveis de campo e de célula. Inicialmente foram desenvolvidas três soluções: Profibus-FMS, Profibus-DP e Profibus-PA, existindo presentemente, em termos comerciais, apenas as duas últimas. O trabalho neste capítulo foca-se unicamente sobre o Profibus-DP-V0 (*Decentralized Peripherals*).

A *stack* de comunicações do Profibus está organizada em camadas segundo o modelo OSI [ISO94], mas utilizando apenas três camadas definidas de acordo com este modelo: física, ligação de dados e aplicação. Esta última camada acaba por

⁷⁶ Para evitar ambiguidades utilizar-se-á o termo *Profibus* para nos referirmos às características da rede em geral, e o termo *Profibus-DP* para nos referirmos particularmente à implementação da componente DP.

agregar algumas das funções previstas nas camadas intermédias do modelo que não são utilizadas.

As duas primeiras camadas, são totalmente iguais nas soluções Profibus-FMS e Profibus-DP [EN96a], possuindo no entanto algumas diferenças (não demasiado significativas) quando comparadas com as utilizadas no Profibus-PA [EN99]. A camada de aplicação, implementa os serviços específicos do Profibus-DP [EN96b].

A arquitectura de comunicações é modular, disponibilizando uma gama alargada de perfis, ao nível da aplicação e do sistema, e oferecendo várias ferramentas de integração e gestão de dispositivos (Fig. 5.1) [Pro04].

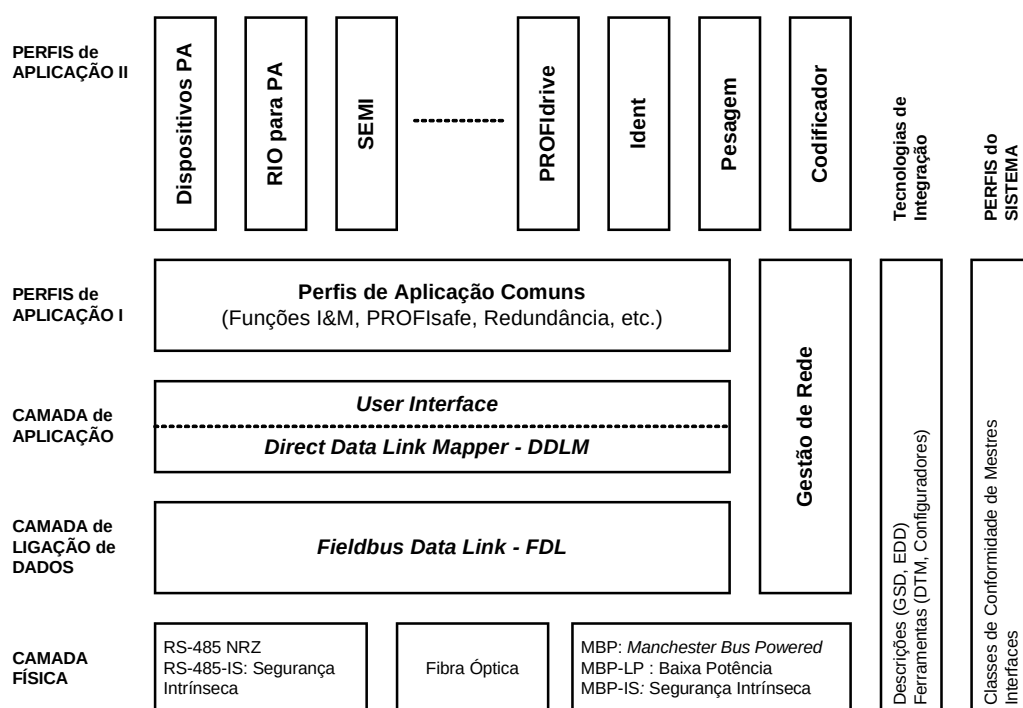


Figura 5.1 - Arquitetura de comunicações do Profibus-DP.

5.2.1 Camada Física

A camada física define uma topologia em *bus*, suportando vários meios físicos de transmissão com diferentes características:

- ♦ **RS-485.** Os dados são transmitidos sob a forma de um sinal diferencial através de uma codificação NRZ (*Non Return to Zero*), utilizando um par torcido blindado. Garante-se assim uma boa imunidade ao ruído electromagnético e uma elevada velocidade de transmissão. Até 32 dispositivos podem estar ligados ao mesmo segmento de rede sem utilizar repetidores e o número máximo de dispositivos na rede é de 126. A velocidade de transmissão pode variar desde 9.6Kbits/s até 12Mbits/s. O

comprimento máximo da rede é função da velocidade de transmissão, podendo variar entre 1200m@9.6Kbits/s até 100m@12Mbits/s por segmento;

- ♦ **RS-485-IS.** Semelhante à solução anterior em termos de funcionamento, mas inclui algumas características especiais que permitem a sua utilização em ambientes intrinsecamente seguros (*intrinsically safe*). Nomeadamente, são definidos os níveis máximos de tensão e de corrente permitidos no meio de transmissão. É utilizado um duplo par torcido blindado e a velocidade de transmissão pode variar desde 9.6Kbits/s até 1.5Mbits/s;
- ♦ **MBP** (*Manchester Coding and Bus Power*). Os dados são codificados utilizando a codificação *Manchester* e os sinais são transmitidos de forma síncrona num par torcido blindado. No mesmo meio físico e juntamente com o sinal transmitido, é também fornecida alimentação para os dispositivos ligados à rede. Tal como no caso anterior, esta tecnologia foi desenvolvida para ser utilizada em ambientes intrinsecamente seguros. A velocidade de transmissão é fixa em 32Kbits/s. O número de dispositivos permitido é mesmo dos casos anteriores;
- ♦ **Fibra Óptica.** Utilizada em aplicações onde os níveis de interferência electromagnética são muito elevados, ou quando se pretende cobrir longas distâncias.

5.2.2 Camada de Ligação de Dados

Numa rede Profibus, existem dois tipos de estações⁷⁷ (dispositivos): mestres e escravos, sendo os primeiros ainda classificados no Profibus-DP como de classe 1 (DPM1) ou classe 2 (DPM2).

Os mestres de classe 1 (DPM1), são utilizados para implementar as tarefas associadas ao controlo do processo (ex. autómatos programáveis). A sua operação consiste na troca de dados (*polling*) com os escravos e na execução das aplicações de controlo. Os mestres de classe 2 (DPM2), são utilizados em tarefas de configuração das estações e na gestão do funcionamento da rede. Os escravos são dispositivos periféricos (ex. variadores de velocidade, I/O analógicos e digitais), que realizam a interface directa com o processo controlado e que respondem a pedidos por parte dos mestres⁷⁸.

A camada de ligação de dados (FDL – *Fieldbus Data Link*) implementa um mecanismo de acesso ao meio híbrido. É utilizado um mecanismo descentralizado baseado na passagem de um *token*⁷⁹ (testemunho) entre estações mestres para definir o acesso ao meio, ao qual está subordinado a um mecanismo baseado num

⁷⁷ Adoptou-se a terminologia definida em [EN96a].

⁷⁸ Assume-se deste ponto em diante que os mestres são todos de classe 1.

⁷⁹ Trama com uma interpretação particular.

modelo Mestre-Escravo (*Master-Slave*) para as comunicações entre as estações mestres e escravos (Fig. 5.2). As comunicações são sempre iniciadas pelos mestres (estações activas), enquanto que os escravos (estações passivas) respondem apenas a pedidos dos mestres. A cada mestre estão associados vários escravos, mas um escravo só pode estar associado a um único mestre.

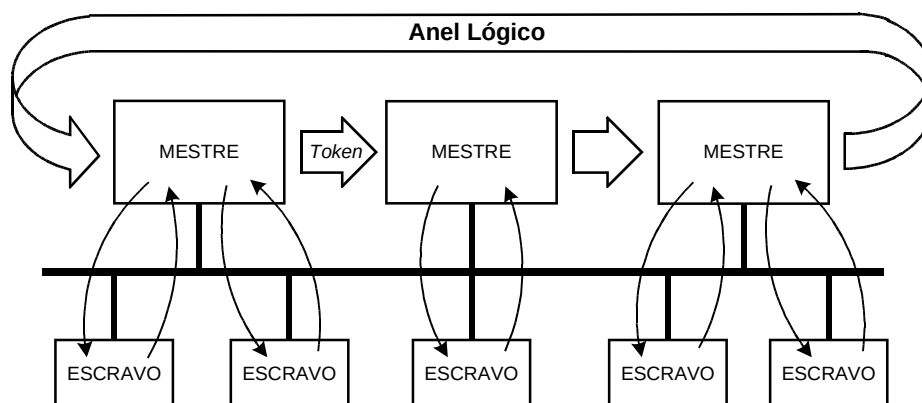


Figura 5.2 - Mecanismo de acesso ao meio.

As estações mestres estão organizadas segundo um anel lógico, com o objectivo de implementar o mecanismo de passagem do *token*. Assim, após um mestre receber o *token* do seu antecessor (referido como PS – *Previous Station*), dispõe de um tempo (limitado) que pode utilizar para a troca de dados com os escravos associados, após o qual deverá passar o *token* ao seu sucessor (referido como NS – *Next Station*). Este processo é repetido pelos vários mestres sucessivamente ao longo do anel.

Para além de gerir o acesso ao meio de transmissão, os mestres também são responsáveis pela gestão do anel lógico. Para isso cada mestre mantém uma lista que representa todas as estações activas no anel (LAS – *List of Active Stations*). De acordo com esta lista, o *token* é passado de forma ascendente, utilizando para isso o endereço lógico dos mestres, até se atingir o mestre com o endereço mais elevado. Após este caso, o *token* é passado ao mestre com o endereço mais baixo.

Embora [EN96a] defina 4 serviços de comunicação: SDN (*Send Data with No Acknowledge*), SDA (*Send Data with Acknowledge*), SRD (*Send and Request Data with Reply*) e CSRD (*Cyclic Send and Request Data with Reply*), o Profibus-DP [EN96b] só utiliza dois destes serviços: SDN e SRD.

O serviço SDN é utilizado pelos mestres para enviar dados sem confirmação para todas as estações da rede (*broadcast*) ou para um grupo particular destas (*multicast*). O serviço SRD é utilizado pelos mestres para enviar dados com confirmação para os escravos. Na resposta (confirmação), os escravos podem também enviar dados. Para cada serviço estão definidos 2 níveis de prioridade: alta e baixa. No entanto, o

Profibus-DP só utiliza o nível de alta prioridade (existem algumas excepções mas que não são importantes no âmbito deste trabalho).

Durante a descrição do modelo de avaliação desenvolvido, serão apresentados com maior detalhe as características dos vários serviços, bem como os mecanismos empregues ao nível da gestão de rede.

5.2.3 Camada de Aplicação

A camada de aplicação do Profibus-DP foi desenvolvida com o objectivo de permitir uma troca de dados rápida e eficiente com os escravos. Esta troca de dados pode assumir várias formas, mas está essencialmente organizada segundo um modelo de transferência cíclico.

Presentemente existem 3 versões desta camada [IEC03a] [IEC03b] [JSW04], resultantes das crescentes necessidades das aplicações suportadas pelo Profibus-DP:

- ♦ **DP-V0.** Disponibiliza as funcionalidades básicas do DP, incluindo troca cíclica de dados e diagnósticos dos escravos, módulos e canais lógicos;
- ♦ **DP-V1.** Em relação à anterior, acrescenta a possibilidade de existirem trocas acíclicas de dados para definição de parâmetros, operação, visualização e gestão de alarmes. Além disso, permite a definição de alarmes com características específicas;
- ♦ **DP-V2.** Em relação à anterior, acrescenta a possibilidade de existirem comunicações isocrónicas (*isochronous*) com os escravos e também a troca de dados de forma directa entre escravos sem utilizar os mestres.

No caso do Profibus-DP-V0, a camada de aplicação é composta por duas sub-camadas. A primeira é denominada *User Interface* e disponibiliza o conjunto de serviços básicos do Profibus-DP, nomeadamente: troca cíclica de dados, informação de diagnóstico dos escravos, parametrização e configuração dos escravos e funções de controlo. A segunda, denominada DDLM (*Direct Data Link Mapper*), tem como função mapear os serviços disponibilizados na camada anterior, nos serviços da camada de ligação de dados.

Durante a descrição do modelo de avaliação desenvolvido, serão apresentados com maior detalhe as características dos vários serviços disponibilizados por esta camada.

5.3 Discussão dos Trabalhos Relevantes Previamente Desenvolvidos

No contexto do problema proposto tratar neste capítulo, os trabalhos previamente desenvolvidos são relativamente reduzidos. No entanto, é possível alargar um pouco este contexto e incluir ao nível da discussão também trabalhos em domínios próximos. Neste sentido, os trabalhos foram agrupados segundo duas linhas de investigação. A primeira, está relacionada com a análise do comportamento de redes do tipo *Token Bus* em cenários de falhas durante as comunicações. A segunda, centra-se os sistemas em tempo-real, e em particular no escalonamento de mensagens⁸⁰.

Nas secções seguintes proceder-se-à uma descrição cronológica dos trabalhos mais relevantes de acordo com a estrutura previamente definida.

5.3.1 Comportamento em Cenários de Falhas Durante as Comunicações

Os trabalhos desenvolvidos neste âmbito e ao nível das redes de campo são de facto muito reduzidos. Para além destes, existem também alguns que se focam em redes com um funcionamento semelhante, nomeadamente para o IEEE 802.4, (*Token Bus*), pelo que se justifica assim também a sua referência

Em [SMM89] é realizado um estudo sobre o desempenho do IEEE 802.4 na presença de falhas transitórias durante as comunicações, através da utilização de modelo de simulação baseado em redes de Petri Estocásticas. O estudo permite avaliar até que ponto a utilização do formalismos baseados em redes de Petri Estocásticas são adequados para modelar e analisar (incluindo a possibilidade de utilizar simulação) o comportamento de uma rede com características semelhantes à do Profibus.

[MPA+98] realiza um estudo sobre o comportamento do IEEE 802.4 em ambientes sujeitos a ruído electromagnético. O trabalho foca-se essencialmente sobre o comportamento dos mecanismos de passagem do *token* entre estações. Para estudar esse comportamento é proposto um modelo analítico baseado numa cadeia de Markov Contínua. O tráfego de mensagens é considerado constante, e as falhas durante as comunicações são modeladas através da ocorrência de *bursts*⁸¹. As medidas realizadas focam-se na determinação dos tempos médios (valores

⁸⁰ Utilizou-se a terminologia mais comum segundo a qual uma mensagem corresponde a uma trama contendo dados do processo trocada entre um mestre e um escravo.

⁸¹ Optou-se por utilizar a terminologia em Inglês por se julgar mais adequada ao contexto.

esperados) de passagem e de rotação do *token*. A natureza o modelo e o tipo de solução pretendida implicaram a adopção de um conjunto aproximações, no sentido de tornar o modelo resolúvel. Para além do estudo em si realizado, este trabalho permite avaliar que tipo de pressupostos e aproximações são necessários para o desenvolvimento de um modelo analítico para uma rede com estas características.

Em [BM99] é realizada uma análise da tolerância a falhas do Profibus. São identificadas os vários tipo de falhas existentes, as respectivas causas e que tipo de mecanismos intervêm durante a sua ocorrência. O comportamento rede é descrito por intermédio de um modelo baseado em redes de Perti Generalizadas e Estocásticas, cuja avaliação é realizada através de simulação. As medidas realizadas centram-se em torno do número de estações operacionais e do número esperado de falhas. Este trabalho sofre de várias deficiências, a começar pelas características do modelo desenvolvido. Os pressupostos assumidos são em alguns casos demasiado simplistas, e noutros mal formulados. Acresce ainda o facto de que vários dos parâmetros do modelo serem completamente desconhecidos.

Em [Wil99a], [Wil99b] e [Wil02] é realizado um estudo sobre a estabilidade do funcionamento do Profibus sobre meios físicos sujeitos a falhas frequentes durante as comunicações, nomeadamente do tipo sem-fios (*wireless*). Este trabalho tem várias componentes. Numa primeira fase [Wil99a], são analisadas e quantificadas quais as causas que mais contribuem para a instabilidade do anel e são propostos dois modelos alternativos para a ocorrência das falhas. Posteriormente, é desenvolvido um modelo de simulação (baseado num ambiente proprietário), a partir do qual são derivadas várias medidas de desempenho relacionadas com a estabilidade do anel. Os resultados obtidos partem, na maioria do casos, do pressuposto que não existe tráfego de mensagens da rede, mas apenas passagens do *token* e mensagens GAP. São também propostas algumas alterações ao funcionamento do Profibus no sentido de melhorar o seu desempenho. Numa segunda fase [Wil99b], é proposto um modelo analítico para o mesmo problema, baseado numa cadeia de Markov Contínua. Dadas as características do modelo, este é obviamente muito simplificado e baseado em pressupostos que facilitem a sua resolução. No entanto os resultados obtidos indicam que os erros cometidos em relação ao modelo de simulação são pequenos. Na sua globalidade este trabalho é de muita qualidade, apresentado contudo duas limitações. A mais importante, está relacionada com o tipo de análise realizada. Esta, praticamente, não incorpora a existência de tráfego de mensagens, e quando este é considerado não está de acordo com as características do Profibus-DP (particularmente com a componente DP). Isto resulta directamente dos objectivos propostos no trabalho, nomeadamente da análise da estabilidade do anel durante a passagem do *token*. A segunda limitação, está relacionada com algumas imprecisões importantes cometidas ao nível do comportamento assumido para o Profibus. Durante a

descrição do modelo desenvolvido neste capítulo, far-se-á uma apresentação mais detalhada das mesmas.

5.3.2 Escalonamento de Mensagens em Sistemas de Tempo-Real

Os trabalhos desenvolvidos no âmbito do escalonamento de mensagens na rede Profibus, não fazem menção directa da existência de falhas de comunicação. No entanto, acabam por considerar os seus efeitos (embora em muito pequena parte) de forma indirecta através da inclusão da possibilidade de existirem retransmissões (devido a erros) durante a transmissão das mensagens. A sua referência justifica-se segundo duas perspectivas. Por um lado, permitem identificar quais os parâmetros mais importantes ao nível do funcionamento da rede neste contexto, Por outro lado, possibilitam uma avaliação muito preliminar do comportamento do Profibus num contexto de falhas transitórias durante as comunicações.

Em [VJ94] é realizada uma análise simples do escalonamento de mensagens numa rede Profibus. Partindo da análise do escalonamento de tarefas com prioridade fixa num processador, desenvolve-se um conjunto de expressões analíticas que permitem calcular o tempo de resposta da mensagem num cenário pessimista (WCRT - *Worst Case Response Time*), e além disso verificar se as *deadlines* das mensagens são violadas, ou não. Além disso, o trabalho também estabelece, no cenário proposto, qual o valor mínimo do parâmetro da rede T_{TR} (*Target Rotation Time*) que garante as condições necessárias para que o escalonamento seja possível.

[TV98a] e [TV98b] a análise anterior é extendida e melhorada, no sentido em que são definidos modelos mais exactos do tipo de mensagens transmitidas. Este aspecto reflecte-se na inclusão de mensagens com diferentes tipos de prioridades e em mensagens do tipo GAP e *Poll List*. São também contabilizadas nesta análise eventuais retransmissões em cenários de erros durante as comunicações.

Em [TV99a], e partindo da análise anterior, são formalizadas um conjunto de propriedades referentes ao suporte pelo Profibus de comunicações em tempo-real, bem como propostas duas análises que consideram a existência de diferentes tipos de restrições ao nível do número de mensagens de baixa prioridade transmitidas.

[TV99b] é consolidado o trabalho anterior, através da consideração de todo o tipo de tráfego de mensagens. O trabalho foca-se na determinação do tempo de ciclo, a partir do qual o valor para T_{TR} pode ser definido com maior rigor.

Em [MAV+00], [CMT+02a] e [CMT+02b] é analisado o escalonamento de mensagens numa rede Profibus-DP, sendo proposta uma abordagem menos pessimista do que a realizada em [TV99a]. Numa primeira fase é avaliado o

comportamento de uma rede mono-mestre. Esta análise é posteriormente extendida a uma rede multi-mestre. São considerados modelos mais realistas para o tráfego de mensagens, nomeadamente através da inclusão da possibilidade de transmissão de mais do que uma mensagem de alta prioridade por visita do *token*, na relaxação das condições impostas ao T_{RR} (*Real Rotation Time*) e na possibilidade do tráfego de baixa prioridade ser periódico. Como nota final importa referir que estes trabalhos baseiam-se em alguns pressupostos que se verificam não existir no Profibus-DP-V0. Nomeadamente é referida a existência de mensagens de baixa prioridade transmitidas pelo mestre, quando estas simplesmente não existem. De facto [EN96b] refere que numa comunicação mestre-escravo, todos os pedidos de serviço da camada de aplicação são mapeados em serviços de alta prioridade (DLL), com as respostas mapeadas em serviços de baixa prioridade (com algumas excepções). No entanto, este último caso só influencia as mensagens recebidas.

5.4 Metodologia Proposta

Tendo em conta os trabalhos desenvolvidos previamente, coloca-se a questão de até que ponto é necessário propor uma nova abordagem, e em caso afirmativo quais os objectivos que se pretendem atingir.

São muitos, e com características diversas, os tipos de sistemas suportados pela rede Profibus-DP. Afigura-se assim difícil (para não dizer impossível) desenvolver um modelo para a avaliação da confiança no funcionamento neste contexto, capaz de ser empregue em todos eles. No entanto, é um facto conhecido que a esmagadora maioria destes sistemas possui requisitos de tempo-real associados à transmissão das mensagens de controlo do processo [Tho98] [TV99a]. Neste sentido, tem toda a lógica focar o desenvolvimento do modelo de avaliação nestes tipos de sistemas. Assim e como resultado desta opção, fica facilitado à partida o tipo de medidas que se pretende obter, tendo em conta que se conhece com detalhe o tipo de serviço fornecido.

Partindo da análise dos trabalhos desenvolvidos, é possível concluir que [Wil99a], [Wil99b] e [Wil02] representam os trabalhos mais relevantes e mais próximos dos objectivos que se pretendem atingir neste capítulo. Focando assim a discussão nestes trabalhos, é possível concluir que embora estes sejam de grande qualidade, não fornecem uma resposta completa em relação ao problema que se pretende analisar. Isto deve-se essencialmente aos objectivos e aos pressupostos assumidos nos mesmos, nomeadamente quando estes assumem nos modelos de avaliação propostos que praticamente não existe tráfego de mensagens. Representando estas mensagens um papel central ao nível do controlo do processo [SK92] [KS94]

[Kop97] [TV99b], torna-se absolutamente necessário caracterizar qual o comportamento do sistema quando ocorrem falhas transitórias durante a respectiva transmissão.

Está assim justificada a necessidade de se desenvolver uma nova abordagem para avaliar a confiança no funcionamento da rede Profibus-DP neste tipo de cenários. O próximo passo, é discutir em que condições esta deve ser desenvolvida e aplicada.

Das discussões realizadas nos capítulos 2 e 4, ficou patente que as redes de Petri Estocásticas representam um formalismo de modelação que oferece um conjunto alargado de vantagens no desenvolvimento de modelos de avaliação da confiança no funcionamento. Desta forma, optou-se neste capítulo por utilizar também este formalismo no desenvolvimento dos modelos propostos. Neste sentido, as discussões mantidas nas secções seguintes devem ser enquadradas neste contexto.

5.4.1 Pressupostos Assumidos

O desenvolvimento do modelo de avaliação foi realizado com base nos seguintes pressupostos:

- ♦ Dada a complexidade do Profibus-DP, não é possível desenvolver um modelo com solução analítica capaz de descrever com o rigor pretendido neste trabalho o respectivo comportamento [MPA+98].

A principal causa desta limitação resulta do facto do Profibus-DP incluir vários temporizadores, ao nível da sua *stack* de comunicações, que estão activos simultaneamente. Se os temporizadores forem representados no modelo por transições determinísticas (como é frequente), isto implica a existência de múltiplas transições determinísticas habilitadas simultaneamente. Como foi discutido no capítulo 2, a existência de uma solução analítica para este tipo de situações só possível em casos muito particulares. Além disso e mesmo nestes casos, as ferramentas que existem presentemente estão limitadas a soluções analíticas que consideram apenas 1 ou 2 transições deste tipo habilitadas simultaneamente.

Face a este conjunto de limitações, optou-se por desenvolver um modelo cuja solução é obtida através de simulação. Dados os problemas típicos associados à obtenção de uma solução com estas características (discutidos no capítulo 2), é necessário desenvolver o modelo por forma a garantir que a solução do mesmo é exequível em termos de recursos computacionais exigidos. Isto aplica-se quer ao tempo necessário para obter os resultados, quer ao tipo e quantidade de recursos (CPU, memória, etc.) necessários;

- ♦ O comportamento da rede foi considerado o mais próximo possível do funcionamento típico das aplicações reais. Nomeadamente, foi incluído o comportamento específico da componente da arquitectura de comunicações referente ao DP. Isto inclui, para além da troca de mensagens de dados, as fases referentes à configuração, parametrização e ao diagnóstico dos escravos;
- ♦ Não se assume qualquer tipo de redundância ao nível do meio físico de transmissão utilizado⁸²;
- ♦ As falhas ocorrem de acordo com um processo de Poisson homogéneo no tempo [Triv02] e afectam apenas as comunicações. Este modelo corresponde ao modelo clássico de “erros independentes” adoptado em [Wil99a]. Ou seja, existe uma probabilidade fixa de uma trama ser transmitida com sucesso que só depende do comprimento da trama e é independente do instante em que esta é transmitida. Assume-se além disso que os efeitos das falhas são vistos de forma igual por todos os nós da rede;
- ♦ Se durante uma transmissão de uma trama ocorrer uma falha (interferência), assume-se que esta provoca a destruição do conteúdo da mesma, através da introdução de erros. É assumido que os erros provocados são sempre detectados pelas estações, com a excepção dos que ocorrem na trama do *token* (os motivos para tal serão apresentados adiante). Com isto pretende-se propositadamente excluir os efeitos de erros não detectados nas tramas. Esta opção partiu do facto de que a avaliação deste efeito teria necessariamente de incluir aspectos referentes à própria aplicação (ex. mecanismos de tolerância a falhas). Como se pretende separar este efeito, ele não é considerado;
- ♦ O sistema que se pretende avaliar possui requisitos de tempo-real associados às comunicações, concretizados através da definição de *deadlines* associadas às mensagens transmitidas;
- ♦ A avaria da rede neste contexto, ocorre quando as *deadlines* associadas às mensagens são violadas. Não se assumem diferentes níveis de gravidade para as consequências das avarias (ex. segura, não-segura). Isto deve-se ao facto deste aspecto ser específico das características da aplicação de controlo, sendo assim deixada em aberto;
- ♦ Na ausência de falhas assume-se que não existem quaisquer violações das *deadlines* associadas às mensagens;
- ♦ A principal medida de interesse é determinar qual a probabilidade da *deadline* associada à respectiva mensagem (ou mensagens) ser violada.

⁸² Este pressuposto foi apenas assumido para simplificar o modelo. Como se poderá ver da discussão nas secções seguintes se necessário pode ser facilmente introduzido.

5.4.2 Ferramenta da Modelação

Como se optou pela utilização da simulação para obter a solução do modelo, é necessário definir previamente quais os requisitos que a ferramenta de modelação utilizada deverá possuir. Estes podem ser sistematizados da seguinte forma:

- ◆ **Complexidade**

Da análise do Profibus-DP [EN96a] [EN96b], fica suficientemente claro que se está na presença de uma arquitectura de comunicações cuja implementação é complexa. Desta forma, é necessário garantir que a ferramenta escolhida tenha uma capacidade de modelação adequada a este problema, caso contrário a construção do modelo será difícil de realizar. Isto pode implicar limitações ao nível da sua análise e validação;

- ◆ **Eficiência do simulador**

Fica patente do ponto anterior, que o modelo proposto apresenta um grau de complexidade elevado. Resulta daqui uma maior exigência computacional durante a simulação, que, eventualmente, pode causar um pior desempenho da mesma (aumento do tempo de simulação). É assim necessário dispor de um simulador eficiente;

- ◆ **Validação**

É necessário que disponha de funcionalidades que permitam validar o modelo obtido. Esta validação vai permitir verificar o grau de conformidade do modelo com a especificação do Profibus-DP [EN96a] [EN96b] e assim eliminar eventuais erros de modelação;

- ◆ **Medidas**

Suporte do tipo de medidas que se pretendem obter, nomeadamente através da utilização do conceito de recompensas (*rewards*).

Após uma avaliação prévia das ferramentas existentes [JM88] [TM93] [HN96], optou-se por utilizar a ferramenta **Möbius** [Wil96] [CCD+01]. Dado que já foram discutidas previamente as capacidades desta ferramenta referentes à modelação utilizando redes de Petri, far-se-á apenas uma apresentação das características do seu simulador.

O simulador do Möbius apresenta um conjunto de características avançadas das quais se destacam:

- ◆ Execução de simulações num ambiente distribuído e heterogéneo. Utilizando uma arquitectura mestre-escravo é possível ter várias simulações a executar simultaneamente em diferentes máquinas e com diferentes sistemas

operativos. O processamento estatístico dos resultados é centralizado numa única máquina;

- ♦ Realização automática do número de simulações necessárias para atingir o nível de precisão definido através das características do intervalo de confiança pretendido;
- ♦ Gestão eficiente das várias listas de eventos (*event list*) associadas à simulação de eventos discretos. Este aspecto é crítico, pois influencia de forma directa o desempenho do simulador;
- ♦ Transposição do modelo para C++. O modelo a simular é convertido para código em C++, compilado utilizando ferramentas standard (ex. g++) e posteriormente executado como um processo normal do sistema operativo. Permite-se assim tirar partido directo da utilização de uma linguagem de programação de alto nível, das funcionalidades ao nível da optimização das ferramentas de compilação e das características do sistema operativo.

5.5 Modelos Desenvolvidos

O desenvolvimento do modelo foi realizado de acordo com uma perspectiva modular, sendo composto por três modelos de menor dimensão: mestre, escravo e global. Estes modelos podem ser combinados de várias formas, com o objectivo de analisar diferentes configurações da rede.

Durante a construção do modelo procurou-se minimizar todos os aspectos que possam prejudicar o desempenho da simulação, tais como:

- ♦ Foram utilizadas o menor número possível de transições habilitadas simultaneamente (imediatas ou temporizadas), por forma a manter as várias listas de eventos com dimensões o mais reduzidas possíveis;
- ♦ Não foi permitido que exista mais do que uma transição imediata habilitada em cada instante. Pretende-se assim obter de um comportamento bem definido;
- ♦ O desenvolvimento do código em C++ para as portas e as transições (SANs) foi realizado de forma a garantir uma execução eficiente.

Nas secções seguintes é realizada uma descrição pormenorizada de cada um dos modelos, bem como a uma discussão mais alargada sobre pressupostos em que estes foram desenvolvidos.

Dado o carácter muito extenso do código desenvolvido, optou-se por não o incluir directamente neste capítulo mas sim no Anexo 3. Neste sentido, qualquer dúvida sobre aspectos particulares do modelo pode ser esclarecida através da consulta do mesmo. A única parte não referida, nem incluída, por não se julgar relevante, diz respeito à inicialização dos modelos.

5.5.1 Modelo do Mestre

Este modelo descreve o comportamento de uma estação mestre. Os mestres são os responsáveis pela passagem do *token*, pela gestão do anel e pela troca de dados com os escravos que lhe estão associados.

O funcionamento do mestre pode ser descrito por intermédio de um diagrama de estados associado à camada de ligação de dados (FDL) [EN96a] (Fig. 5.3). Adaptando esse mesmo diagrama, utilizaram-se as seguintes posições da rede de Petri para indicar os estados correspondentes: *Offline*, *Listen* (*Listen Token*), *Active* (*Active Idle*), *Pass* (*Pass Token*), *Ck_Pass* (*Check Token Pass*) e *Use* (*Use Token*). Pormenores sobre o significado de cada um dos estados e das respectivas transições podem ser encontrados em [EN96a]. Refira-se contudo, que durante as descrições dos modelos o significado de cada um dos estados vai ser alvo de uma apresentação e discussão, que embora menos formal vai permitir perceber o seu contexto.

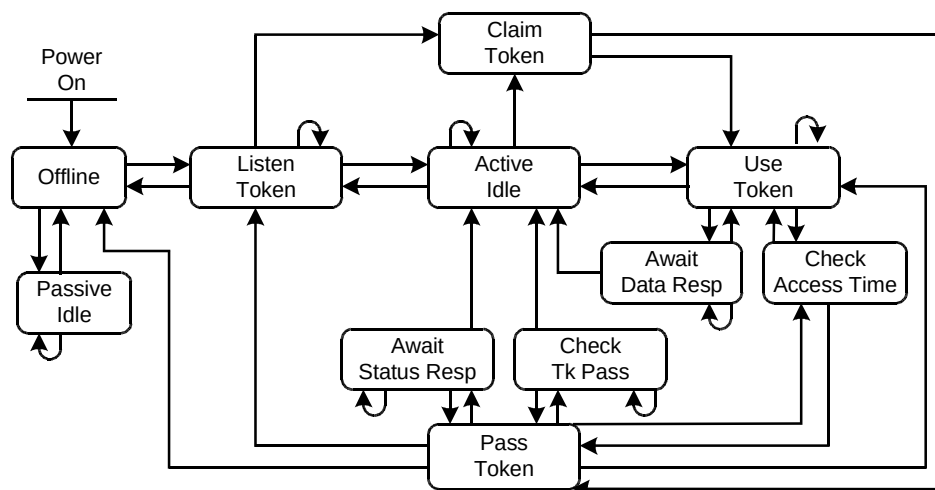


Figura 5.3 - Diagrama de estados do Profibus.

O modelo do mestre resulta da composição de vários modelos específicos: passagem do *token*, transmissão do *token*, rotação do *token*, gestão do anel, re-inicialização do anel, re-inicialização do mestre, transmissão de mensagens e gestão dos escravos. Nas secções seguintes cada um destes modelos será discutido em particular.

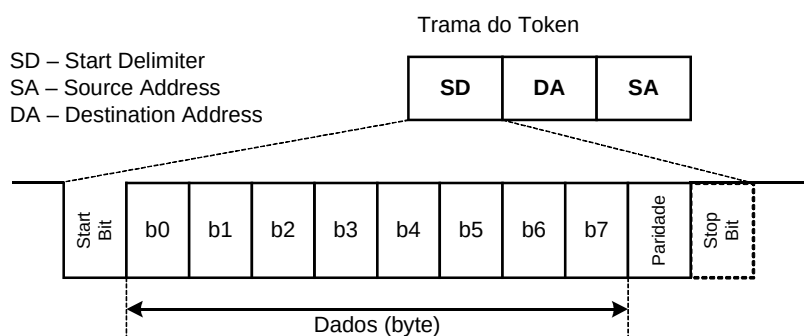


Figura 5.5 - Trama do token.

Quando o mestre se encontra na posição (estado) *pass*, significa que possui o *token* e que pretende passá-lo ao próximo mestre no anel. A passagem do *token* é representada pela transição determinística t_{TT} , cuja duração corresponde ao tempo necessário para transmitir a respectiva trama. Durante a transmissão o mestre monitoriza o meio de transmissão. No final da transmissão do *token* um de quatro cenários, representados pelas 4 selecções da transição t_{TT} , pode ter ocorrido:

1. **A transmissão decorreu com um erro grave que foi detectado.** A primeira selecção é escolhida (o_{eg}) com uma probabilidade p_{eg} . A existência destas condições é referida por [EN96a]. No entanto, a sua descrição (que não é muito explícita) parece indicar que esta situação só ocorre num cenário de falhas permanentes. Nomeadamente, quando durante a monitorização se verificar uma total ausência de recepção do que está a ser transmitido. Contudo, experiências levadas à prática numa rede Profibus-DP real, realizadas por [CPA03] [CP04] (na qual o autor colaborou), parecem indicar que algumas falhas transitórias são capazes de induzir este comportamento, nomeadamente quando o delimitador inicial (SD - Fig. 5.5) da trama de *token* possui um erro. Este delimitador encontra-se codificado com uma distância de Hamming de 4 (segundo [EN96a] o objectivo é assegurar um nível de protecção elevado contra deslocações (*shifts*) de bits), pelo que é bastante robusto a interferências. Segundo os resultados obtidos por [CP04] uma alteração importante no delimitador (um erro) é assumida pelo mestre como uma indicação da existência de uma situação grave (aqui denominado como erro *grave*) ao nível da transmissão (de facto a falha é transitória mas é interpretada como permanente);
2. **A transmissão decorreu com erros ligeiros que não foram detectados.** A segunda selecção é escolhida (o_{el}) com uma probabilidade p_{el} . Como a integridade da trama do *token* contra erros de transmissão é apenas assegurada pelo bit da paridade associado a cada carácter transmitido (Fig. 5.5), é possível que ocorram erros na trama que não sejam detectados por outras estações (note-se que o mestre que transmite detecta sempre o erro, porque compara o que recebe com o que transmite). Este tipo de erro (aqui

denominado como *ligeiro*) podem dar origem a perturbações complexas no anel, nomeadamente podem gerar-se endereços de origem (SA - Fig. 5.5) ou destino (DA - Fig. 5.5) errados, e assim originarem passagens indevidas do *token*. Este comportamento já tinha sido observado por [Wil99a]. Quando ocorre esta situação a posição global *TKE* é marcada (ver modelo global);

3. **A transmissão decorreu com erros ligeiros que foram detectados** (por todos os mestres no anel). A terceira selecção é escolhida (o_el_d) com probabilidade p_el_d ;
4. **A transmissão decorreu sem erros.** A quarta selecção vai ser escolhida (o_ok) com probabilidade $(1 - p\_eg - p\_el - p\_el\_d)$.

Nos três últimos casos, a posição *ck_pass* é marcada, indicando que o mestre aguarda por uma confirmação (indirecta) da passagem do *token*. Esta confirmação ocorre se existir actividade no meio de transmissão, representado pela posição global *BUS*, durante um intervalo de tempo inferior ao *slot time*. Este intervalo de tempo é representado pela transição t_ST . O disparo desta transição indica que o *token* não foi passado com sucesso ao próximo mestre no anel. Isto pode ocorrer por vários motivos, alguns já discutidos e outros que serão concretizados adiante.

Existe no entanto uma situação importante que merece ser mencionada. Se durante o *slot time*, o *token* não for aceite pelo próximo mestre (ex. este está fora do anel) e existir qualquer tipo actividade no meio de transmissão (incluindo perturbações e tramas inválidas), o mestre que possui o *token* assume que este foi passado e transita para o estado *active*. Este é de facto um comportamento que leva à perda do *token* (porque de facto este não foi passado com sucesso) e que não foi previsto por [Wil02]. Experiências reais [CP04], demonstram que a probabilidade de ocorrência deste cenário é muito superior à do *token* com erros não detectados.

Para contemplar este comportamento a transição t_ST possui duas selecções. A primeira (o_st) refere-se a uma ausência de perturbações durante este período, e é seleccionada com probabilidade $(1 - p\_e\_st)$. A segunda (o_st_e) corresponde à situação oposta, sendo seleccionada com probabilidade p_e_st .

O Profibus tolera a existência de alguns erros durante a transmissão do *token* através da utilização de um mecanismo de retransmissão. Nesta situação existem três cenários:

1. **Existiu um erro grave.** A marcação da posição tk_eg_ct representa o número de erros graves, sendo neste caso incrementada. Se ocorrerem dois erros ligeiros consecutivos deste tipo o mestre vai para o estado *offline* e o *token* é perdido por não ter sido passado correctamente;

2. **Existiram erros ligeiros.** A marcação da posição tk_el_ct representa o número de erros ligeiros, sendo neste caso incrementada. Se ocorrerem dois erros ligeiros consecutivos o mestre vai para o estado *listen*. No caso do *token* conter erros detectados, o *token* é perdido por não ter sido passado. Se os erros não forem detectados e darem origem a endereços válidos, o *token* pode eventualmente ter sido passado (de forma indevida) a um outro mestre. Caso contrário, o *token* é perdido;
3. **Não existiram erros.** A marcação da posição tk_r_ct representa o número de transmissões, sendo neste caso incrementada. São permitidas até 2 retransmissões do *token*. Caso isto aconteça, o mestre que possui o *token* assume que o próximo mestre (*NS*) não está disponível e vai tentar transmitir o *token* para o sucessor deste e assim sucessivamente. São necessários dois *tokens* consecutivos e iguais (e sem erros detectados) para que um mestre aceite o *token* de um outro mestre que não estava registado como a sua *PS* (*Previous Station*). As posições globais $MAD1...MADn$ estão marcadas, através de um identificador particular⁸³, com os endereços dos mestre da rede de acordo com a sua posição no anel. É assim possível a cada mestre determinar (em termos de modelo) quais os endereços dos seus sucessores.

Este modelo interage de forma muito próxima com o modelo da Transmissão do Token (§5.5.1.2), que é executado imediatamente após o disparo da transição t_TT . Durante esta execução deste último modelo, o modelo da Passagem do Token suspende a sua execução até o primeiro ter terminado. Após esta fase e se o número máximo de erros for atingido, a transição imediata t_tk_me vai disparar removendo o testemunho de ck_pass e colocando o mestre nos estados de *listen* ou *offline*, de acordo com o tipo de erro que ocorreu. Caso contrário, a transição t_ST (*Slot Time*) irá disparar e a execução do modelo procede de acordo com o definido anteriormente.

Durante as várias fases da transmissão do *token*, a posição *BUS* é marcada com diferentes tipos de identificadores simbolizando diferentes tipos de situações referentes ao *token*, como por exemplo: correcto, com erros detectados, com erros não detectados, tipo de perda, etc. Isto permite desenvolver o modelo de forma muito flexível.

5.5.1.2 Transmissão do *Token*

O objectivo deste modelo é representar a sequência de eventos que ocorrem imediatamente após a transmissão do *token* (Fig. 5.6).

Este modelo foi desenvolvido com base no princípio da existência de um anel lógico entre os vários mestres. As posições globais *BUS_UP* e *BUS_DW* vão ser

⁸³ Os *identificadores* representam uma marcação específica de uma posição.

utilizadas para implementar esse mesmo anel lógico. Neste sentido, a posição *BUS_UP* de um modelo vai ser compartilhada com a posição *BUS_DW* do modelo seguinte e assim sucessivamente, criando desta forma um anel lógico. A existência de várias portas e transições serve apenas para separar os mecanismos lógicos em causa.

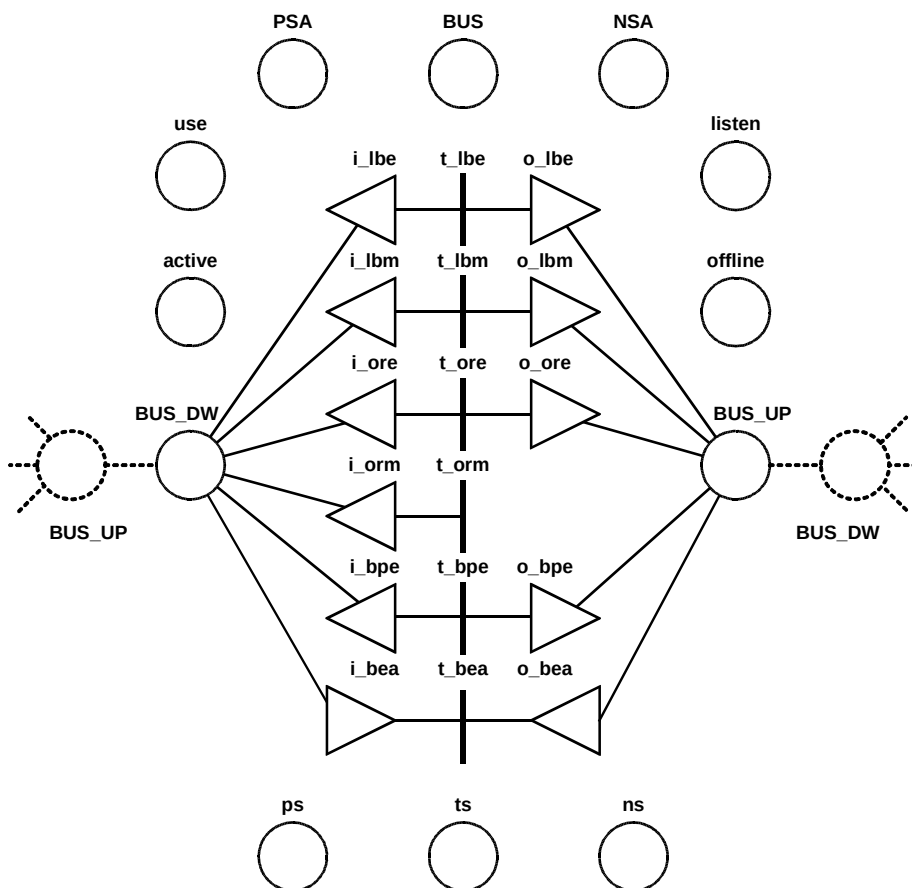


Figura 5.6 - Transmissão do token.

Um mestre conhece à partida quais os endereços do seu antecessor e do seu sucessor no anel. Esta informação está armazenada sobre a forma de um identificador nas posições *ps* e *ns* respectivamente. O endereço do próprio mestre está definido em *ts*. Estas posições são marcadas inicialmente de forma semelhante a *MADn*, e o seu conteúdo pode alterar-se em função do comportamento da rede (excepto *ts*). Exemplos destas situações são a verificação que o seu sucessor não está operacional (após ter esgotado as retransmissões), e a inserção de um mestre no anel.

O funcionamento geral do modelo é o seguinte. Após transmitir o *token* (disparo de *t_TT* no modelo anterior), o mestre que possui o *token* coloca na posição global *NSA* o endereço do mestre que julga ser o seu sucessor (*ns*), na posição global *PSA* o seu próprio endereço (*ts*), na posição *BUS* um identificador referente às características do *token* transmitido (ok, erro, etc.) e na posição *BUS_UP* um

identificador que indica que o *token* vai ser transmitido. Imediatamente, as transições t_{lbe} dos vários modelos vão disparar em sequência, executando operações locais às várias estações. A sequência de disparos termina no mestre que possui o *token* com o disparo da transição t_{lbm} . Permite-se assim que o *token* “circule” pelas várias estações, emulando um anel lógico.

Durante a transmissão do *token* podem ocorrer as seguintes operações nos modelos dos restantes mestres:

- ♦ **Token transmitido correctamente**

Neste caso só o mestre que a quem se destina o *token* vai reagir, ou seja quando $NSA=ts$ e $PSA=ps$ (note-se que ts e ps são posições locais a cada modelo). A verificação desta condição corresponde à aceitação do *token*, implicando uma transição do estado do mestre que o aceita de *active* para *use*. Para informar que o *token* foi aceite, o mestre que o aceitou coloca um identificador em *BUS*. Quando esta informação “chega” ao mestre que possuía o *token*, dá origem a uma transição do estado ck_{pass} para *active*, implicando assim a desabilitação da transição t_{ST} (*slot time*). Note-se que este procedimento corresponde a um comportamento lógico e não real;

- ♦ **Token transmitido com erros não detectados.** Este caso corresponde a um dos seguintes cenários:

- **Endereço de destino errado**

Neste caso NSA vai conter um endereço errado (ver modelo global), e só o mestre a quem se destina ($NSA=ts$, caso exista) pode reagir. Assim, se este mestre estiver no estado *active*, não deve aceitar imediatamente o *token*, mas só após duas recepções consecutivas e iguais⁸⁴ (com origem no mesmo mestre). Caso isto se verifique, a sequência é semelhante à passagem do *token* sem erros, mas com a possibilidade de colocar mestres fora do anel. Este último caso é sinalizado com um identificador em *BUS*;

- **Endereço de origem errado e igual ao da estação**

Neste caso PSA vai conter um endereço errado (ver modelo global) em que $PSA=ts$. O comportamento do mestre depende do estado em que este se encontra. Assim, se estiver em *listen* e receber dois *tokens* consecutivos com as mesmas características, assume que existe uma outra estação no anel com um endereço igual ao seu e transita para o estado *offline*. Se estiver no estado *active*, e se se verificarem as mesmas condições do caso anterior, assume igual pressuposto, mas no entanto transita para o estado *listen*;

⁸⁴ Cada vez que um mestre “lê” o *token* guarda os endereços de destino e de origem. Desta forma pode determinar após receber um outro *token* quais as relações de semelhança com o anterior. Este mecanismo foi implementado através de duas posições auxiliares não representadas, por uma questão de simplicidade, no modelo.

- **Endereço de destino igual ao de origem**

Neste caso *NSA* ou *PSA* estão errados, mas com a particularidade de serem iguais. Nesta situação qualquer mestre no estado *active* que monitorize duas transmissões consecutivas com as mesmas características deve transitar para o estado *listen*.

- ♦ **Saída de mestres do anel**

Quando um mestre aceita um *token* que em princípio não lhe devia ter sido passado, ocorre a saída do anel (transição para *listen*) de todos os mestres que se situam (em termos lógicos) entre a estação que possuía o *token* e a estação que o aceitou. Em relação a este caso existe uma importante divergência em relação a [Wil02]. Neste último trabalho afirma-se que os mestres saem do anel logo que monitorizem um *token* vindo de um mestre antecessor e cujo endereço de destino seja em termos lógicos superior ao seu. Neste caso, a ideia subjacente seria de que o mestre sentir-se-ia “ultrapassado” e assim abandonaria o anel. No entanto nem o standard [EN96a] se refere explicitamente a este tipo de comportamento, nem a verificação experimental [CP04] o comprovou. Verificou-se sim, que este cenário ocorre se os mestres em causa detectarem qualquer actividade (transmissão de tramas) do mestre que aceitou o *token*. Este comportamento tem mais lógica que o anterior, pois só após uma efectiva passagem do *token* é que os mestres se podem sentir “ultrapassados”, caso contrário a estabilidade do anel do Profibus seria muito má.

Ao nível do modelo este comportamento foi modelado da seguinte forma. Durante o disparo da transição t_lhm , o mestre que passou o *token* verifica se foi realizado algum pedido para colocar estações fora do anel (neste caso *BUS* foi marcado com um identificador). Em caso afirmativo, vai ocorrer um disparo sequencial das transições t_ore dos restantes modelos, que vão colocar os mestres intermédios fora do anel. Esta sequência termina com o disparo da transição t_orm , referente ao mestre que aceitou o *token*. Os mestres que são colocados fora do anel são aqueles que em termos lógicos se situam entre o mestre que enviou o *token* e o mestre que o aceitou.

O restante grupo de transições t_hpe e t_bea implementam um anel duplo com funções relacionadas com a inicialização do modelo e com a “limpeza” de situações transitórias. O seu funcionamento é muito simples: o mestre que possui o *token* dá origem a um disparo sequencial das transições t_hpe nos vários modelos, que permitem a execução de operações locais aos mestres. A sequência termina no mestre que possui o *token*. No caso particular da inicialização do modelo, a sequência continua no sentido oposto através do disparo das transições t_bea , terminando de novo no mestre que possui o *token*.

Três aspectos merecem uma nota final. Em primeiro lugar, este modelo assenta na ideia de que após aceitar o *token*, o mestre que o possui vai sempre transmitir algo (nem que seja passar o *token* de novo), e sem erros. Embora isto não seja exactamente verdade, a probabilidade de ocorrer uma sequência consecutiva de erros em todas as tramas é demasiado baixa para ser considerada. Desta forma, assume-se que após aceitar o *token* o mestre vai pelo menos transmitir uma trama (*token*, dados ou gestão do anel) correcta. Note-se que as consequências deste pressuposto só afectam o mecanismo de saída dos mestres do anel. Em segundo lugar, é importante reparar que este modelo é executado em tempo nulo, dado ser apenas constituído por transições imediatas. Por fim, note-se que o modelo só é activado após a transmissão do *token* (t_TT dispara), enquanto que no sistema real as estações reagem durante a transmissão do *token*. No entanto o seu comportamento é equivalente, pois o que interessa é o estado dos mestres após a transmissão do *token*.

5.5.1.3 Rotação do *Token*

O objectivo deste modelo é representar os mecanismos associados à rotação do *token* (Fig. 5.7).

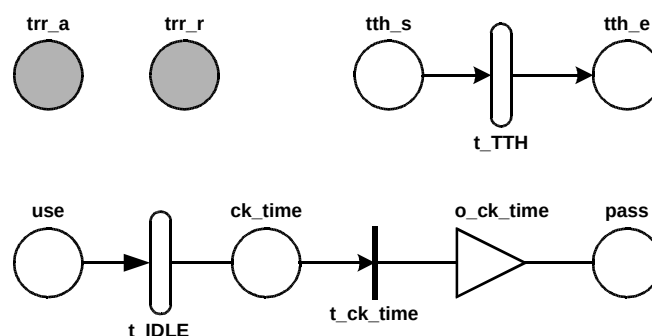


Figura 5.7 - Rotação do *token*.

Após receber o *token* o mestre possui a autorização necessária para realizar uma troca de mensagens⁸⁵ com as restantes estações. Esta troca de mensagens só pode ocorrer durante um intervalo de tempo limitado, definido por um temporizador denominado T_{TH} (*Token Holding Time*).

Imediatamente após a recepção do *token*, o mestre inicia a medida do tempo de rotação do *token*. Esta medida termina na próxima recepção do *token* e resulta no T_{RR} (*Real Rotation Time*). No mesmo instante inicia-se a medida do próximo tempo de rotação do *token*.

⁸⁵ Por uma questão de simplificação de linguagem, deste ponto em diante utilizar-se-á o termo *mensagem* para referir qualquer tipo de troca de dados (transmissões de tramas) entre as estações. Como ao nível semântico existem diferenças, em cada situação será dada uma indicação particular do tipo de mensagem a que se refere o contexto.

O T_{TH} é obtido como a diferença entre o T_{TR} (*Target Rotation Time*), que é um parâmetro de operação da rede, e o T_{RR} .

$$T_{TH} = T_{TR} - T_{RR} \quad (5.1)$$

A implementação deste mecanismo ao nível das redes de Petri não é simples, porque o valor de T_{RR} é função do tempo real e não é um parâmetro do modelo. Este problema foi resolvido recorrendo a duas funcionalidades do Möbius. A primeira diz respeito à utilização do conceito de *posições extendidas*. Este tipo de posições implementa um formalismo semelhante às redes de Petri Coloridas [Wang98]. Mais precisamente, permitem definir a respectiva marcação utilizando uma estrutura de dados (em C++) definida pelo utilizador. A segunda funcionalidade, garante o acesso (via código) ao tempo do próprio simulador e permite conhecer assim o tempo de simulação do modelo.

Utilizando estas funcionalidades, o temporizador T_{TH} foi modelado da seguinte forma. Logo após a recepção do *token*, é calculada (via código) a diferença entre o tempo de simulação actual e a “marcação” da posição *trr_a*. Esta é uma posição extendida, cuja marcação contém o valor do instante de tempo (de simulação) em que foi recebido o último *token* pelo mestre. A diferença assim calculada corresponde ao T_{RR} . Obtido este valor, a posição *trr_r* (também uma posição extendida) é “marcada” com este $T_{TR} - T_{RR}$ valor e a posição *trr_a* é “marcada” com o instante de tempo (de simulação) actual, permitindo o posterior cálculo de T_{RR} quando o *token* for recebido de novo. A marcação da posição *trr_r* contém assim o valor de T_{TH} .

No caso de *trr_r* ser positivo, a posição *tth_s* é marcada e a transição determinística *t_TTH* é habilitada. Esta transição tem uma duração que é igual à “marcação” de *trr_r*. Esta transição corresponde assim à implementação do T_{TH} . Quando esta dispara a posição *tth_e* é marcada correspondendo assim à expiração deste valor. O mesmo acontece se *trr_r* for negativo.

Após receber o *token* (posição *use* marcada) o mestre não têm autorização para começar imediatamente a transmitir mensagens, necessitando de aguardar um intervalo de tempo prévio denominado *tempo de espera* (T_{IDx} - *Idle Timer*). Este tempo tem como objectivo garantir quer a existência de um tempo mínimo de sincronismo entre estações (T_{SYN} - *Syn Interval Timer*), quer assegurar que os tempos de atraso das várias estações (T_{SDR} - *Station Delay Time*) são respeitados. Além disso, este tempo é função do tipo de tramas que se pretendem transmitir. Os (muitos) detalhes que definem o valor para T_{IDx} podem ser encontrados em [EN96a]. Em termos de modelo, este tempo está representado pela transição determinística *t_IDLE*, cuja duração tem em conta todos os aspectos atrás mencionados. Quando esta transição dispara a posição *ck_time* é marcada e a

transição imediata t_{ck_time} dispara, permitindo a execução do código da porta o_{ck_time} . Esta porta vai ser a responsável por muitas funções, entre as quais a verificação das condições para a transmissão de mensagens (Fig. 5.8).

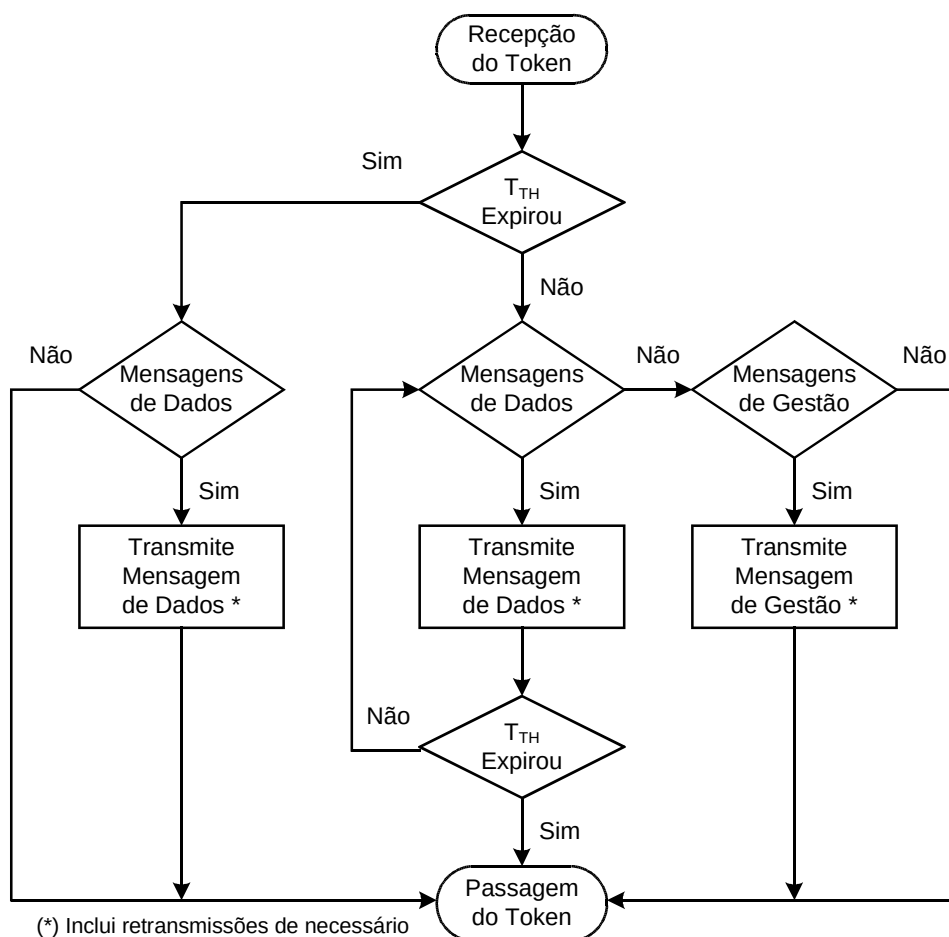


Figura 5.8 - Mecanismo de transmissão de mensagens.

Se não existirem mensagens para transmitir a posição *pass* é marcada de imediato, iniciando-se assim uma nova passagem do *token*.

No caso se existirem mensagens, a sua transmissão é governada pelo valor de T_{TH} . Assim, se quando receber o *token* T_{TH} não tiver expirado (*tth_e* marcada), a estação pode continuar a transmitir mensagens até T_{TH} expirar, ou então até não existirem mais mensagens para transmitir. Se ao receber o *token* e T_{TH} já tiver expirado, pode no máximo transmitir uma mensagem de alta prioridade⁸⁶.

Estando reunidas as condições para transmitir mensagens, a respectiva transmissão obedece a uma certa ordem. Em primeiro lugar são transmitidas as mensagens referentes aos escravos (dados) e posteriormente as mensagens de gestão do anel

⁸⁶ Note-se que no Profibus-DP todas as tramas referentes à transmissão de dados são de alta prioridade. A única exceção que interessa considerar diz respeito apenas às mensagens de gestão do anel.

(caso existam apenas). Cada vez que uma mensagem é transmitida é verificado se T_{TH} expirou ou não. No final da transmissão das mensagens, a posição *pass* é marcada iniciando-se assim uma nova passagem do *token*.

5.5.1.4 Gestão do Anel

O objectivo deste modelo é representar os mecanismos de gestão do anel implementados pela camada de ligação de dados (FDL) (Fig. 5.9).

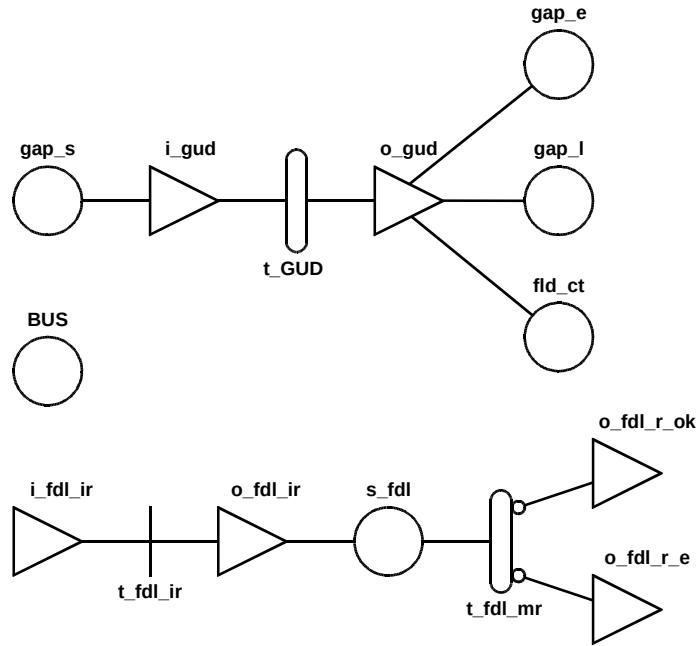


Figura 5.9 - Gestão do anel.

A transmissão de mensagens de gestão do anel (*FDL messages*) [EN96a] é iniciada por um temporizador denominado T_{GUD} (*GAP Update Timer*). O valor desta temporização é definida pela seguintes expressão

$$T_{GUD} = G \cdot T_{TR} \quad (5.2)$$

em que G é denominado *factor de GAP* (*GAP Factor*), que é um parâmetro de operação da rede.

Enquanto o mestre participar no anel (não estiver nem em *offline*, nem em *listen*), deve periodicamente verificar o estado das estações que estão na sua lista de GAP (*GAPL - GAP List*) através do envio de mensagens FDL⁸⁷. Esta lista é constituída pelos endereços de todas as estações (incluindo escravos) que se situam em termos lógicos entre o endereço da estação e o seu sucessor (*ns*) no anel.

⁸⁷ Utilizar-se-à esta denominação por se julgar mais adequada ao contexto.

A existência de uma marcação em gap_s indica que a estação cumpre as condições para enviar mensagens de gestão do anel. Esta marcação habilita a transição determinística t_GUD , que representa o temporizador T_{GUD} , cujo disparo indica a respectiva expiração. Neste caso a posição gap_e é marcada, simbolizando esta ocorrência, e são calculadas o número de estações que existem na GAPL do mestre. Este cálculo é possível, porque o modelo tem acesso quer ao seu endereço (ts), quer ao endereço do seu sucessor (ns). Este cálculo resulta na marcação da posição gap_l .

Após a expiração de T_{GUD} , e caso existam condições para tal (ver rotação do *token*), o mestre transmite uma única mensagem⁸⁸ FDL para uma estação da sua GAPL. As várias transmissões são realizadas de forma ascendente e de acordo com os endereços das estações (independente destas estações existirem ou não). Existe uma situação particular quando o endereço mais elevado é atingido (HSA – *High Station Address*). Neste caso a sequência continua no endereço 0.

O controlo da sequência de transmissões é realizado a partir da posição fdl_ct , que indica quantas mensagens deste tipo foram enviadas até ao momento. Quando este valor atingir o valor dado pela marcação de gap_l , significa que foram verificadas todas as estações na GAPL e o T_{GUD} é re-inicializado.

A inserção de um mestre no anel dá-se se este estiver no estado *listen*, e após receber um pedido para tal. Este pedido é realizado por um outro mestre que ao percorrer as estações da sua GAPL, envia um pedido de inserção no anel (mensagem FDL) a esse mesmo mestre. Neste caso e após receber o pedido, se o mestre em causa reunir um conjunto de condições específicas (ver adiante) envia uma resposta positiva, da qual resulta a respectiva inserção no anel lógico.

A transmissão de mensagens FDL foi implementada no modelo de transmissão de mensagens descrito numa secção adiante. O modelo descrito nesta secção refere-se apenas à recepção e ao envio da resposta a mensagens deste tipo.

Assim e neste cenário, quando o mestre recebe uma mensagem FDL a si destinada e enviada por um outro mestre, a transição imediata t_fdl_ir vai disparar iniciando assim a respectiva resposta. Neste caso *BUS* é marcado com um identificador que indica que vai existir uma resposta (o motivo desta abordagem ver-se-á mais adiante) e s_fdl é também marcada, indicando assim o início da transmissão da mensagem de resposta. A duração desta transmissão (incluindo também o T_{SDR} da estação) é modelada através da transição determinística t_fdl_mr .

⁸⁸ Estas mensagens são de baixa prioridade.

Esta transição possui duas selecções, simbolizando assim a ocorrência, ou não, de erros de comunicação durante a resposta:

- ♦ **Não existiram erros durante a resposta**

É escolhida a primeira selecção (*o_fdl_r_ok*) com probabilidade $(1-p_{e\_fdl\_r})$ e *BUS* é marcado com um identificador que indica uma resposta correcta. O mestre que enviou o pedido actualiza qual o seu sucessor (*ns*) com o endereço do mestre que inseriu no anel e também a respectiva GAPL (*gap_l*). O mestre que foi inserido, actualiza o seu antecessor (*ps*) com o endereço do mestre que lhe enviou o pedido e entra no estado *active*.

Existe aqui um pressuposto que deve ser mencionado. Todos os mestres que não estão em *offline*, mantêm uma lista dos mestres activos (que não estão em *offline*) no anel, denominada LAS (*List of Active Stations*). Esta lista funciona como uma espécie de imagem que o mestre tem do anel, e é construída dinamicamente pelo mestre através da monitorização das tramas que são transmitidas na rede. Segundo [EN96a] a inserção só é possível (resposta afirmativa ao pedido de inserção) após o mestre monitorizar duas passagens completas do *token* pelo anel, tendo assim possibilidade de construir uma LAS consistente.

Tendo em conta as relações temporais entre a rotação do *token* e a inserção no anel, considerou-se que este requisito estava automaticamente cumprido. Ou seja, a resposta é sempre positiva. No entanto, isto significa que o mestre que é inserido deve conhecer à partida (devido à sua LAS) o seu antecessor e seu sucessor no anel. Contudo, no modelo proposto a única parte da LAS que é modelada é a referente ao seu antecessor (*ps*) e sucessor (*ns*), sendo a restante ignorada (para o objectivo pretendido é suficiente).

Em relação ao seu antecessor não existem problemas porque o mestre só será contactado pelo antecessor correcto (o modelo assegura isso), mas em relação ao seu sucessor é mais complicado porque não se implementou o mecanismo de monitorização do meio (seria complicado e além disso iria degradar o desempenho do modelo). Para resolver este problema, optou-se por utilizar uma posição auxiliar (não representada) que é utilizada pelo mestre que envia o pedido para indicar qual o seu sucessor. Assim, o mestre que é inserido utiliza esta informação para designar o seu sucessor. Por de trás desta abordagem está a ideia que o mestre que insere e o mestre que é inserido possuem a mesma LAS e portanto estão de acordo qual é o respectivo sucessor (o mesmo de ambos);

- ♦ **Existiram erros durante a resposta**

É escolhida a segunda selecção (*o_fdl_r_e*) com probabilidade $p_{e\_fdl\_r}$, e *BUS* é marcado com um identificador que indica uma resposta com erros. Neste

caso o mestre que recebeu o pedido actualiza a sua LAS, mas o mestre que recebe a resposta não. Existe assim uma inconsistência ao nível da LAS de ambos os mestres. Contudo e se existirem condições para tal (ver modelo da transmissão de mensagens), o mestre inicial retransmite o pedido.

5.5.1.5 Re-Inicialização do Anel

O objectivo deste modelo é representar os mecanismos utilizados durante a re-inicialização do anel (Fig. 5.10).

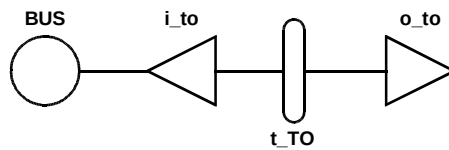


Figura 5.10 - Re-inicialização do anel.

A re-inicialização do anel ocorre quando há uma perda do *token*. Esta situação pode ocorrer pelos motivos previamente discutidos. O Profibus gere este problema, através de um temporizador denominado T_{TO} (*Time-Out Timer*), que monitoriza a existência de actividade no meio de transmissão.

Este temporizador é inicializado com o seguinte valor⁸⁹

$$T_{TO} = 6 \cdot T_{SL} + 2 \cdot n \cdot T_{SL} \quad (5.3)$$

logo após a transmissão ou a recepção do último bit de uma trama. Quando este temporizador expira, significa que o *token* foi perdido e é necessário iniciar a re-inicialização do anel. A forma como este tempo é definido (função do endereço da estação) implica que a tarefa de re-inicializar o anel caiba ao mestre que possui o endereço mais baixo e que não esteja em *offline*.

Quando a posição global *BUS* é marcada com um identificador que indica uma perda do *token* e a estação está no estado *active* ou *listen* (depende do tipo de evento que levou à perda do *token*), a transição determinística t_{TO} fica habilitada. Esta transição tem uma duração igual ao T_{TO} definido para a estação. Após o seu disparo o mestre passa para o estado *use*, indicando assim que possui o *token*.

Existe neste ponto uma imprecisão importante em [Wil02]. Segundo este último, a perda do *token* leva também à perda da LAS de cada mestre, obrigando assim a uma inicialização (e não re-inicialização) do anel. Neste caso o anel ficaria apenas com um mestre activo (o que possui o menor endereço lógico) e com os restantes em *listen* e à espera de serem inseridos, um-por-um, no anel. Contudo, [EN96a] diz expressamente que isto não acontece. Ou seja, após a perda do *token* os mestres

⁸⁹ n é o endereço da estação.

mantêm a sua LAS, e quando T_{TO} expira a rede recupera o seu funcionamento normal. Caso o pressuposto assumido por [Wil02] fosse verdadeiro qualquer perda do *token* iria obrigar à “construção” do anel desde o início.

No entanto o modelo proposto contempla um caso particular deste. Quando são transmitidos consecutivamente dois *tokens* iguais cujo endereço de destino e origem são iguais, isto é interpretado pelos mestres como a sequência original de inicialização do anel. Neste caso assumiu-se que existe perda da LAS em todas os mestres, e dá-se de facto a inicialização do anel. A identificação deste tipo de ocorrência é realizada através de um identificador em *BUS*.

5.5.1.6 Re-Inicialização do Mestre

O objectivo deste modelo é representar os mecanismos de re-inicialização de um mestre (Fig. 5.11).

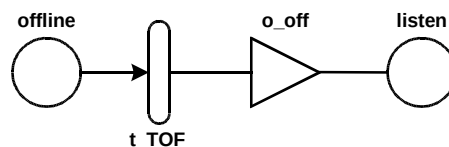


Figura 5.11 - Re-inicialização do mestre.

A passagem de um mestre para o estado *offline* ocorre após a monitorização de várias situações graves (já descritas previamente) e na maioria das aplicações reais resulta num estado que só pode ser recuperado após intervenção humana (*reset* da estação).

Por forma a representar este comportamento, admitiu-se que após um mestre ser colocado em *offline* decorre um tempo (fixo ou não) após o qual o mestre é colocado no estado *listen*. Permite-se assim, que este tempo possa decorrer de uma intervenção humana, ou de um re-inicialização do mestre de forma automática. Esta actividade é representada pela transição t_TOF . Note-se que as características da transição (tipo e parâmetros) vão depender do comportamento específico do sistema.

5.5.1.7 Transmissão de Mensagens

O objectivo deste modelo é representar os mecanismos associados à transmissão de mensagens (Fig. 5.12).

A troca de dados entre as mestres e escravos está organizada sob a forma de mensagens, segundo um modelo do tipo pedido-resposta. Dado que os mecanismos de transmissão de mensagens são independentes da semântica das mesmas, optou-se por um modelo capaz de representar a transmissão de todo o tipo de mensagens.

As transmissão das mensagens é realizada através de um conjunto de serviços implementados ao nível da *User Interface*, que posteriormente são mapeados, utilizado a sub-camada DDLM (*Direct Data Link Mapper*), em serviços correspondentes da camada de ligação de dados. Assume-se que os erros nas mensagens são sempre detectados pela estação a quem se destina, que neste caso, e segundo [EN96a], deve ignorá-las.

[EN96a] e [EN96b] prevêem a existência de vários tipos de mensagens trocadas entre as estações. Baseado nas condições típicas de utilização de uma rede Profibus-DP, optou-se por considerar que só uma parte destas são de facto utilizadas, considerando-se assim as seguintes:

- ♦ Troca de dados entre mestres e escravos;
- ♦ Diagnóstico dos escravos;
- ♦ Parametrização dos escravos;
- ♦ Configuração dos escravos;
- ♦ Comandos de controlo globais (*DDLM_Global_Control* [EN96b]).

Os serviços associados a estas mensagens são mapeadas no serviço SRD (*Sent and Request Data with Reply*) da camada de ligação de dados, excepto a última que é mapeada no serviço SDN (*Sent Data with No Acknowledge*). O modelo também inclui a transmissão de mensagens de gestão do anel, associadas à camada de ligação de dados (ver modelo de gestão do anel).

Quando estão reunidas as condições para a transmissão de uma mensagem de pedido (ver rotação do *token*) as posições *msg_p_s* e *msg_tp* são simultaneamente marcadas. A primeira, indica que se vai iniciar a transmissão de uma mensagem. A segunda, indica, através da sua marcação, qual o tipo de mensagem que se vai transmitir. Este tipo é definido por um identificador utilizado para cada tipo de mensagem transmitida. Este identificador vai permitir também determinar qual a duração da mensagem e quais as respectivas probabilidades de transmissão com sucesso. Além disso, a posição global *NSA* é marcada com o endereço da estação para a qual se pretende transmitir a mensagem (cada mestre sabe à partida quais as estações com que vai comunicar), garantindo assim que apenas a estação a quem se destina a mensagem a vai processar.

A transição *t_msg* representa a transmissão da mensagem de pedido, cuja duração é função do seu tipo. Esta transição possui duas selecções, simbolizando assim a ocorrência, ou não, de falhas de comunicação (erros) durante a transmissão da mensagem.

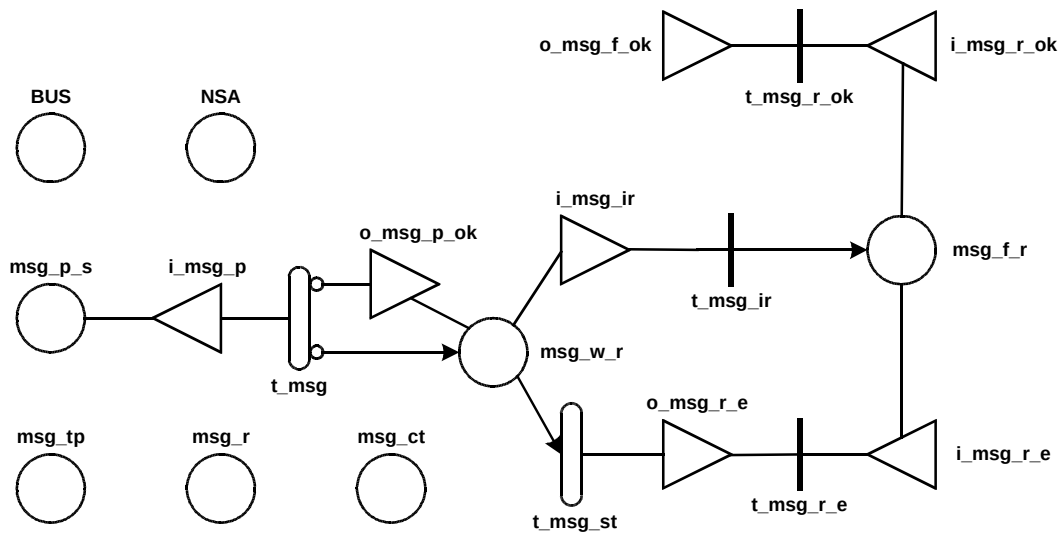


Figura 5.12 - Transmissão de mensagens.

No caso de não existirem erros durante a transmissão da mensagem de pedido, é escolhida a primeira selecção ($o\_msg\_p\_ok$), com uma probabilidade do tipo ($1-p\_e\_XX$) em que p_e_XX representa a probabilidade de transmissão sem sucesso (com erros) de uma mensagem do tipo XX . Neste caso *BUS* é marcado com um identificador do tipo de pedido. O mestre que enviou a mensagem fica a aguardar pela respectiva resposta, marcando para isso a posição msg_w_r .

Se existirem erros durante a transmissão da mensagem de pedido, é escolhida a segunda selecção com uma probabilidade do tipo p_e_XX e a posição msg_w_r é marcada. A marcação desta posição vai permitir habilitar a transição determinística t_msg_st , que representa o *slot time* associado à espera de uma resposta. O disparo desta transição indica que não existiu resposta, porque a mensagem de pedido continha erros e não foi processada pelo eventual destinatário. Neste caso o Profibus-DP prevê a possibilidade de se retransmitir a mensagem (limitado a uma retransmissão [EN96b]). Ao nível do modelo, a posição msg_r vai ser utilizada como contador de retransmissões. Enquanto o número máximo não for atingido, o mestre continua a tentar transmitir. Neste sentido a posição msg_p_s é marcada de novo. Caso contrário, é marcada a posição *use* seguindo-se o mecanismo normal verificação das condições de T_{TH} (ver rotação do *token*).

Voltando ao caso da não existência de erros, após a transmissão da mensagem de pedido (msg_w_r marcada) a estação que recebe o pedido marca a posição *BUS* com um identificador indicando que recebeu esse pedido. Seguidamente, a transição imediata t_msg_ir vai disparar e marcar a posição msg_f_r . O disparo desta transição tem como objectivo desabilitar a transição t_msg_st e assim permitir que a resposta possa ter a duração prevista. Note-se que vai sempre existir uma resposta (mesmo com erros) porque o pedido foi enviado correctamente.

Na posição *msg_f_r*, o mestre aguarda pelo fim da resposta. Esta espera é função da duração da resposta e é controlada pelo modelo do escravo ou por um outro modelo do mestre (mensagens FDL). Tal como o pedido, a resposta pode conter erros. A indicação do tipo de resposta faz-se através de um identificador na posição *BUS*.

No caso de existirem erros na resposta, a transição imediata *t_msg_r_e* dispara originando uma sequência de operações semelhantes às indicadas para os erros durante a transmissão da mensagem de pedido. No caso contrário, a transição imediata *t_msg_r_ok* dispara indicando que recebeu uma resposta correcta. Este disparo vai permitir executar o código da porta *o_msg_f_ok*, que por sua vez vai desencadear um conjunto de operações. O tipo de operações realizadas depende do tipo de pedido e do tipo de resposta. No caso de mensagens trocadas com os escravos, vão também depender do estado em que cada escravo se encontra (ver modelo de gestão dos escravos). De qualquer forma, no final destas operações posição *use* é marcada, seguindo-se o mecanismo normal verificação das condições de T_{TH} .

No caso de mensagens sem resposta (*DDLML_Global_Control*), não são considerados a existência de erros nos pedidos, dado que não têm qualquer influência. Assim e após a transmissão da mensagem, passa-se logo ao estado *use*.

A posição *msg_ct* é utilizada para indicar que pelo menos uma mensagem foi transmitida, estando assim relacionada com os mecanismos de rotação do *token*.

5.5.1.8 Gestão dos Escravos

O objectivo deste modelo é representar os mecanismos de gestão dos escravos, incluindo as violações das *deadlines* associadas às mensagens de dados trocadas entre mestres e escravos.

A troca de mensagens entre mestres e escravos no Profibus-DP é governada por várias máquinas de estados cujo comportamento é complexo [EN96b]. A *User Interface* de cada mestre, implementa uma máquina de estados global (*Scheduler*), que governa o comportamento do mestre e uma máquina de estados para cada escravo associado ao mestre (*Slave-Handler*).

Um mestre pode estar num dos seguintes estados⁹⁰:

- ♦ **Offline.** As comunicações com as restantes estações estão bloqueadas. O mestre está fora do anel;

⁹⁰ Utilizaram-se as denominações definidas em [EN96b].

- ♦ **Stop.** A camada de ligação de dados (FDL) está activa, ou seja o mestre toma parte activa no anel. São apenas possíveis comunicações com mestres DPM2 para configuração;
- ♦ **Clear.** O mestre tenta parametrizar e verificar as configurações dos escravos. Além disso, são também realizadas trocas de dados normais com escravos. No entanto, os dados transmitidos para os escravos têm o valor 0;
- ♦ **Operate.** Troca normal de dados com os escravos.

Os estados desta máquina de estados podem ser mapeados nos estados da camada de ligação de dados. Assim, o estado *Offline* corresponde ao estado equivalente da FDL, enquanto os restantes correspondem aos estados que a FDL ocupa quando o mestre toma parte activa no anel.

Quando é invocado um serviço ao nível da *User Interface*, este é enviado sob forma de um pedido à máquina de estados global. Esta por sua vez reencaminha esse pedido para a máquina de estados do escravo a quem o pedido se destina, que após o processamento envia a o pedido e espera pela respectiva resposta. Esta resposta segue o percurso inverso até ao serviço invocado na *User Interface*. As máquinas de estado dos escravos utilizam os serviços implementados na sub-camada DDLM para implementar as sequências pedido-resposta.

Cada máquina de estados dos escravos é gerida de forma independente e reage de acordo com o estado do escravo. Os escravos são processados de acordo com uma sequência fixa. Cada pedido enviado para a máquina de estados é mapeado num único pedido de serviço ao nível da DDLM. Ou seja, após receber a resposta a esse pedido (afirmativa ou não), passa-se a processar o pedido para o próximo escravo e assim sucessivamente.

Embora a máquina de estados dos escravos seja complexa, é possível, grosso modo, agrupar os respectivos estados da seguinte forma:

- ♦ Diagnóstico;
- ♦ Parametrização;
- ♦ Verificação da configuração;
- ♦ Troca de dados de controlo do processo industrial.

Não sendo lógico representar na totalidade o comportamento de todas as máquinas de estados, optou-se por reduzi-la ao seu funcionamento típico. Assim, a máquina de estados do mestre pode ser deduzida a partir dos vários estados da camada de ligação de dados. A máquina de estados do escravo foi condensada, considerando-se que cada estado corresponde a uma sequência típica de troca de mensagens.

De acordo com esta abordagem, foram identificadas as seguintes sequências de trocas de mensagens (Fig. 5.13):

- ♦ Inicialização do escravo, correspondendo à seguinte sequência de mensagens:
 - Diagnóstico inicial;
 - Envio de parâmetros;
 - Verificar configuração;
 - Diagnóstico final.
- ♦ Troca de mensagens de dados referentes ao controlo do processo industrial. Esta sequência corresponde à situação normal;
- ♦ Troca de mensagens de dados, mas com indicação da necessidade de diagnóstico. Se o escravo enviar uma resposta com um pedido de diagnóstico, a próxima mensagem a enviar para o escravo corresponde a um pedido deste tipo. Só após a conclusão com sucesso deste mesmo pedido, é que é possível retornar à sequência normal de troca de dados.

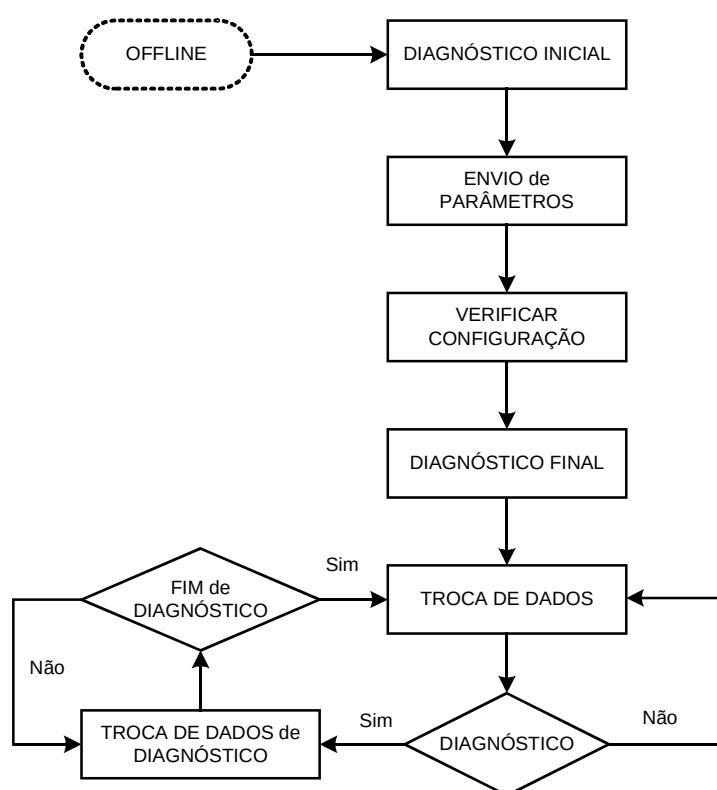


Figura 5.13 - Sequência de mensagens.

A troca de mensagens de dados corresponde ao funcionamento normal (a estação toma parte activa no anel). A sequência de inicialização é necessária após a estação transitar do estado *offline*. Note-se que se excluiu a existência de erros de

parametrização e de configuração, dado que este aspecto só depende do (bom) desempenho de quem configura a rede. A máquina de estados assim implementada corresponde ao funcionamento normal e real.

Para que possam ser transmitidas as sequências de mensagens destinadas aos escravos, é necessário conhecer qual o estado em que cada escravo se encontra (Fig. 5.14). Esta informação é mantida, para cada escravo, por intermédio de duas posições. A posição global $SLVx$ possui o endereço do escravo x associado ao mestre. Como cada escravo está associado a um único mestre é possível conhecer o seu endereço à partida, facilitando assim a implementação do mecanismo de envio de mensagens. O estado do escravo é indicado pela posição slv_x_op , cuja marcação (identificador) representa esse mesmo estado.

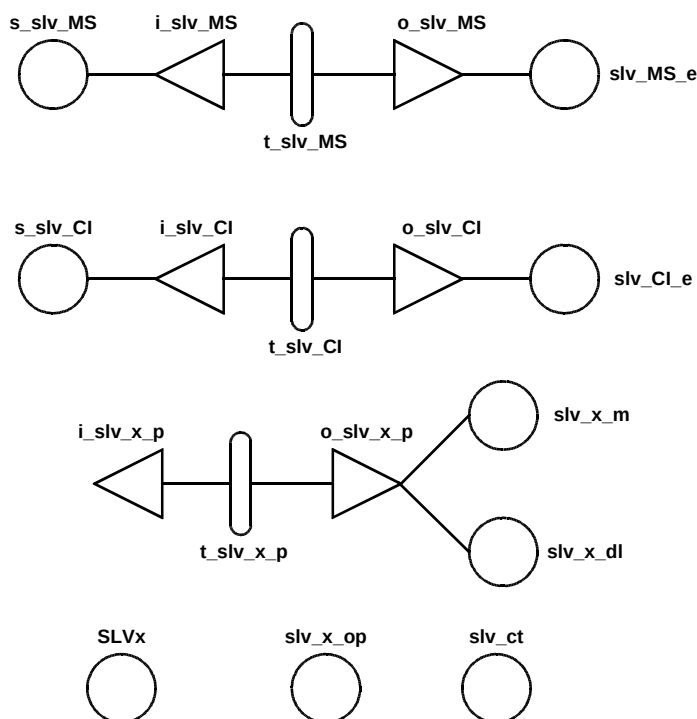


Figura 5.14 - Gestão dos escravos.

A troca de mensagens de forma cíclica definida pelo Profibus-DP [EN96b], é realizada por intermédio do funcionamento sincronizado de dois temporizadores. O T_{MSI} (*Min Slave Interval*) especifica o intervalo mínimo para a troca de mensagens com os escravos (*poll cycle*). Enquanto este temporizador não expirar, não são transmitidas mensagens para os escravos. Quando expira, inicia-se a troca de mensagens. Para cada escravo é transmitida uma mensagem de cada vez, e os escravos são processados de forma sequencial. Quando forem transmitidas todas as mensagens (com sucesso ou não) o temporizador é re-inicializado.

O T_{CI} (*Master Control Interval*) especifica o período de actualização do estado (*status*) dos dados do processo (enviados e recebidos) na interface entre a aplicação

e a *User Interface*. Além disso, quando expira é enviado também um comando global para todos os escravos, indicando qual o estado do mestre. A sua re-inicialização só pode ocorrer durante a transmissão de mensagens.

Ao nível do modelo, o temporizador T_{MSI} é implementado por intermédio da transição determinística t_{slv_MS} . Esta transição é habilitada através da marcação da posição s_{slv_MS} . Quando a transição dispara a posição slv_MS_e é marcada. A transição só é de novo habilitada quando forem transmitidas as mensagens para todos os escravos associado ao mestre. Isto é controlado através da posição slv_ct , cuja marcação é incrementada de cada vez que é transmitida uma mensagem (com sucesso ou não) para um escravo. Quando a marcação desta posição atinge o número de escravos associados ao mestre, sabe-se que todos foram já contactados.

O temporizador T_{CI} é implementado de forma análoga, utilizando as posições s_{slv_CI} (início), slv_CI_e (expiração) e a transição t_{slv_CI} (duração). O temporizador só é re-inicializado durante a fase de transmissão de mensagens. Neste caso é enviado um comando global (*DDLM_Global_Control*) para as restantes estações.

Este modelo tem uma forte interacção com o modelo de rotação do *token*, que define as condições para a transmissão das mensagens, e com o modelo de transmissão de mensagens, que implementa a respectiva transmissão. Note-se que a transmissão das mensagens vai ser realizada de acordo com o estado do escravo, que por sua vez podem afectar o respectivo estado.

É importante referir que o comportamento aqui descrito só é válido se ao nível da *User Interface* esta não ocupar o estado *Clear*. Optou-se por este pressuposto, dado que corresponde à situação mais frequente na prática.

A violação das *deadlines* das mensagens é implementada de acordo com um esquema clássico proposto em [TV99a]. A cada mensagem de dados⁹¹ está associada uma *deadline*, cuja duração é igual ao seu período de produção. Este período é definido de forma conceptual e não está relacionado com T_{CI} . Assumiu-se também que para cada escravo só é transmitida e uma mensagem de dados⁹². Se após a produção da mensagem de dados e num intervalo de tempo igual sua *deadline*, esta não for correctamente transmitida considera-se que existiu uma avaria da rede.

A produção da mensagem de dados referente ao escravo x é definida através da transição determinística $t_{slv\_x\_p}$, cuja duração corresponde ao seu período de produção. Esta transição está permanente habilitada para corresponder a uma

91 As mensagens de dados estão associadas ao controlo do processo.

92 Esta restrição pode ser facilmente relaxada bastando para isso indicar quantas mensagens estão associadas a cada escravo.

produção contínua de mensagens. Quando esta transição dispara, a marcação da posição *slv_x_m* é analisada, resultando num dos seguintes cenários:

- ♦ **A posição não está marcada.** Isto significa que a mensagem produzida anteriormente já foi transmitida. Neste caso *slv_x_m* é de novo marcada indicando que existe uma nova mensagem para transmitir. Quando esta mensagem for transmitida com sucesso (ver modelo de transmissão de mensagens), a marcação desta posição é removida;
- ♦ **A posição está marcada.** Neste caso significa que a mensagem produzida anteriormente ainda não foi transmitida. Ou seja, existiu uma violação da respectiva *deadline*. Nesta situação a posição *slv_x_dl* é marcada indicando esta ocorrência. A marcação desta posição só é removida quando a mensagem for transmitida (pedido e resposta) com sucesso.

Como nota final refira-se que o modelo não inclui a possibilidade da aplicação enviar directamente comandos globais (ex. *Sync* ou *Freeze* [EN96b]). Esta opção resultou do facto deste comportamento ser dependente das características da própria aplicação. No entanto refira-se que esta funcionalidade pode ser facilmente incluída, utilizando para isso a parte do modelo referente às mensagens de controlo global, dado que o seu comportamento é idêntico.

5.5.2 Modelo do Escravo

Este modelo descreve o comportamento de uma estação escrava (Fig. 5.15). O escravo recebe pedidos por parte do mestre ao qual está associado e envia as respectivas respostas. Este modelo foi desenvolvido segundo os mesmos princípios do modelo de mensagens do mestre.

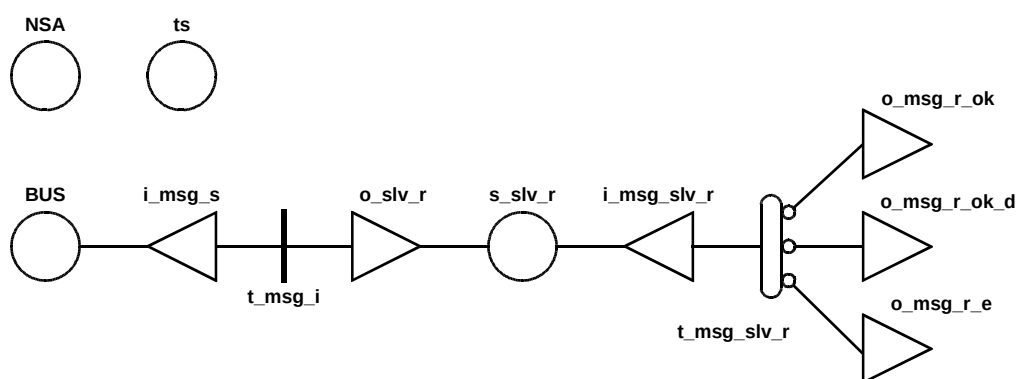


Figura 5.15 - Modelo do escravo.

Quando um mestre envia um pedido (sem erros) a um escravo associado, marca a posição global *BUS* com um identificador que indica qual o tipo de pedido e a posição *NSA* com o endereço do escravo respectivo. Este último, sabe que a

mensagem é dirigida a si porque compara com o seu próprio endereço, definido pela marcação da posição ts (do escravo).

Se o pedido for dirigido a esse escravo, a transição imediata t_{msg_i} dispara, marcando BUS com um identificador que indica que vai ser enviada uma resposta. Este identificador vai desabilitar a transição $t_{msg_{st}}$ do modelo de transmissão de mensagens do mestre. Além disso, a posição s_{slv_r} é também marcada com a identificação do tipo de pedido. Este identificador vai permitir definir a duração do tipo de resposta e quais as probabilidades de transmissão com sucesso da mesma.

A duração da transmissão da resposta (incluindo também o T_{SDR} da estação) é modelada através da transição determinística $t_{msg_{slv_r}}$, cuja duração é função do tipo de pedido realizado. Esta transição possui três selecções, simbolizando assim a ocorrência, ou não, de falhas de comunicação (erros) durante a transmissão das mensagens.

No caso de não existirem erros durante a transmissão da mensagem de resposta, é escolhida a primeira selecção ($o_{msg_r_{ok}}$), com uma probabilidade do tipo $(1-p_e_{XX})$ em que p_e_{XX} representa a probabilidade de transmissão sem sucesso (com erros) da resposta a uma mensagem do tipo XX . Neste caso BUS é marcado com um identificador que indica que a resposta está correcta, permitindo assim ao modelo de mensagens do mestre processar a resposta.

A segunda selecção corresponde a uma variante a primeira, e é utilizada para indicar pedidos de diagnóstico. Neste caso assume-se que quando se recebe uma mensagem de pedido de dados, existe uma probabilidade p_{diag} do escravo indicar na resposta um pedido de diagnóstico. Esta indicação é realizada através da marcação da posição BUS .

Se existirem erros durante a transmissão da mensagem de resposta é escolhida a terceira selecção com uma probabilidade do tipo p_e_{XX} , e a posição BUS é marcada de forma correspondente.

Refira-se por fim, que os escravos implementam um mecanismo de *watchdog* referente às comunicações. Mais concretamente, se não existirem comunicações com o mestre durante um intervalo de tempo pré-definido as saídas do escravo são colocadas num estado seguro (*fail-safe*). A modelação deste mecanismo permitiria estabelecer a diferença entre avarias seguras ou não. Contudo, como esta é uma característica da própria aplicação, optou-se por não a implementar.

5.5.3 Modelo Global

Este modelo vai permitir a interacção entre os modelos dos mestres e dos escravos, sendo além disso também responsável por modelar os tipos de erros não detectados que ocorrem no *token* (Fig. 5.16).

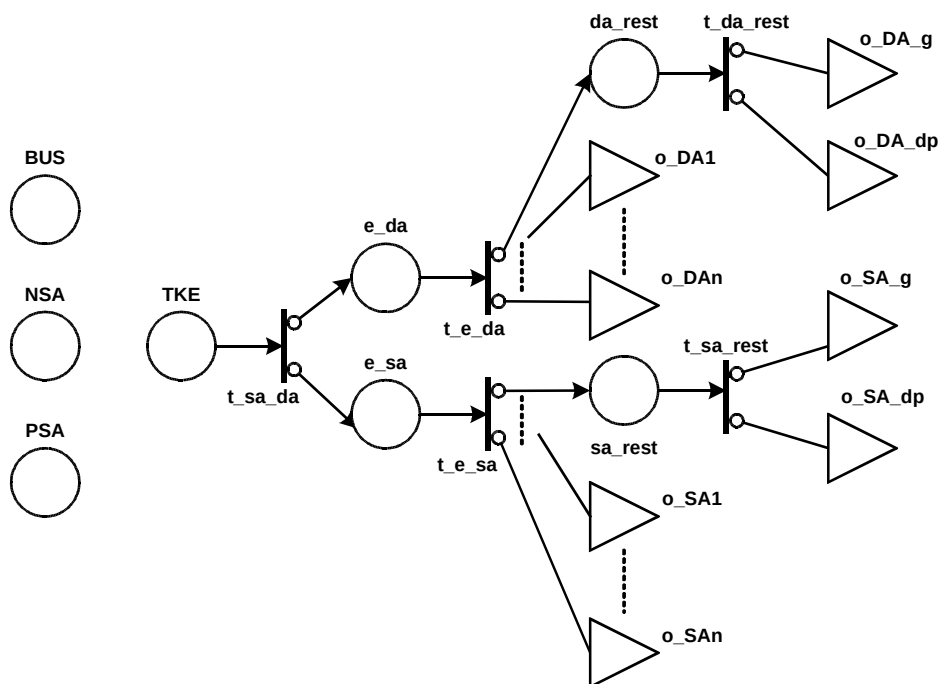


Figura 5.16 - Modelo global.

No desenvolvimento deste modelo assumiu-se que um erro no *token* não detectado só atinge o endereço de destino ou de origem, mas não ambos. Este pressuposto baseia-se no facto de que a probabilidade de que ambos os endereços conterem erros, e que estes não serem detectados, é extraordinariamente baixa e portanto desprezável.

Quando é gerado um *token* com um erro não detectado, a posição *TKE* é marcada e a transição imediata *t_sa_da* fica habilitada. Esta transição tem duas selecções, correspondendo cada uma a um erro no endereço de destino e de origem respectivamente. A escolha da primeira selecção marca a posição *e_da*, e a segunda a posição *e_sa*. As selecções têm uma probabilidade de 0.5 associada a cada uma.

Um erro num endereço (de origem ou destino) pode manifestar-se da seguinte forma:

- ♦ O endereço errado não corresponde a nenhuma estação existente. Neste caso a primeira selecção de *t_e_sa(da)* é escolhida e posição *da_rest* (*sa_rest*) é marcada;

- ♦ O endereço errado corresponde a uma estação no anel. Neste caso a selecção escolhida o_DA_i (o_SA_i) vai marcar *NSA* (*PSA*) com o endereço do mestre i , alterando assim o seu valor original. Note-se que é necessário incluir uma escolha por cada mestre no anel. Assume-se neste caso que a probabilidade de alterar qualquer endereço é a mesma.

No caso do endereço gerado não corresponder a nenhuma estação existente, pode ainda surgir o problema de existirem 2 *tokens* consecutivos e iguais. Por forma a incluir esta hipótese (diga-se muito remota), admitiu-se que existe uma probabilidade p_e_dup deste cenário ocorrer (nas secções seguintes apresentar-se-á a forma de a calcular). Este comportamento é representado pela transição imediata t_da_rest (t_sa_rest), com as correspondentes selecções. Desta forma, se a primeira selecção for escolhida (o_DA_g ou o_SA_g), isto corresponde a gerar um endereço diferente do anterior. No caso da segunda selecção for escolhida (o_DA_dp ou o_SA_dp), é gerado um endereço igual ao anterior. O endereço gerado vai ser utilizado para vai marcar *NSA* ou *PSA* conforme o caso. É utilizado um identificador especial para distinguir as duas situações.

Por fim importa referir, que se este modelo gerar endereços que acabem por não modificar os conteúdos de *NSA* ou *PSA*, o modelo do mestre considera que não existiu qualquer erro no *token*.

5.5.4 Parâmetros do Modelo

No modelo proposto existem dois tipos de parâmetros, ambos relacionados com as transmissões das mensagens: as probabilidades e as durações das transições. Nas secções seguintes são discutidas a forma como estes parâmetros podem ser determinados para um caso geral.

5.5.4.1 Probabilidades Associadas às Transições

Num contexto associado às comunicações, o modelo de Poisson proposto para a ocorrência das falhas é usualmente apresentado de outra forma, sendo caracterizado por um parâmetro denominado *taxa de erro do bit* (BER – *Bit Error Rate*). Este parâmetro é definido como a probabilidade de ocorrer um erro num bit, ou seja, como o complementar da probabilidade de não ocorrerem quaisquer falhas durante a transmissão do bit. Este valor pode ser facilmente calculado a partir das propriedades do processo de Poisson (ver capítulo 3) como

$$BER = 1 - \exp(-\lambda \cdot T_{bit}) \quad (5.4)$$

em que λ é a taxa de ocorrência das falhas e T_{bit} a duração de um bit. Implícito a esta expressão está o facto de se assumir que esta probabilidade é independente do valor efectivo do bit, o que é um pressuposto credível.

Como discutido previamente, as tramas no Profibus são transmitidas como sequências de caracteres, cada um é constituído por 11 bits: 1 *start* bit, 8 bits de dados, 1 bit de paridade e 1 *stop* bit (Fig. 5.17).

Trama do Token

SD	DA	SA

Trama sem Dados

SD	DA	SA	FC	FCS	ED

Trama com Tamanho Fixo para os Dados

SD	DA	SA	FC	DATA	FCS	ED

Trama com Tamanho Variável para os Dados

SD	LE	LEr	SD	DA	SA	FC	DATA	FCS	ED

SD – Start Delimiter

SA – Source Address

DA – Destination Address

FC – Frame Control

FCS – Frame Check Sequence

ED – End Delimiter

LE – Frame Length

LEr – Frame Length Repeated

Figura 5.17 - Formato das tramas do Profibus.

Se se assumir que um erro num bit da trama é suficiente para corromper o seu conteúdo (ou seja, dar origem a erros), a probabilidade da transmissão correcta de uma trama $Prob_{TC}$ pode ser obtida como a probabilidade da transmissão correcta de todos os seus bits

$$Prob_{TC} = (1 - BER)^n \quad (5.5)$$

em que n é o numero de bits da trama. Utilizando esta expressão é possível calcular as probabilidades de transmissão com sucesso (ou não) de todas as tramas, e assim determinar os respectivos parâmetros associados às transições do modelo.

Para todas os tipo de tramas, excepto a do *token*, assume-se que o erro é sempre detectado. Cada trama do Profibus (excepto a do *token*) contém um campo (1 caracter) denominado FCS (*Frame Check Sequence*) que é utilizado para detectar erros de transmissão. Em [Wil99a] é analisada a eficiência deste mecanismo em comparação com um código de detecção de erros (CRC) de 32 bits. Embora se demonstre a pior qualidade do FCS em relação ao CRC, os resultados apresentados

indicam que a probabilidade de ocorrerem erros que não são detectados neste tipo de tramas é muito baixa ($<10^{-9}$). Fica assim demonstrado que o pressuposto assumido pode ser considerado válido.

Em relação ao *token* a situação é mais complexa, pois nem todos os erros são detectados. [Wil99b] desenvolve uma expressão para a probabilidade do *token* possuir até 2 erros sem serem detectados $Prob_{TKE\_2E}$ que a seguir se reproduz

$$\begin{aligned} Prob_{TKE\_2E} &= \binom{33}{2} BER^2 \cdot (1 - BER)^{(33-2)} \cdot \frac{18 \cdot 8}{\frac{33!}{(33-2)!}} \\ &= \frac{144}{1056} \cdot \binom{33}{2} BER^2 \cdot (1 - BER)^{(33-2)} \end{aligned} \quad (5.6)$$

Esta expressão pode ser facilmente deduzida aplicando o seguinte raciocínio. Embora a trama do *token* não inclua um FCS, se ocorrer um erro no delimitador inicial ou no *start* ou *stop* bits dos restantes caracteres da trama, este é sempre detectado (estes dois últimos pela UART). Além disso, como a paridade é ímpar, qualquer número ímpar de erros é também detectado. Se nos restringimos apenas à ocorrência de 2 bits com erros, estes só podem ocorrer, e não serem detectados, se ambos ocorrerem no mesmo carácter (SA ou DA, Fig 5.17). A probabilidade de ocorrerem 2 erros nos 33 bits da trama do *token* é dada por uma distribuição Binomial

$$\binom{33}{2} BER^2 \cdot (1 - BER)^{(33-2)} \quad (5.7)$$

O número de combinações dos bits errados dentro da trama, por forma a não serem detectados é $18 \cdot 8$, dado que para o primeiro bit existem 18 possibilidades (SA e DA incluindo os bits de paridade) e para o segundo apenas 8. Dado que o número de arranjos para colocar 2 bits nos 33 é $33!/(33-2)!$, a expressão fica automaticamente deduzida. Note-se, que podem também ser deduzidas expressões para um maior número de erros, contudo o valor obtido para a probabilidade de um erro não detectado é de tal forma baixo que pode ser desprezado. A equação (5.6) pode ser assim utilizada para obter a probabilidade de um erro ligeiro no *token* não detectado.

Por fim, falta apenas analisar a situação de um erro grave no *token*. Não existindo informação concreta sob as condições em que este erro ocorre, assumiram-se pressupostos de certa forma intuitivos para o seu cálculo. Como este erro só ocorre se o delimitador inicial (SD – *Start Delimiter*) for perturbado, e que este se encontra “protegido” por uma distância de Hamming de 4, assumiu-se que este erro só ocorre se forem modificados 4 ou mais bits (quaisquer) nesse delimitador,

independentemente dos restantes bits da trama. Neste caso a respectiva probabilidade $Prob_{TKE\_grave}$ pode ser definida como

$$Prob_{TKE\_grave} = \sum_{k=4}^{11} \binom{11}{k} BER^k (1-BER)^{(11-k)} \quad (5.8)$$

seguindo assim um raciocínio semelhante ao que foi utilizado para o cálculo do erro não detectado no *token*. Note-se que o delimitador é um carácter fixo e conhecido à partida, assumindo-se assim que a estação que envia o *token* tem capacidade para determinar o número de erros. Pode-se também concluir daqui o motivo pelo qual se considerou que um erro na delimitador inicial é sempre detectado na estação a quem o *token* é dirigido.

Os parâmetros da transição t_{TT} (ver modelo da passagem do *token*), ficam definidos de acordo com a Tabela 5.1.

Parâmetro	Equação
Probabilidade de transmissão com sucesso	Equação (5.5)
Probabilidade de erro grave	Equação (5.8)
Probabilidade de erro ligeiro não detectado	Equação (5.6)
Probabilidade de erro ligeiro detectado	$1 - [(5.5) + (5.8) + (5.6)]$

Tabela 5.1 - Parâmetros da transição t_{TT} .

Se se assumir que um *token* com erros não detectados pode conter qualquer endereço (*DA* ou *SA*) na gama 0...255 e que estes endereços errados têm todos igual probabilidade de ocorrência, então se a rede campo possuir M mestres as probabilidades associadas às transições t_{e_da} e t_{e_sa} (§5.5.3) podem ser definidas de acordo com a Tabela 5.2.

Seleccção	Equação
1	$1 - M / 255$
2 ... M+1	$1 / 255$

Tabela 5.2 - Parâmetros das transições t_{e_da} e t_{e_sa} .

No caso de um erro duplo referente a endereços de mestres não existentes que estão associados às transições t_{da_rest} e t_{sa_rest} (§5.5.3), a respectiva probabilidade $Prob_{TKE\_duplo}$ pode ser definida como

$$Prob_{TKE\_duplo} = 1 / (255 - M) \quad (5.9)$$

A probabilidade de ocorrerem erros durante o *slot time* $Prob_{E\_ST}$, pode ser obtida de forma análoga à equação (5.4)

$$Prob_{E\_ST} = 1 - \exp(-\lambda \cdot (T_{ST} - T_{SYN})) \quad (5.10)$$

em que T_{ST} corresponde ao *slot time* e T_{SYN} ao tempo de sincronização [EN96a]. A inclusão deste último termo resulta do facto de que só após o intervalo T_{SYN} é que o mestre de facto monitoriza o meio de transmissão.

5.5.4.2 Duração das Transições

A duração das transições existentes nos modelos, são obviamente função da velocidade de transmissão da rede. Isto implica que quando se pretende avaliar o comportamento da rede para diferentes velocidades se tenha que recalculer todos estes parâmetros. No entanto, como a esmagadora maioria das transições estão associadas à transmissão de mensagens, é possível definir a sua duração em função do número de caracteres transmitidos. Obtêm-se assim um modelo independente da velocidade de transmissão.

No entanto existem ainda algumas transições que são função da velocidade da rede. Estão nesta categoria: T_{IDLE} (*Idle Time*), T_{SL} (*Slot Time*) e T_{SDR} (*Station Delay Response*). Valores típicos para estes parâmetros podem ser encontrados em [EN96b]. Refira-se que mesmo neste caso é possível utilizar a abordagem acima descrita.

Todas as transições temporizadas foram implementadas com uma política de concorrência do tipo servidor único (*single server*), e uma política de memória do tipo PRD (PRD - *Preemptive Repeat Different*).

O tempo de propagação dos sinais no meio físico de transmissão não foi considerado, contudo, se necessário, pode ser facilmente incluído no modelo, bastando para isso adicionar o respectivo valor à duração de cada transição temporizada associada à transmissão das tramas.

5.5.5 Definição das Medidas de Interesse

A probabilidade da *deadline* da mensagem de dados associada ao escravo x ser violada, pode ser obtida através da definição de uma taxa de recompensa (*reward rate*) de valor unitário na posição slv_x_dl , e do cálculo do valor da recompensa esperada (instantânea) em regime estacionário. Se se estiver interessado no conjunto de todas as mensagens, basta definir a medida como a soma de todas estas taxas de recompensa.

Refira-se que os modelos propostos, pela forma como foram desenvolvidos, podem ser utilizados para obter um número grande de outras medidas, nomeadamente as associadas à capacidade de desempenho (*performability*). Exemplos destas medidas são:

- ♦ Frequência de transmissão de *tokens* com erros não detectados: associar um impulso de recompensa unitário (*impulse reward*) à segunda escolha da transição t_{TT} , e calcular o valor esperado (instantâneo) da recompensa obtida em regime estacionário;
- ♦ Probabilidade de uma estação se encontrar em *offline*: associar uma taxa de recompensa unitária à posição *offline*, e calcular o valor esperado (instantâneo) da recompensa obtida em regime estacionário.

5.5.6 Modelo de Avaliação Completo

O modelo de avaliação completo da rede de campo é obtido através da composição dos modelos individuais propostos (Fig. 5.18). Cada mestre da rede é representado por um modelo individual, o mesmo acontecendo com os escravos. Existe apenas um modelo global que permite a interacção entre todos. A composição do modelo foi realizada através da primitiva *Join* da ferramenta.

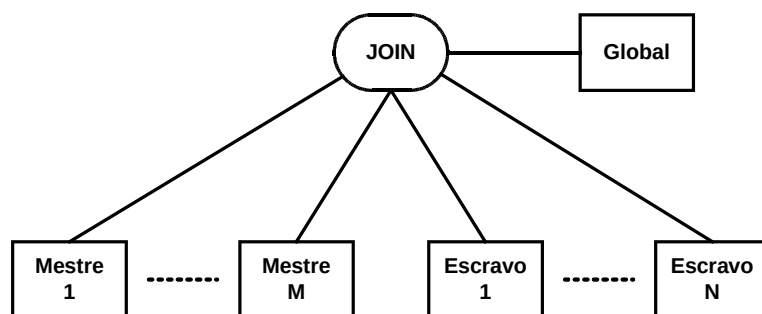


Figura 5.18 - Composição do modelo.

As ligações entre os modelos são realizadas por intermédio das posições globais. Posições com o mesmo nome em modelos diferentes representam uma posição única. Exceptuam-se os casos das posições *BUS_DW* e *BUS_UP* do modelo dos mestre. Neste caso a posição *BUS_UP* do modelo coincide com a posição *BUS_DW* do modelo seguinte (em termos de posição do anel) e assim sucessivamente, implementando desta forma um anel lógico.

Refira-se como nota final, que os modelos propostos podem também ser utilizados sem modificações para analisar uma rede mono-mestre.

5.6 Caso de Estudo

Com o objectivo de validar o modelo proposto foi definido um caso de estudo.

O trabalho discutido neste capítulo focou-se no desenvolvimento de modelos que permitam realizar a avaliação da confiança no funcionamento de uma rede Profibus-DP num cenário de falhas transitórias que ocorrem durante as comunicações. Nunca foi um objectivo deste trabalho realizar qualquer estudo sobre o comportamento específico da rede, mas sim disponibilizar modelos que permitissem realizar de forma eficiente uma avaliação numa gama alargada de cenários. Pretende-se assim fornecer os meios necessários que permitam realizar o tipo de avaliação que se julgue mais adequada a cada situação. Neste sentido, o caso de estudo proposto foca-se exclusivamente na análise das características do modelo, e nomeadamente nas características do seu desempenho.

Nas próximas secções pretende-se apresentar a forma como os parâmetros foram definidos e também discutir os vários aspectos associados à obtenção da solução do modelo.

5.6.1 Definição dos Valores dos Parâmetros

Não sendo este caso de estudo focado num sistema em particular, optou-se por analisar uma rede cujas características fossem próximas das condições típicas de aplicação das mesmas.

Neste sentido, definiu-se uma rede de campo composta por 3 mestres, cada um associado a 3 escravos. Os endereços dos mestres são: 1, 5 e 9. Os escravos associados, ocupam os endereços intermédios: 1=[2,3,4]; 5=[6,7,8]; 9=[10,11,12] (Tab. 5.3).

Estrutura da Rede de Campo	
Mestres	3
Escravos	9
HSA	13

Tabela 5.3 - Estrutura da rede de campo.

Os parâmetros associados à transmissão das mensagens estão definidos na Tabela 5.4. Para aqueles que dependem da velocidade de transmissão utilizou-se como referência 500Kbits/s. Cada trama transmitida tem um parte fixa e uma parte variável. Esta última representa os dados associados ao controlo do processo que são transmitidos entre os mestres e os escravos. Como estes valores são específicos

do sistema que se pretende avaliar e tendo em conta os objectivos definidos anteriormente utilizaram-se valores de referência [EN96b]. O factor de GAP (G) foi definido como 6, e foi assumido que o tempo de re-inicialização do mestre é nulo, ficando assim independente de factores externos. Por uma questão de simplicidade de análise, considerou-se que os parâmetros das transmissões são iguais para todos os mestres.

Parâmetros das Transmissões (em caracteres)		
Descrição	Parte Fixa	Parte Variável
<i>Token</i>	3	
Tempo de espera (T_{ID1})	3	
Tempo de espera (T_{ID2})	9	
Slot Time (T_{SL})	18	
Atraso da estação (T_{SDR})	9	
Pedido FDL	6	
Resposta FDL	6	
Pedido Dados	9	16
Resposta Dados	9	16
Pedido Diagnóstico	11	6
Resposta Diagnóstico	11	6
Pedido Parâmetros	11	6
Resposta Parâmetros	11	6
Pedido Configuração	11	6
Resposta Configuração	11	
Pedido Controlo Global	11	2

Tabela 5.4 - Parâmetros associados à transmissão das mensagens.

Em relação aos parâmetros relacionados com o tempo de ciclo do Profibus, nomeadamente o T_{TR} (*Target Rotation Time*), era possível utilizar 2 abordagens. A primeira, próxima de [TV99a], consiste em a partir das *deadlines* obter o T_{TR} mínimo que as satisfaz. A segunda, mais relacionada com a forma típica de utilizar o Profibus, consiste em definir um tempo de ciclo mínimo a partir do qual se garante à partida que as *deadlines* são cumpridas [Vit04]. Dada a simplicidade da sua utilização optou-se por esta última.

Desta forma e assumindo o seguinte cenário: quando um mestre recebe o *token* transmite 1 mensagem para cada escravo associado, com eventuais retransmissões, 1 mensagem de controlo e uma 1 mensagem de FDL, o tempo de ciclo máximo pode ser estimado como

$$T_{ciclo} = M \cdot (T_{Token} + T_{ID1}) + M \cdot E \cdot (T_{Dados} + T_{SDR} + T_{ID1}) \cdot R + M \cdot (T_{Controlo} + T_{ID2}) + M \cdot (T_{FDL} + T_{SDR} + T_{ID1}) \quad (5.11)$$

em que T_{Token} , T_{Dados} , $T_{Controlo}$ e T_{FDL} são os tempos necessários para transmitir respectivamente o *token* e as mensagens de dados (pedido e resposta), comandos globais e mensagens *FDL* (pedido e resposta). O número de mestres e de escravos é dado respectivamente por M e E , e o número de retransmissões por R . Note-se que esta expressão é muito pessimista pois assume que de cada vez que um mestre recebe o *token* transmite todo tipo de mensagens possíveis. O T_{TR} deve ser escolhido como

$$T_{TR} \geq T_{ciclo} \quad (5.12)$$

Com isto garante-se que cada mestre tem tempo tempo suficiente para transmitir as mensagens que necessita. Neste caso considerou-se que ambos os parâmetros são iguais. Em relação ao T_{MSI} (*Min Slave Interval*) optou-se por considerar que é igual a $\frac{1}{4}$ de T_{TR} . Com isto obtêm-se, em média, 4 pedidos de troca de dados num intervalo de tempo igual a T_{TR} .

As *deadlines* referentes às mensagens de dados, foram consideradas iguais em todos os mestres e com um valor igual a $2 \cdot T_{TR}$. Obtêm-se assim uma margem de manobra razoável (é necessário, em média, ter transmissões com erros em dois períodos consecutivos e iguais T_{TR} , que por sua vez já está sobre-dimensionado). Quanto ao T_{CI} (*Control Interval*), considerou-se igual á *deadline*, reflectindo assim a ideia que os dados são actualizados com um período igual à produção de mensagens. Para o cenário em estudo, os valores dos parâmetros anteriores estão definidos na Tabela 5.5.

Parâmetros de Operação da Rede (caracteres)	
T_{TR}	1281
T_{MSI}	320
T_{CI}	2562
<i>Deadline</i>	2562

Tabela 5.5 - Parâmetros associados à operação da rede.

Para terminar, saliente-se de novo que os valores indicados previamente têm apenas como objectivo definir um cenário minimamente consistente para avaliar o desempenho do modelo, e não têm (nem pretendem) como objectivo definir a forma como os parâmetros da rede devem ser calculados.

5.6.2 Caracterização do Desempenho do Modelo

Quando se dispõe de um modelo de simulação a primeira questão que se coloca está obviamente relacionada com os aspectos directamente associados ao seu desempenho. Um modelo pode estar estruturalmente correcto e no entanto necessitar de recursos computacionais de tal forma exigentes, ou de um tempo de execução de tal forma elevado, que a sua utilidade se torna diminuta. É assim necessário caracterizar o desempenho do modelo na perspectiva da simulação.

O desempenho é tipicamente definido através do tempo de simulação necessário para obter os resultados (medidas) dentro de um determinado contexto. Tentar caracterizar o desempenho do modelo proposto é naturalmente uma tarefa complexa, dado que existem muitos factores envolvidos dos quais se destacam:

1. Plataforma computacional

As características, ao nível do *hardware*, da plataforma computacional utilizada para executar as simulações vai influenciar fortemente o tempo necessário para obter os resultados. Neste contexto salientam-se os seguintes factores:

- Características do CPU (arquitectura: 32, 64 bits; velocidade; etc.);
- Memória (capacidade; velocidade de acesso; etc.);
- Ambiente de simulação centralizado ou distribuído.

2. Características do simulador

Este aspecto pode ser analisado segundo duas perspectivas. Por um lado as características associadas ao funcionamento interno do simulador, nomeadamente à forma como são geridas as listas de eventos. Por outro lado, a incorporação no mesmo de técnicas de redução do tempo de simulação (ex. minimização da variância, amostragem de importância, simulação distribuída);

3. Escalonamento das mensagens

É fácil de verificar que num cenário onde os tempos de resposta da mensagens são muito próximos das respectivas *deadlines*, a probabilidade de uma *deadline* ser violada é bastante superior em relação à situação inversa, implicando desta forma tempos de simulação mais curtos;

4. Taxa de erro do bit (BER)

O valor deste parâmetro influencia de forma directa a probabilidade de ocorrer uma violação da *deadline*, dado que representa de forma indirecta a

taxa de falhas, implicando assim um maior, ou menor, número de erros durante as comunicações;

5. Precisão dos resultados

O grau de precisão dos resultados obtidos com simulação decorrem das características dos intervalos de confiança utilizados. De acordo com a discussão realizada em (§2.8) as características destes intervalos, nomeadamente o seu nível de confiança e largura, têm uma influência determinante no tempo de simulação.

Face a um tão grande número de aspectos é fácil de perceber que é praticamente impossível definir uma métrica para caracterizar o desempenho do modelo que seja verdadeiramente independente dos factores acima mencionados.

Por forma a tentar resolver este problema, procurou-se seguir uma abordagem pragmática para a definição de uma métrica de desempenho. Assim, é verificar que os factores 1), 3), 4) e 5) representam aqueles que possuem maior influência no tempo de simulação. Dado que 3) e 4) são inerentes ao próprio cenário avaliado, se for escolhido um cenário típico é de esperar que a métrica reflecta também este aspecto. A mesma perspectiva pode ser utilizada em relação ao factor 1), utilizando uma plataforma com características típicas. Em relação ao factor 5) é necessário que a métrica seja independente dos aspectos estatísticos da simulação. Em situações típicas este é certamente o factor mais importante, dado que define o número de amostras necessárias para obter os resultados (medidas) que por sua vez estão fortemente ligadas ao tempo de simulação.

Assim e de acordo com a abordagem anterior, a métrica foi definida como o tempo de simulação necessário para realizar uma única execução do modelo (uma realização amostral), durante um intervalo de tempo simulado fixo⁹³. A métrica assim definida reflecte directamente o desempenho do modelo, e não depende das características do processo estatístico utilizado para obter os resultados, ou seja, do número de simulações necessárias para obter as medidas dentro do intervalo de confiança pretendido.

Na continuação desta abordagem, também não foram utilizadas as funcionalidades do ambiente de simulação distribuído disponibilizado pela ferramenta. Todas as simulações foram realizadas numa única máquina, executando apenas um único processo de simulação.

⁹³ Isto é equivalente a uma simulação terminal, mas em que se garante que o modelo é apenas executado uma única vez. Estes aspectos podem ser parametrizados na ferramenta.

5.6.3 Resultados Obtidos

Nesta secção são apresentados os resultados obtidos nas simulações para a rede proposta, utilizando uma velocidade de transmissão de referência de 500Kbits/s. As experiências foram realizadas num PC equipado com um processador Pentium III @ 730Mhz, 256KB cache, 512MB de RAM, e com um sistema operativo Linux (Mandrake 10.0) kernel 2.6.3.

A Tabela 5.6 apresenta os resultados obtidos de acordo com a métrica definida anteriormente para avaliar o desempenho do modelo. Neste caso, são indicados os tempos de execução reais necessários para executar 1 simulação (uma realização amostral) correspondendo a um intervalo de tempo simulado fixo. Foi utilizada uma simulação terminal, cuja duração corresponde ao intervalo definido.

Cenário	Tempo Real de Simulação - 1 simulação	
	Intervalo simulado – 1s	Intervalo simulado – 10s
BER=0	0.53 segundos	3.6 segundos

Tabela 5.6 - Tempo real necessário para executar 1 simulação.

A partir da análise destes resultados, verifica-se que a o ganho em relação ao tempo simulado varia entre 2 a 3 vezes. Refira-se que os tempos apresentados não são tempos de execução no CPU do processo de simulação mas sim tempos reais, obtidos com todo o conjunto de processos típicos a serem executados em simultâneo pelo sistema operativo. A utilização de um cenário em que $BER=0$, resulta do facto desta situação corresponder à maior “carga” sobre simulador. De facto, a existência de falhas durante as comunicações vai originar cenários em que os intervalos entre os eventos são maiores (em média). Logo o número de eventos processados por unidade de tempo é menor.

Face à forma como estes resultados foram obtidos, estes devem ser interpretados como valores conservativos, devido ao seguinte conjunto de aspectos:

1. A plataforma computacional utilizada possui, na perspectiva dos padrões actuais, um poder de cálculo que pode ser considerado médio/baixo em termos comparativos, nomeadamente ao nível do CPU. Por forma a comprovar este aspecto, foram realizadas experiências com uma outra máquina equipada com um CPU Pentium IV @ 1800Mhz (com as restantes características semelhantes) e com sistema operativo Windows XP. Verificou-se neste caso que o tempo de simulação se reduzia para menos de metade⁹⁴;

⁹⁴ Estes resultados não foram incluídos pois nestas experiências foi difícil determinar com rigor necessário o tempo real de simulação, bem como controlar os vários processos que eram executados em paralelo pelo sistema operativo.

2. A rede de campo utilizada neste caso de estudo tem uma dimensão apreciável (12 nós), e possui um número de mestres bastante razoável (em termos de aplicações práticas é mais comum encontrar redes com muitos escravos e muito poucos mestres, tipicamente são mono-mestres). Quando se compara o modelo do mestre com o do escravo, verifica-se que o primeiro possui uma complexidade muitíssimo superior. Desta forma, quer o número, quer a complexidade destes modelos vão contribuir para um pior desempenho global;
3. O cenário definido neste caso ($BER=0$), corresponde a uma maior exigência computacional.

A avaliação da qualidade do modelo proposto em relação ao seu desempenho, pode ser definida através da relação entre o tempo simulado e o tempo real de simulação. Quanto maior for esta relação, melhor será o desempenho do modelo. Embora neste caso a relação não seja muito elevada, tendo em conta os vários aspectos atrás discutidos referentes à forma como os resultados foram obtidos, é possível dizer que esta relação é suficientemente boa (e tem potencialidades para ser ainda melhor) para que se possa concluir que o modelo possui de facto um bom desempenho.

Por forma a comprovar o desempenho efectivo do modelo, realizou-se uma simulação completa com o objectivo de determinar as medidas de interesse (Tab. 5.7). A medida definida incidia sobre a probabilidade de violação da *deadline* de 1 mensagem específica, utilizando-se para o BER os valores de referência definidos em [Wil02]. Os resultados apresentados indicam o tempo real de simulação que foi necessário para obter a medida de interesse, no ambiente computacional definido previamente.

De acordo com a medida de interesse definida, todas as simulações foram obtidas em regime estacionário utilizando a técnica da média dos lotes (*batch means*). A utilização desta técnica no Möbius deve ser realizada com algum cuidado, dado que a dimensão dos lotes é indicada manualmente. Desta forma, foi necessário definir previamente os seguintes parâmetros:

♦ Duração do período transitório do processo estocástico

Durante esta fase o processo estocástico ainda não chegou ao seu regime estacionário. Desta forma, se as amostras recolhidas fossem consideradas para o cálculo das medidas, a sua contribuição pode tornar o respectivo estimador enviesado. A técnica adoptada para resolver este problema passa por ignorar as amostras recolhidas durante este período. O problema coloca-se agora na determinação da duração desta fase.

A abordagem utilizada neste trabalho foi dupla. Por um lado, utilizou-se a técnica descrita em [Tys99], e por outro realizaram-se várias simulações

terminais (*terminating simulations*) a partir das quais se observou experimentalmente a evolução do processo. Utilizando ambas as informações foi possível estabelecer (aproximadamente) a duração do período transitório;

◆ Dimensão dos lotes

Este parâmetro está intimamente relacionado com a independência das amostras obtidas em cada lote. Se por um lado se pretende que a dimensão dos lotes aumente para diminuir a correlação (covariância normalizada) entre os mesmos, por outro pretende-se o menor número de simulações. Este é um problema de difícil resolução (a menos que se tenha acesso à função correlação), que normalmente é resolvido recorrendo a heurísticas. A abordagem seguida neste trabalho foi escolher um intervalo de tempo que fosse pelo menos 2 ordens de grandeza superior à taxa de falhas considerada.

No sentido da discussão anterior, a duração dos lotes e do período transitório foram definidos como 10^3 e 10^5 caracteres respectivamente, correspondendo a 2.2ms e 2.2 segundos respectivamente para uma velocidade de transmissão de 500Kbits/s.

Cenário	Tempo Real de Simulação – Simulação Completa Intervalo de confiança: 95% - Máximo erro relativo: 10%
$BER=10^{-3}$	37 segundos
$BER=10^{-4}$	2268 segundos

Tabela 5.7 - Tempo real de uma simulação completa.

O resultado obtido para $BER=10^{-3}$ parece bastante surpreendente face à complexidade do modelo. Contudo, uma análise mais pormenorizada demonstra os motivos para este bom desempenho. Tomemos como exemplo a trama referente a um pedido de mensagem de dados (25 caracteres) e calcule-se a probabilidade da sua transmissão decorrer com erros

$$Prob_{\text{Mensagem com Erros}} = 1 - (1 - 10^{-3})^{(25 \cdot 11)} = 0.241 \quad (5.13)$$

Face a um valor tão elevado para a probabilidade de insucesso, não surpreende assim que o modelo apresente um desempenho tão bom. Refira-se que este BER corresponde a uma taxa de falhas ≈ 500 falhas/s, o que representa a um ambiente com uma elevada interferência electromagnética.

O segundo cenário corresponde a uma taxa de falhas de ≈ 50 falhas/s, o que na prática é um valor relativamente baixo. Desta forma, os resultados devem ser interpretados de forma positiva. Numa primeira abordagem poder-se-ia julgar que tendo a probabilidade de erro diminuído cerca de 10 vezes (a probabilidade de erro para a mensagem acima indicada é neste caso de 0.027), que o tempo de simulação

aumentaria na mesma proporção. No entanto este último aumentou cerca de 60 vezes. Este comportamento pode justificar-se da seguinte forma. A largura do intervalo de confiança é $\approx 1/\sqrt{n}$, em que n é o número de amostras. Se se admitir que a medida realizada toma valores ≈ 10 vezes menores (mera suposição baseada no comportamento do BER), e como a largura relativa do intervalo de confiança se manteve, isto implica que o número de amostras terá que aumentar ≈ 100 vezes. Como o número de amostras é um indicativo da duração da simulação, justifica-se assim este comportamento.

Os resultados obtidos a partir da simulação completa permitem caracteriza-los de forma muito positiva, em função do seguinte conjunto de aspectos:

1. Os valores escolhidos para os vários parâmetros do modelo não foram particularmente favoráveis. Basta lembrar que as *deadlines* foram definidas em função de um tempo de ciclo exageradamente sobre-dimensionado, e que as mensagens transmitidas são todas muito curtas, dando assim origem a uma baixa probabilidade de ocorrência de erros;
2. O cenário $BER=10^{-4}$ corresponde na prática a um baixo nível de ocorrência de falhas, quando comparado com os restantes aspectos temporais do modelo. Neste caso o tempo de ciclo é de cerca 28ms ($\approx 35\text{Hz}$), o que dá em média 2 falhas por cada 3 ciclos. Tendo em conta a relação existente entre o tempo de ciclo e as *deadlines*, é fácil de concluir que as perturbações introduzidas na rede de campo vão ser muito reduzidas. Aliás, só tem sentido realizar uma avaliação desta natureza quando se consideram taxas de falhas que impliquem um certo nível de perturbações na rede que tenham sentido ser medidas. Tendo em conta que o cenários escolhidos para o BER representam casos limites e opostos, é de admitir que os valores intermédios produzirão melhores resultados;
3. O intervalo de confiança foi definido com um nível de confiança elevado (0.95), e com uma largura relativa muito reduzida (10%), o que implica que o número de simulações realizadas seja comparativamente maior. No segundo cenário para o BER a situação torna-se ainda mais exigente porque as medidas de interesse tomam valores muito baixos, o que implica que a largura efectiva do intervalo seja muito reduzida, resultando assim num maior número de simulações;
4. Não foram utilizadas as potencialidades da ferramenta referentes à utilização de um ambiente de simulação distribuído. É de esperar que num contexto desta natureza os resultados obtidos referentes ao tempo de simulação possam sofrer melhorias dramáticas.

No próximo capítulo será descrita uma técnica que permite utilizar também este modelo (sem quaisquer modificações) em cenários onde as falhas ocorrem em *bursts*. A técnica passa por definir dois intervalos de tempo, correspondentes à ocorrência e não-ocorrência de falhas, simular apenas o comportamento referente à de ocorrência de falhas e posteriormente obter a solução completa através da utilização de técnicas probabilísticas.

Um dos objectivos propostos para este capítulo passava também pela utilização da técnica da decomposição comportamental para incluir os resultados obtidos em modelos hierarquicamente superiores. Desta forma e no contexto dos modelos desenvolvidos no capítulo 4, estes resultados seriam incluídos no modelo **BUS**. Assim, a partir do *BER* pode ser obtida a taxa de ocorrência das falhas transitórias que afectam as comunicações. Através desta taxa, e conhecendo o respectivo factor de cobertura ao nível do *hardware*, é possível determinar a taxa real com que estas falhas ocorrem. Estes dois parâmetros permitem definir a componente **HW** do modelo **BUS** referente às falhas transitórias. A componente **SW** deste mesmo modelo, mais concretamente o factor de cobertura ao nível do *software*, é definido pela probabilidade, em regime estacionário, de uma *deadline* não ser violada. Este parâmetro é obtido directamente do modelo proposto considerando o conjunto de todas as mensagens.

5.7 Conclusões

Neste capítulo é proposto um modelo para avaliar a confiança no funcionamento de uma rede Profibus-DP em cenários de falhas transitórias que ocorrem durante as comunicações.

O modelo desenvolvido representa o comportamento da *stack* de comunicações completa do Profibus-DP, nomeadamente os aspectos específicos da camada de aplicação. Isto inclui para além das mensagens de dados trocadas entre mestres e escravos, as fases de configuração, parametrização e verificação dos escravos.

O modelo é extremamente rigoroso e detalhado na definição do comportamento da rede em cenários de falhas, incluindo vários aspectos importantes que anteriores abordagens não contemplaram. É assim possível concluir que se está perante um modelo que representa fielmente o comportamento de uma rede Profibus-DP neste tipo de cenários.

Embora a solução do modelo seja obtida através de simulação, os resultados obtidos a partir do caso de estudo realizado permitem concluir que o modelo

possui um bom desempenho a este nível. O contexto conservador em que os resultados foram obtidos, permite antever que a utilização de um ambiente de simulação distribuído pode melhorar de forma dramática os tempos de simulação.

É assim possível concluir que o modelo desenvolvido é rigoroso na abordagem e perfeitamente exequível ao nível da solução, utilizando recursos computacionais acessíveis e tempos de execução perfeitamente razoáveis em função do cenário de falhas definido.

Face a este conjunto de aspectos, é possível concluir que o modelo desenvolvido cumpre com os objectivos propostos inicialmente.

Esta página foi intencionalmente deixada em branco

Capítulo 6

Avaliação da Rede CAN em Cenários de Falhas Transitórias

6.1 Introdução

Entre as várias redes de campo existentes, o CAN representa actualmente a solução com o maior número de aplicações ao nível dos sistemas de segurança crítica (*safety-critical systems*). Isto deve-se fundamentalmente às características do seu funcionamento especialmente adequadas aos sistemas de tempo-real, bem como às excelentes propriedades ao nível da tolerância a falhas, nomeadamente no que diz respeito à integridade dos dados transmitidos.

Tal como discutido no capítulo anterior, a avaliação do comportamento de uma rede de campo em cenários de falhas transitórias que ocorrem durante as comunicações justifica uma análise detalhada pela relevância que assume. A crescente utilização do CAN em sistemas de segurança crítica, têm vindo a motivar a comunidade científica para desenvolvimento de trabalhos de investigação que se focam precisamente neste contexto. No entanto e ao nível do desenvolvimento de

modelos de avaliação, são relativamente poucos os trabalhos que assumem a existência de comportamentos de natureza estocástica, nomeadamente no que diz respeito à ocorrência das falhas e às respectivas consequências ao nível do comportamento da rede. Mesmos os poucos que assumem estes aspectos, acabam por definir cenários ou demasiado simplistas, ou muito pessimistas, essencialmente com o objectivo de garantir à partida uma solução eficiente para o modelo.

No sentido de ultrapassar as limitações destes trabalhos, é proposto neste capítulo um modelo de avaliação, no contexto definido previamente, que procura descrever com maior rigor as condições reais de funcionamento da rede, sem introduzir pressupostos simplistas ou pessimistas, e mantendo a respectiva eficácia ao nível da solução, permitindo assim obter medidas mais consistentes com a realidade. Para além deste objectivo principal, o modelo proposto também se enquadra na abordagem definida no capítulo 4, nomeadamente através da utilização da decomposição comportamental para incorporar os resultados obtidos.

Este capítulo representa uma versão extendida dos trabalhos publicados pelo autor em [PA04a] e [PA04d].

6.2 O CAN

O CAN (*Controller Area Network*) [CAN91] [IS099b] foi desenvolvido nos finais dos anos 80 pela Bosch com a intenção de dotar os veículos automóveis de uma rede de comunicações capaz de satisfazer os requisitos das respectivas aplicações de controlo [CV04]. A *stack* de comunicações encontra-se estruturada de acordo com o modelo OSI [ISO94], mas utilizando unicamente duas camadas: física e de ligação de dados (Fig. 6.1).

6.2.1 Camada Física

A camada física define uma topologia em *bus*, suportando vários meios físicos com diferentes características:

- ♦ Tolerante a falhas [ISO04a];
- ♦ Condutor único (*single wire*) [SAE00];
- ♦ Ponto-a-ponto [ISO03b];
- ♦ Fibra óptica;
- ♦ Alta-velocidade [ISO03a].

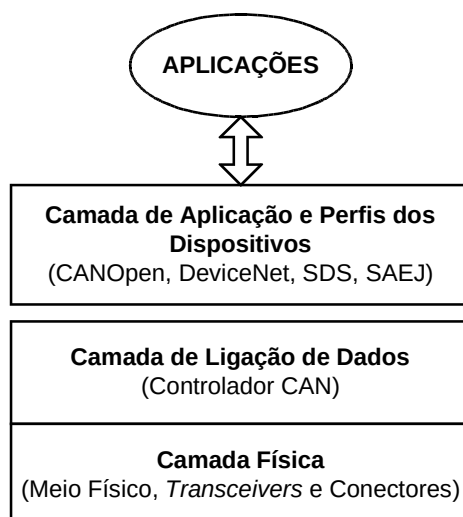


Figura 6.1 - Arquitectura de comunicações do CAN.

O meio físico de alta-velocidade é tipicamente o mais utilizado. Neste último caso, os dados são transmitidos num par torcido sob a forma de um sinal diferencial, garantindo assim um elevado nível de imunidade ao ruído. É utilizada uma codificação NRZ (*Non Return to Zero*), o que permite obter uma baixa densidade espectral, com a garantia de uma boa utilização da largura de banda disponível. A transmissão é realizada de forma síncrona, utilizando um sofisticado mecanismo de re-sincronização aplicado bit-a-bit. Este último aspecto é ainda reforçado pela utilização de uma técnica de *bit-stuffing*, com o objectivo de aumentar o número de transições no sinal transmitido. De acordo com esta técnica, quando 5 bits consecutivos com o mesmo valor surgem na trama que vai ser transmitida, o nó transmissor insere um bit adicional (*bit-stuff*) de valor complementar. Os nós receptores realizam a operação inversa.

Por forma a garantir que o meio físico possui as propriedades eléctricas adequadas, as ligações dos nós ao *bus* (*stubs*) devem ser o mais curtas possíveis. Além disso, o meio físico deve ser adaptado em ambas as extremidades por resistências de terminação com o objectivo de suprimir eventuais reflexões do sinal transmitido.

A velocidade de transmissão é função do comprimento da rede. Tipicamente, esta pode variar desde 50Kbits/s @1000m até 1Mbits/s @40m. Esta limitação não se deve primariamente, e como seria de esperar, a eventuais limitações no funcionamento dos *transceivers* dos nós ou às características do meio físico, mas sim à forma *quasi-estacionária* como as sinais são transmitidos: para cada bit transmitido é garantido o tempo suficiente para que o sinal estabilize no *bus* antes de ser amostrado, de forma quase simultânea, por todos os nós da rede.

6.2.2 Camada de Ligação de Dados

A camada de ligação de dados implementa um protocolo do tipo CSMA/CR (*Carrier Sense Multiple Access with Deterministic Collision Resolution*), garantindo assim uma arquitectura de comunicações do tipo multi-mestre (*multi-master*), com um mecanismo de acesso ao meio físico baseado em prioridades. A troca de dados entre nós realiza-se através da transmissão de tramas (Fig. 6.2). As tramas podem ser de quatro tipos:

- ♦ **Dados.** Utilizadas para a transmissão de dados entre nós;
- ♦ **Remotas.** Utilizadas para requisitar a transmissão de dados. Em termos estruturais são semelhantes às tramas de dados;
- ♦ **Erro.** Utilizadas para sinalizar a ocorrência de erros;
- ♦ **Sobrecarga (*Overload*).** Utilizadas para estender o intervalo de tempo entre tramas quando surgem situações de sobrecarga na transmissão.

A discussão realizada nesta secção vai focar-se essencialmente sobre as tramas de dados e de erros, dado que são as mais importantes e as mais comuns de encontrar nas aplicações reais.

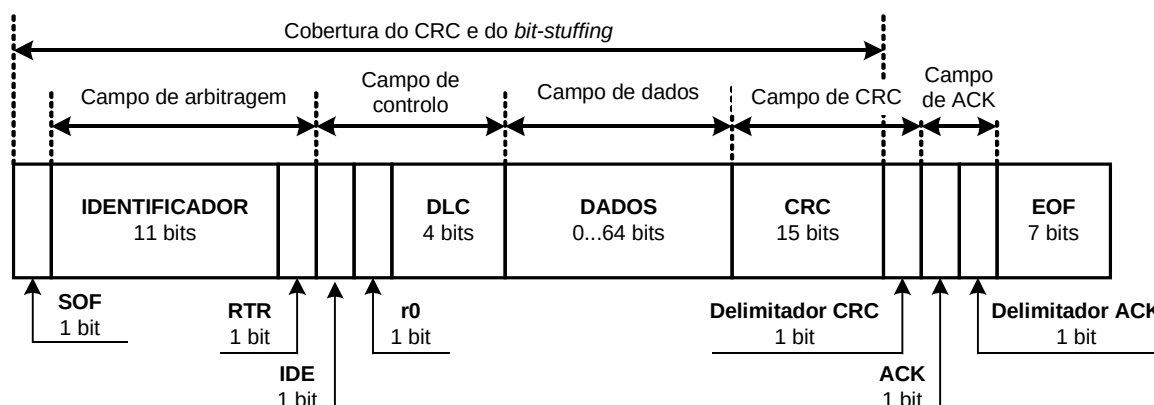


Figura 6.2 - CAN 2.0A: formato de uma trama de dados.

O mecanismo de acesso ao meio é simultaneamente simples e sofisticado, sendo suportando directamente através da própria estrutura da trama de dados transmitida, nomeadamente através do campo identificador.

Cada nó está interligado ao meio físico de transmissão através de uma saída em colector aberto (*open collector*). Assim, o meio físico funciona como um AND lógico dos sinais transmitidos por todos os nós da rede. Desta forma, cada bit transmitido no meio físico assume uma das seguintes representações: *recessivo* (1 lógico), quando todos os nós transmitem bits recessivos, e *dominante* (0 lógico), quando pelo menos um nó transmite um bit dominante.

O processo de transmissão de uma trama de dados⁹⁵ é o seguinte (Fig. 6.3):

1. Os nós interessados em transmitir aguardam que o meio físico esteja livre (*idle* – 1 lógico). Ou seja, que não existam outros nós presentemente a transmitir;
2. Verificada a condição anterior, os nós interessados podem começar a transmitir. Devido às técnicas de sincronização utilizadas, a transmissão das tramas vai realizar-se de forma sincronizada (SOF – *Start of Frame*, Fig. 6.2) e simultânea pelos vários nós;
3. Enquanto transmite cada bit cada nó monitoriza o estado do meio físico;
4. Se durante o campo de arbitragem um nó transmitir um bit recessivo e monitorizar um dominante, então aborta a transmissão e passa a receber os dados (trama) que estão presentemente a ser transmitidos;
5. No final do campo de arbitragem o nó com o menor identificador (supondo o identificador representado como um número inteiro sem sinal cujo bit menos significativo se situa o mais à direita) garante o acesso incondicional ao meio de transmissão e continua a transmitir o restante da trama;
6. Para os nós que “perderam” o processo de arbitragem, as respectivas tramas que não foram transmitidas aguardam por uma futura retransmissão. Este mecanismo está implementado de forma automática;
7. Após a transmissão de uma trama, a transmissão da seguinte só pode iniciar-se após ter decorrido um tempo mínimo, denominado *intermissão* (3 bits recessivos).

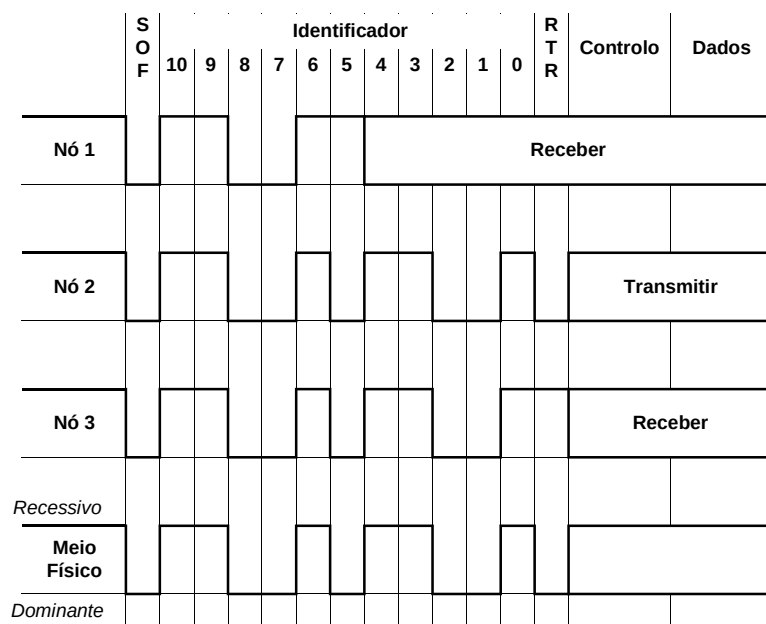


Figura 6.3 - Mecanismo de acesso ao meio.

⁹⁵ Estas tramas são normalmente designadas por mensagens.

O processo de arbitragem é assim não-destrutivo, dado que a transmissão da trama de dados com o menor identificador se realiza sem qualquer interrupção. O identificador têm assim um duplo objectivo. Por um lado identifica a trama transmitida, e por outro indica a prioridade da mesma, definindo assim um mecanismo de escalonamento ao nível das próprias tramas de dados. O CAN suporta dois tipos de identificadores: standard, constituídos por 11 bits (CAN 2.0A) e extendidos, constituídos por 29 bits (CAN 2.0B). Uma descrição detalhada dos vários tipos de tramas e do respectivos campos pode ser encontrada em [CAN91], [Ruf95] e [ISO99].

Durante a transmissão de uma trama de dados todos os nós monitorizam o meio de transmissão tentando detectar eventuais situações de erro [CAN91] [ISO04b]. Estas situações são de vários tipos, afectando quer a estrutura, quer o conteúdo das tramas. Em relação a este último caso, saliente-se que as tramas de dados do CAN incluem um campo (CRC, Fig. 6.2) destinado a assegurar a integridade dos dados transmitidos. Este campo resulta da aplicação, pelo nó emissor, de um código polinomial de 15 bits (CRC – *Cyclic Redundancy Code*) ao dados transmitidos. Após receberem a trama, os nós receptores realizam uma operação semelhante e comparam o valor obtido com o campo CRC da trama. Caso existam divergências é porque ocorreram erros durante a transmissão, implicando assim a rejeição (não aceitação) da trama. Segundo [CAN91] este código, conjuntamente com os restantes mecanismos, possui as seguintes propriedades:

- ♦ Todos os erros globais são detectados;
- ♦ Todos os erros locais aos emissores são detectados;
- ♦ Até 5 bits errados distribuídos aleatoriamente nas tramas são detectados;
- ♦ *Bursts*⁹⁶ (sequências) de erros com comprimento inferior a 15 bits são detectados;
- ♦ Qualquer número de erros ímpar é detectado;
- ♦ A probabilidade de uma trama conter erros e estes não serem detectados é de $\approx 4.7 \times 10^{-11}$.

As situações de erro são sinalizadas, pelo nó que as detectou, através da transmissão de uma trama de erro. O início da transmissão da trama de erro encontra-se bem definido [CAN91], sendo função do campo da trama de dados onde o erro ocorre. Na maioria das situações a transmissão começa no bit imediatamente seguinte ao do erro detectado, obtendo-se assim baixas latências ao nível da recuperação de erros.

⁹⁶ Utilizou-se a terminologia em Inglês porque se julgou mais adequada.

As tramas de erro são constituídas por dois campos: a *flag* de erro e o delimitador de erro (Fig. 6.4). As *flags* de erros podem ser de dois tipos: a *flag* de erro activa é composta por 6 bits dominantes, enquanto que a *flag* de erro passiva é composta por 6 bits recessivos. A transmissão da *flag* de erro activa viola as regras do *bit-stuff* ou da sintaxe da trama correntemente transmitida, reforçando assim as condições de erro. Como os nós podem não detectar o erro no mesmo instante, é possível que a transmissão das várias tramas de erro se sobreponham, originando assim uma *flag* de erro que pode durar entre 6 a 12 bits. Após a transmissão da *flag* de erro, cada nó transmite bits recessivos e ao mesmo tempo monitoriza o meio físico até um bit recessivo ser detectado. A partir deste instante envia mais 7 bits recessivos completando assim o delimitador de erro.

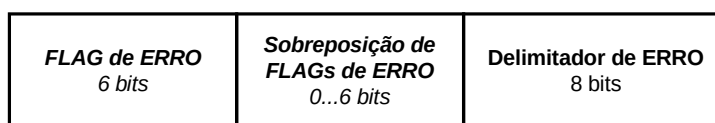


Figura 6.4 - Trama de erro.

Associado aos mecanismos de gestão de erros, cada nó mantém dois contadores referentes às condições de erros das tramas transmitidas (TEC – *Transmission Error Count*) e recebidas (REC – *Receive Error Count*). Estes contadores permitem ao nó implementar uma política de confinamento de erros. Quando ocorrem situações de erro os contadores são incrementados, sendo posteriormente decrementados quando estas situações deixam de ocorrer. As regras de actualização dos contadores são um pouco complexas e os detalhes podem ser encontrados em [CAN91] e [ISO04b]. Neste sentido, cada nó pode estar num dos seguintes estados:

- ♦ Erro-activo;
- ♦ Erro-passivo;
- ♦ Desligado (*bus off*).

Os nós que estão nos estados erro-activo e erro-passivo podem tomar parte nas transmissões de dados da mesma forma, embora no segundo caso o tempo de intermissão necessário aumente de 3 para 11 bits. Contudo, reagem às condições de erro de forma distinta, enviando *flags* de erro activas e passivas respectivamente. O objectivo é minimizar os efeitos sobre o funcionamento da rede provocados por um nó no estado erro-passivo (que está neste estado por ter experimentado várias condições de erro, podendo este efeito ser sinónimo de uma avaria do mesmo). A passagem para o estado erro-passivo acontece quando estes contadores atingem o valor de 127. Quando os contadores atingem o valor de 255, o nó é desligado logicamente da rede. A partir deste ponto o nó não pode transmitir mais tramas. O nó só retorna ao estado de erro-activo caso seja efectuado um *reset* ao mesmo, ou depois de 128 ocorrências de 11 bits recessivos consecutivos monitorizados no meio de físico.

Refiram-se por fim as características mais importantes das tramas de sobrecarga (*overload*). Estas tramas são transmitidas quando se verificam as seguintes situações:

- ♦ Por questões de processamento interno, o nó receptor necessita que a transmissão da próxima trama de dados (ou remota) seja atrasada;
- ♦ É detectado um bit dominante durante o tempo de intermissão.

Em termos estruturais estas tramas são semelhantes às tramas de erro, sendo constituídas por dois campos: a *flag* de sobrecarga e o delimitador de sobrecarga. Tal como nas tramas de erro podem surgir situações de sobreposição de várias *flags* de sobrecarga.

FLAG de SOBRECARGA 6 bits Dominantes	Sobreposição de FLAGS de SOBRECARGA 0...6 bits	Delimitador de SOBRECARGA 8 bits Recessivos

Figura 6.5 - Trama de sobrecarga.

Ao nível da camada de ligação de dados surgiram também algumas variantes do CAN, das quais se destacam: o TTCAN (*Time-Triggered CAN*) [FMD+00], [ISO04b] e o FTT-CAN (*Flexible Timmed-Triggered CAN*) [APF02], ambas baseadas num modelo de comunicações do tipo *time-triggered*, sem contudo alterar as características fundamentais do próprio CAN.

6.2.3 Camada de Aplicação

Embora a especificação do CAN não contemple uma camada de aplicação, a sua flexibilidade aliada à existência de uma grande variedade de microcontroladores que implementam directamente o protocolo [CV04] criaram as condições necessárias para o desenvolvimento de várias soluções a este nível. Estas camadas foram desenvolvidas segundo diferentes perspectivas e domínios de aplicação diversos, destacando-se presentemente as seguintes:

- ♦ **CanOpen** [CAN02] [EN02]

Desenvolvido pela CiA (*CAN in Automation*), originalmente a partir do CAL (CAL – *CAN Application Layer*), com o objectivo de disponibilizar uma solução no domínio das máquinas-ferramentas (embora actualmente o seu uso se tenha diversificado). A sua especificação é baseada num modelo orientado ao objecto, que descreve o comportamento dos dispositivos (nós) da rede. A informação é trocada entre dispositivos utilizando o conceito de objectos de comunicação (Fig. 6.6). Estes objectos podem ser de vários tipos: comunicação em tempo-real (PDO - *Process Data Objects*), dados de configuração (SDO - *Service Data Objects*), funções especiais (*Time Stamp*, *Sync message*, *Emergency message*, etc.) e gestão da rede (*Boot-up message*, *NMT message*,

Error Control, etc.). Embora a arquitectura adoptada assuma que cada nó da rede é um mestre, na prática é geralmente utilizado um esquema do tipo mestre-escravo para a comunicação entre dispositivos;

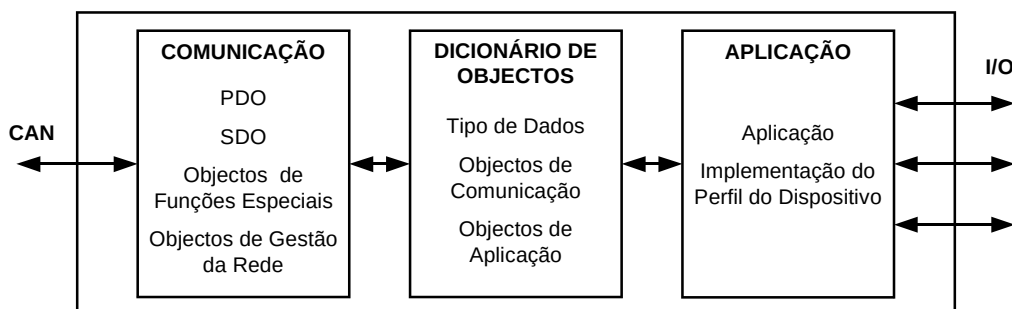


Figura 6.6 - CANOpen: modelo dos dispositivos.

♦ SDS (*Smart Distributed System*) [SDS99]

Desenvolvido pela Honeywell Micro Switch com o objectivo de disponibilizar uma solução para a interligação de sensores e actuadores inteligentes. A sua especificação é baseada num modelo orientado ao objecto muito simples, que descreve o comportamento dos dispositivos da rede. Essencialmente são especificados quatro tipo de serviços básicos, que implementam as seguintes funcionalidades: escrita, leitura, eventos e acções (Tab. 6.1);

Serviço	Função
<i>Read</i>	Ler o valor de um atributo num objecto de um dispositivo.
<i>Write</i>	Modificar o valor de um atributo num objecto de um dispositivo.
<i>Event</i>	Notificar a ocorrência de um evento num objecto de um dispositivo.
<i>Action</i>	Enviar um comando para um objecto executar uma determinada acção.
<i>COS ON</i>	Notificar uma mudança de estado de um dispositivo que está no estado ON.
<i>COS OFF</i>	Notificar uma mudança de estado de um dispositivo que está no estado OFF.
<i>Write ON</i>	Serviço especializado que escreve o estado ON de um dispositivo.
<i>Write OFF</i>	Serviço especializado que escreve o estado OFF de um dispositivo.
<i>Connection</i>	Permite abrir/fechar conexões lógicas individuais.
<i>Channel</i>	Permite estabelecer e utilizar canais <i>Multicast</i> e <i>Peer-to-Peer</i> .

Tabela 6.1 - SDS: serviços da camada de aplicação.

♦ DeviceNet [DNE00]

Desenvolvido pela Allen-Bradley como uma solução para domínios de aplicação associados às indústrias de manufactura. A sua especificação é baseada num modelo orientado ao objecto, que descreve quer o comportamento, quer as funções dos dispositivos da rede (Fig. 6.7). De forma

geral, os objectos podem classificar-se em três tipos: comunicação, sistema e específicos da aplicação.

Os objectos e os serviços são endereçados de acordo com um esquema hierárquico que utiliza o identificador das tramas transmitidas e a informação enviada no campo de dados. A comunicação entre nós (ponto-a-ponto ou *multicast*) desenrola-se de acordo com um esquema orientado à conexão. Basicamente existem dois tipos de comunicações: *mensagens explícitas*, para a troca de dados genéricos entre dispositivos (ex. configuração, gestão e diagnóstico), e *mensagens de I/O*, para troca de mensagens de tempo-real de acordo com o modelo produtor-consumidor.

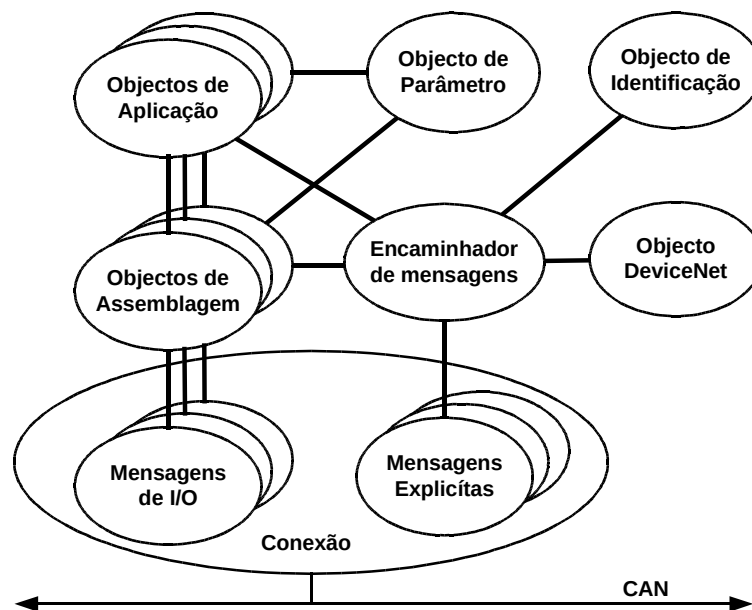


Figura 6.7 - DeviceNet: modelo de objectos.

6.3 Discussão dos Trabalhos Relevantes Previamente Desenvolvidos

No contexto do problema discutido neste capítulo, os trabalhos previamente desenvolvidos podem ser enquadrados em duas grandes linhas de investigação. A primeira, mais relacionada com o comportamento geral da rede CAN em cenários de falhas em geral, e que inclui também o desenvolvimento de arquitecturas tolerantes a falhas. A segunda, que embora incorpore muitos dos aspectos da anterior, centra-se mais nos sistemas de tempo-real e em particular no escalonamento de mensagens numa rede CAN em cenários de falhas durante as comunicações.

Nas secções seguintes proceder-se-à uma descrição cronológica dos trabalhos mais relevantes de acordo com a estrutura previamente definida.

6.3.1 Comportamento Geral em Cenários de Falhas

Em [Cha94] é realizada uma análise do desempenho dos mecanismos de detecção de erros do CAN. A partir de um modelo analítico para o canal de transmissão do tipo Gilbert-Elliot (2 níveis de erros) [WM95], são obtidas expressões que permitem calcular a probabilidade das mensagens transmitidas conterem erros que não são detectados pelos vários nós da rede (*residual error probability*). É um trabalho especialmente importante, pois é comum considerar que esta probabilidade pode ser obtida directamente das características do CAN (cerca de $4,7 \times 10^{-11}$). Contudo, e devido ao efeito combinado de vários factores (*bursts* de erros, *bit-stuffing*, estrutura das tramas, distribuição espacial dos nós e capacidade dos mecanismos de detecção de erros), verifica-se que este valor está sobre-avaliado. Este estudo é especialmente importante no domínio dos sistemas de segurança crítica (*safety-critical systems*).

Em [RV95] e [VRM97] é realizado um estudo sobre os tempos de inaccessibilidade do CAN quando sujeito a vários tipos de erros de comunicação. São derivadas várias expressões analíticas que auxiliam a descrever este comportamento. Este trabalho contém informação relevante para os trabalhos de relacionados com o escalonamento de mensagens.

[Ruf97a], [Ruf97b] e [RBA99] analisam a robustez da camada física em relação às falhas permanentes e transitórias mais comuns. É proposta e definida uma arquitectura de comunicações (redundante) e tolerante a falhas, quer ao nível do meio físico, quer ao nível da interface de comunicações dos nós.

[BO97] realiza um estudo sobre o comportamento da camada física em ambientes com ruído electromagnético. Este trabalho analisa os efeitos das resistências de terminação, dos *stubs*, de diferentes tipos de condutores e do isolamento galvânico dos dispositivos. O estudo para além de se basear em dados experimentais, permite, entre outros, definir comparativamente quais os efeitos que as falhas com origem externa têm nas mensagens transmitidas.

[RPM+98], [JBA+98] e [RVA99] propõem uma *stack* de comunicações, a partir do CAN, que implementa um conjunto de protocolos de difusão atómica (*atomic broadcasts*) com diferentes tipos de qualidade de serviço. Este trabalho resulta da observação da existência de cenários durante a transmissão de mensagens, onde devido à ocorrência de erros de comunicação, se verificam quer omissões, quer inconsistências nos dados transmitidos.

Em [Tra99] é realizado um estudo muito abrangente sobre as vulnerabilidades do CAN a erros de comunicação que afectam mais do que um bit (*bursts* de erros), e dos quais resultem mensagens com erros que não são detectados pelo CRC. É um trabalho que se centra em analisar de que forma o *bit-stuffing* permite mascarar as situações de erro. Os objectivos são semelhantes a [Cha94], contudo são impostas menos restrições, dado que é utilizado um modelo de simulação para avaliar o comportamento da rede. Para além de identificar algumas inconsistências em [Cha94], realiza também uma análise comparativa com o mesmo, verificando (como seria de esperar) que o trabalho original apresenta em algumas situações resultados optimistas. Este trabalho disponibiliza informações muito importantes em relação à capacidade de detecção do CRC, nomeadamente a identificação e quantificação probabilística dos cenários onde o CRC deveria ser capaz de detectar erros simples, e não o faz. Nestes casos verifica-se que a probabilidade de detecção de erros é várias ordens de grandeza inferior à capacidade teórica do CAN.

Em [GN01] e [GN02] é realizado um estudo sobre os mecanismos de contenção de falhas, nomeadamente sobre a evolução dos contadores de erros TEC e REC. Para isso são desenvolvidos modelos baseados em cadeias de Markov Contínuas que descrevem a evolução, estocástica, dos contadores quando são transmitidas e recebidas mensagens com erros. São analisados dois cenários: o tempo que decorre (valor esperado e variância) até o nó se desligar da rede e o tempo que decorre até o nó se tornar erro-passivo. No primeiro caso o tempo corresponde ao tempo de absorção até ao estado Desligado, enquanto que no segundo ao tempo de estadia no estado Erro-passivo. Embora a definição dos parâmetros das cadeias de Markov seja baseada em valores esperados (e não determinísticos) do tráfego da rede e os modelos contenham algumas simplificações devida à natureza analítica dos mesmos, o trabalho permite avaliar quais as condições (tráfego e taxa de falhas) em que é necessário considerar o funcionamento destes mecanismos.

Em [GNP02a] e [GNP02b] é proposta uma arquitectura onde os nós da rede possuem algum tipo de redundância ao nível dos controladores CAN. Para isso é proposta uma solução de *hardware* que implementa um gestor de redundância desses mesmos controladores. O principal objectivo é permitir a introdução de mecanismos tolerantes as falhas, sem contudo realizar quaisquer alterações no *software* da aplicação.

[FOF+04] propõe uma experiência com o objectivo de determinar a probabilidade de erro num bit (BER – *bit error rate*) de uma rede CAN quando sujeita a ambientes com ruído electromagnético. Os resultados publicados são muito preliminares, mas demonstram a elevada robustez do CAN a interferências externas.

6.3.2 Escalonamento de Mensagens em Cenários de Falhas Durante as Comunicações

Em [TB94] e [TBW95] é realizado um estudo detalhado do escalonamento de mensagens numa rede CAN. Partindo da análise do escalonamento de tarefas com prioridade fixa num processador, desenvolvem-se expressões analíticas que permitem calcular o tempo de resposta da mensagem num cenário pessimista (WCRT - *Worst Case Response Time*), e além disso verificar se as respectivas *deadlines* são violadas, ou não. A análise permite também incluir os efeitos dos erros de comunicação através de um modelo de falhas simples. Neste caso, é definida ao nível do escalonamento uma função de sobrecarga na recuperação de erros (*error recovery overhead function*), cujo valor resulta da imposição de um limite superior à sobrecarga no tráfego da rede devido às retransmissões de mensagens. A principal objecção a esta abordagem está no facto da função ser obtida a partir de parâmetros determinísticos (número e dimensão dos *bursts* de erros num intervalo fixo), não representando assim a ocorrência natural dos fenómenos associados às falhas.

Em [PVT00] o trabalho anterior é extendido através da incorporação dos tempos de inacessibilidade do CAN determinados por [RV95]. Contudo, os modelos apresentados para a ocorrência das falhas continuam a ser determinísticos.

[PHN00] estende o modelo de erros proposto em [TB94] considerando a existência de várias fontes para as falhas. É assumido que cada fonte produz uma interferência com um padrão fixo e específico, consistindo num *burst* inicial de erros e posteriormente numa distribuição dos erros com um intervalo mínimo pré-definido. Embora ao nível dos conceitos o modelo seja interessante, a sua implementação assume também um comportamento determinístico.

Em [NSS00] é proposto um modelo estocástico para a ocorrência de falhas durante a transmissão das mensagens mais próximo da realidade. O modelo é analítico, e têm em conta quer a frequência de ocorrência das falhas, modelada através de um processo de Poisson, quer a gravidade das mesmas, modelada por intermédio de uma função distribuição. O trabalho foca-se na determinação da probabilidade da mensagem violar a sua *deadline* num cenário pessimista (WCDFP – *Worst Case Deadline Failure Probability*), e não no escalonamento das mensagens em si. Embora o trabalho realizado constitua um avanço significativo, inclui duas imprecisões que fazem aumentar o pessimismo da análise. Em primeiro lugar não reflecte correctamente as condições em que uma mensagem poder violar a sua *deadline*. Em segundo lugar, um *burst* de falhas de comprimento n é equivalente a n falhas isoladas (transmissão de uma trama de erro) o que na prática não se verifica.

[HNP00], [HNP01] e [HNN+02] estende o trabalho realizado em [PHN00] através de uma simplificação do modelo de falhas e da inclusão da possibilidade de fases aleatórias entre as interferências. Como é baseado num modelo determinístico, as interacções entre as mensagens e os erros podem ser analisadas através de simulação de forma simples. As medidas focam-se na determinação da probabilidade de uma mensagem violar a sua *deadline*. O modelo centra-se na análise do comportamento da rede durante um período de tempo igual ao menor múltiplo comum do período das mensagens transmitidas (LCM – *Least Common Multiple*). A ideia é assumir que o comportamento durante este intervalo pode ser extrapolado para o restante tempo da funcionamento. No entanto esta abordagem tem algumas limitações. Em primeiro lugar, para cada fonte de interferência é necessário determinar o respectivo padrão. Em segundo lugar, a combinação de várias fontes de erro aumenta a complexidade da análise. Em terceiro lugar, o modelo de ocorrência das falhas é determinístico.

Em [BBB02] é utilizado o conceito de *weakly hard deadlines*, para definir as condições de escalonamento das mensagens. Neste caso assume-se que numa sequência de várias transmissões consecutivas da mesma mensagem algumas destas podem violar a respectiva *deadline*, sem originarem problemas para o sistema em causa. Partindo destas considerações, é realizada uma análise semelhante a [TB94].

Em [BBN02] é proposto um modelo estocástico com o objectivo de resolver os problemas presentes em [NSS00]. A abordagem é semelhante a [TB94], contudo a função de erro é agora determinada de forma probabilística, através da utilização de uma árvore de probabilidades. Esta estrutura representa todos os possíveis percursos probabilísticos decorrentes da ocorrência de falhas durante as comunicações, resultando assim numa solução analítica. Considerando apenas as contribuições dos percursos mais relevantes (numericamente), é obtida a função distribuição do tempo de resposta de cada mensagem, a partir qual é possível verificar se a respectiva *deadline* é violada. É proposto um algoritmo, que mantendo a necessária exactidão, permite percorrer a árvore de forma temporalmente eficiente. Em [BAN04] o trabalho é melhorado utilizando uma forma mais eficiente em termos computacionais de obter os resultados.

[BB03] propõe várias alternativas para a implementação de um guarda do *bus* (*bus-guardian*), que evite comportamentos onde um nó possa transmitir fora do respectivo contexto mensagens (*babbling idiot*), devido à existência de falhas no mesmo. É discutido o comportamento da rede com este dispositivo e realizada uma análise do escalonamento (semelhante a [TB94]) nestas condições.

6.4 Metodologia Proposta

Tendo em conta os trabalhos desenvolvidos previamente, coloca-se a questão de até que ponto é necessário propor uma nova abordagem, e em caso afirmativo quais os objectivos que se pretendem atingir.

Admitindo tal como no capítulo anterior, que os sistemas que se pretendem analisar possuem requisitos de tempo-real associados às transmissão das mensagens de controlo do processo, [BBN02] e [BAN04] representam os trabalhos mais relevantes e mais próximos dos objectivos propostos para este capítulo.

Focando a discussão nestes trabalhos, a análise dos mesmos parece indicar que estes já disponibilizam um boa solução para o problema em causa. Nomeadamente, é desenvolvido um modelo analítico, com todas as vantagens daí decorrentes, que combina de forma adequada a modelação da ocorrência das falhas com um baixo custo da solução em termos dos recursos computacionais exigidos. No entanto, também se verifica que estes trabalhos são baseados num conjunto de pressupostos dos quais resultam cenários muito adversos, implicando assim que os resultados obtidos sejam muito pessimistas. Nomeadamente, os modelos propostos nestes trabalhos assumem que os erros durante a transmissão vão ocorrer sempre no último bit da mensagem (trama) e que vão causar a situação mais grave ao nível do bloqueio (escalonamento) das mensagens. Isto por si só não é problemático, dado que existem vários tipos de aplicações (ex. sistemas críticos com SIL 4 [IEC04b]) que justificam este tipo de abordagem. Contudo, na maioria das situações esta torna-se desnecessária, dado que se for tomado em conta tal nível de pessimismo este aspecto teria também que se reflectir no tipo de soluções adoptadas ao nível da arquitectura de controlo do sistema, resultando desta forma em soluções muito complexas e dispendiosas.

É assim possível concluir que as abordagens propostas neste trabalhos são adequadas a sistemas de segurança crítica que exijam níveis de integridade muito elevados, mas quando aplicadas a sistemas com menores níveis de exigências, resultam em abordagens muito pessimistas. Torna-se assim necessário dispor de modelos de avaliação que reduzam este nível de pessimismo, sem contudo incorrerem na utilização de pressupostos demasiado simplistas que acabem por retirar credibilidade aos resultados obtidos.

Para além destes aspectos, os modelos de falhas propostos nos trabalhos anteriores são muito simples e não reflectem de forma precisa as respectivas condições de ocorrência em ambientes industriais (ex. *bursts*⁹⁷ de falhas). Isto deve-se

⁹⁷ Optou-se por utilizar a terminologia em Inglês por se julgar mais adequada ao contexto.

obviamente aos compromissos resultantes de utilizar um modelo analítico, nomeadamente ao nível da sua solução.

Com base na discussão anterior, é possível concluir que se justifica a necessidade de se desenvolver uma nova abordagem para avaliar a confiança no funcionamento da rede CAN no contexto proposto neste capítulo. A metodologia proposta é semelhante à discutida no capítulo anterior, nomeadamente baseando-se na utilização de redes de Petri Estocásticas para desenvolver os modelos necessários à avaliação do comportamento da rede.

6.4.1 Pressupostos Assumidos

O desenvolvimento do modelo de avaliação foi realizado com base nos seguintes pressupostos:

- ♦ O comportamento da rede foi considerado o mais próximo possível do funcionamento típico das aplicações reais, por oposição à utilização de cenários pessimistas;
- ♦ O modelo que descreve a ocorrência das falhas durante as comunicações deve representar de forma mais exacta as suas características típicas. Neste sentido, assume-se que falhas ocorrem em *bursts* e têm uma duração temporal limitada (Fig. 6.8). Este comportamento corresponde a um cenário de interferências electromagnéticas típico de um ambiente industrial [Sho94] [KWS00];

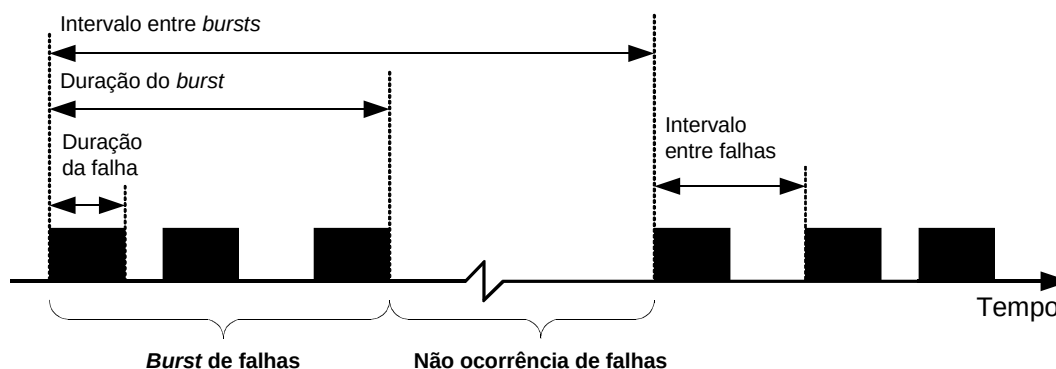


Figura 6.8 - Ocorrência das falhas.

- ♦ As falhas ocorrem de acordo com um processo de Poisson não-homogéneo no tempo [Triv02]. Neste caso a taxa de ocorrência das falhas pode variar ao longo do tempo (na descrição dos modelos serão particularizados as características deste processo). Além disso, assume-se que os seus efeitos são vistos por todos os nós da rede de forma igual;

- ♦ Não se assume qualquer tipo de redundância ao nível do meio físico de transmissão utilizado⁹⁸;
- ♦ As tramas transmitidas pelos nós são de dois tipos: dados⁹⁹ e erros.

As tramas de sobrecarga (*overload*) estão associadas à ocorrência de situações muito particulares que dependem do estado da aplicação, ou da ocorrência de um bit dominante durante o tempo de intermissão. Em relação à primeira situação, a sua inclusão implicaria também a necessidade de modelar o comportamento da aplicação de controlo. Ora como se pretende apenas focar os aspectos relacionados com as comunicações não seria lógico portanto incluí-la. Em relação à segunda situação, esta pode surgir se ocorrerem falhas durante o tempo de intermissão. Neste caso o modelo contempla esta situação de embora de forma indirecta.

As tramas remotas não foram incluídas por duas razões. A mais importante tem a ver com o facto destas tramas serem pouco utilizadas, em comparação com as de dados, em aplicações reais. A segunda razão, tem a ver com o facto destas tramas (em termos de modelo) serem semelhantes às tramas de dados, e desta forma, se necessário, poderem ser facilmente incluídas no modelo;

- ♦ Quando uma falha (interferência) ocorre, assume-se que esta corrompe o estado lógico do meio físico de transmissão (transmissão de tramas, *bus idle* ou tempos de intermissão) através da introdução de erros. Estes erros vão desencadear os mecanismos de sinalização de erros do CAN correspondendo à transmissão de tramas de erros. É também assumido que esses erros (devidos às falhas) são sempre detectáveis por todos os nós da rede. Este último aspecto é suportado pelos trabalhos de [Cha94] e [Tra99], onde se demonstra que a probabilidade de existirem erros não detectados é muito pequena, podendo assim ser desprezada neste contexto;
- ♦ O sistema que se pretende avaliar possui requisitos de tempo-real associados às comunicações, concretizados através da definição de *deadlines* associadas às mensagens transmitidas;
- ♦ A avaria da rede neste contexto, ocorre quando as *deadlines* associadas às mensagens são violadas. Não se assumem diferentes níveis de gravidade para as consequências das avarias (ex. segura, não-segura). Isto deve-se ao facto deste aspecto ser específico das características da aplicação de controlo, sendo assim deixado em aberto;
- ♦ Na ausência de falhas assume-se que não existem quaisquer violações das *deadlines* associadas às mensagens;

98 Este pressuposto foi apenas assumido para simplificar o modelo. Como se poderá ver da discussão nas secções seguintes, se necessário pode ser facilmente introduzido.

99 Denominadas *mensagens* deste ponto em diante.

- ♦ A principal medida de interesse é determinar qual a probabilidade da *deadline* associada à respectiva mensagem (ou mensagens) ser violada.

6.5 Modelos Desenvolvidos

O modelo proposto para o CAN é composto por três modelos de menor dimensão: ocorrência das falhas, transmissão de uma mensagem e sinalização dos erros.

Nas secções seguintes proceder-se-á a uma descrição pormenorizada de cada um dos modelos, bem como a uma discussão mais alargada sobre pressupostos em que estes foram desenvolvidos.

6.5.1 Modelo da Ocorrência das Falhas

Este modelo descreve a ocorrência das falhas durante as comunicações (Fig. 6.9). A ideia subjacente ao mesmo baseia-se no pressuposto de que as falhas ocorrem em *bursts* e têm uma duração temporal limitada.

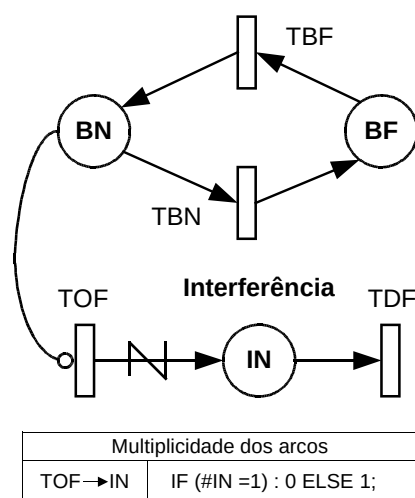


Figura 6.9 - Modelo da ocorrência das falhas.

A utilização de um processo modulado de Poisson de Markov (MMPP - *Markov Modulated Poisson Process*) permite definir um processo de Poisson cuja taxa é modelada por uma cadeia de Markov [Ger00]. É assim possível com este processo estocástico modelar uma taxa de ocorrência de falhas variável no tempo, o que o torna ideal para representar cenários de *bursts* de falhas.

Embora fosse simples modelar múltiplas taxas de falhas, foi assumido neste modelo apenas modelar 2 níveis: com falhas (taxa fixa) e sem falhas. Esta opção foi

motivada por duas razões. Primeiro, a inclusão de vários níveis iria introduzir um nível de complexidade elevado no modelo, que em muitos casos é desnecessária. O uso de múltiplas taxas também obrigaria a conhecer os respectivos valores, o que em termos práticos é difícil de determinar. Segundo, o modelo pode ser facilmente adaptado para incluir múltiplas taxas se necessário bastando para isso realizar uma simples replicação do mesmo. Desta forma, o modelo deve ser encarado como um bloco básico que pode ser utilizado para desenvolver modelos de ocorrência de falhas mais complexos.

As posições *BN* e *BF* representam respectivamente a situação de ausência de falhas, e a situação onde ocorrem *bursts* de falhas. A transição entre níveis é governada pelas transições exponenciais *TBN* e *TBF*, cujas durações correspondem respectivamente ao período de ausência de falhas e de ocorrência dos *bursts*. Inicialmente a posição *BN* encontra-se com um testemunho. O conjunto destas posições e transições implementa uma cadeia de Markov com 2 estados.

O intervalo entre a ocorrência das falhas é modelado por intermédio de processo de Poisson, reflectindo assim o seu comportamento característico. Devido às suas propriedades este processo estocástico permite representar em simultâneo o comportamento de várias fontes de falhas [Triv02]. Em termos de modelo, este comportamento é representado pela transição exponencial *TOF*, cuja duração corresponde ao inverso da taxa do processo de Poisson acima referido. O disparo desta transição só possível quando existir um testemunho em *BF* simbolizando assim o período em que é permitido que ocorram *bursts* as falhas. Este comportamento é assegurado por intermédio do arco inibidor que liga a posição *BN* à transição *TOF*.

Quando a transição *TOF* dispara é colocado um testemunho na posição *IN*, indicando assim a ocorrência de uma interferência¹⁰⁰. O arco que liga a transição *TOF* à posição *IN* tem multiplicidade variável¹⁰¹ por forma a garantir que o número de testemunhos em *IN* nunca exceda o valor de 1. Com isto pretende-se modelar a situação em que a ocorrência de falha durante a ocorrência de uma outra, corresponde na prática a uma única ocorrência.

A duração de cada falha (interferência) é modelada através da transição *TDF*. Aqui, assumiu-se que a duração das falhas têm propriedades idênticas à respectiva ocorrência, sendo caracterizada por uma duração de acordo com uma distribuição exponencial. Desta forma, *TDF* é uma transição exponencial cuja duração corresponde ao valor esperado desta distribuição. A transição *TDF* só dispara quando existe um testemunho em *IN*, representado assim a existência de uma falha. Após o seu disparo o testemunho é removido de *IN* simbolizando o fim da

¹⁰⁰A interferência é a consequência da falha.

¹⁰¹Representado por uma linha em *zig-zag*.

falha. Assume-se que as falhas têm uma duração muito inferior à respectiva ocorrência, o que é um pressuposto muito credível.

6.5.2 Modelo da Transmissão de uma Mensagem

Este modelo foi desenvolvido com o objectivo de descrever o comportamento da rede durante a transmissão de uma mensagem (Fig. 6.10).

Para além dos pressupostos definidos previamente, foram além disso considerados adicionalmente os seguintes:

- ♦ As mensagens são produzidas periodicamente (período constante) pela tarefa que é executada no nó. Este é um pressuposto comum utilizado ao nível dos sistemas que possuem requisitos de tempo-real associados à execução das tarefas [Kop97]. Cada nó produz as suas mensagens de forma independente;
- ♦ Associada a cada mensagem existe uma *deadline*. Considerou-se que a *deadline* de uma mensagem é igual ao período de produção da mesma, correspondendo assim à abordagem clássica do problema [TB94]. Note-se contudo, que poder-se-ia, com algumas alterações ao modelo, incluir outros pressupostos;
- ♦ Se após a produção da mensagem e num intervalo de tempo igual sua *deadline*, esta não for correctamente transmitida considera-se que existiu uma avaria da rede;
- ♦ Se ocorrer um erro durante a transmissão da mensagem assume-se que este é imediatamente detectado pelos nós, iniciando-se no bit imediatamente seguinte à ocorrência do erro a respectiva sinalização. Este pressuposto baseia-se no funcionamento mais comum dos mecanismos de sinalização dos erros do CAN;
- ♦ A duração da transmissão da mensagem é constante e função do volume de dados transmitidos. Assumiu-se que o tempo de propagação no meio de transmissão é desprezável¹⁰².

O modelo proposto reflecte apenas a transmissão de uma única mensagem. Adiante serão indicadas formas de utilizando este modelo representar todas as mensagens que são transmitidas na rede.

A produção da mensagem é modelada através da transição determinística *TPM*, cuja duração corresponde ao período de produção da mesma. Quando esta transição dispara, é colocado um testemunho na posição *PM* simbolizando assim a respectiva produção. A ausência de posição anterior à transição *TPM* garante que

¹⁰²Este aspecto, se necessário, pode ser facilmente incluído no modelo.

esta está permanentemente habilitada, representando assim a produção contínua de mensagens.

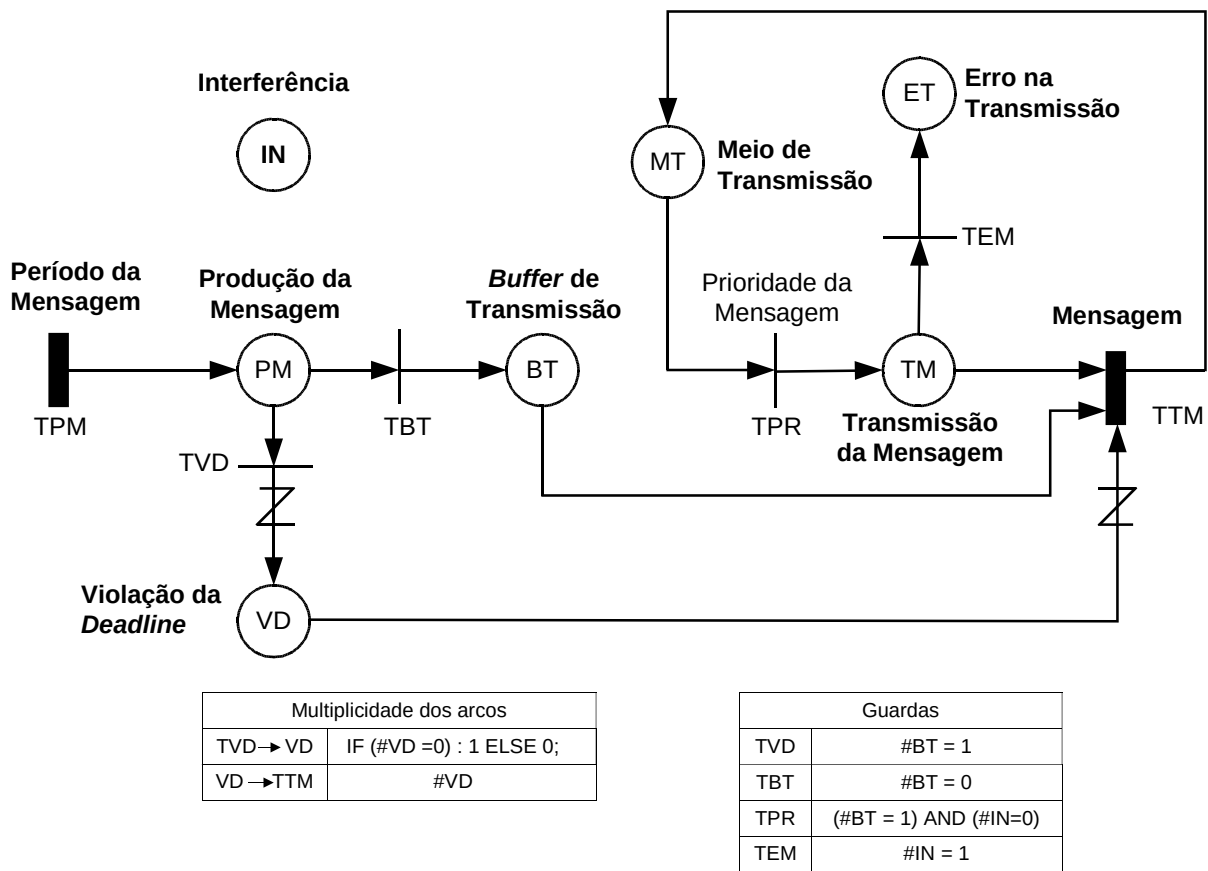


Figura 6.10 - Modelo da transmissão de uma mensagem

Atingida a marcação anterior, uma de duas situações pode ocorrer: se a mensagem produzida anteriormente (a sua última instância) foi transmitida com sucesso, a posição *BT*, à qual corresponde o *buffer* de transmissão, está vazia. Isto dá origem ao disparo da transição imediata *TBT* que remove o testemunho de *PM* e coloca-o em *BT*. No caso contrário, significa que a mensagem produzida anteriormente ainda não foi transmitida e que portanto foi violada a respectiva *deadline*. Nesta situação, a transição imediata *TVD* dispara, removendo o testemunho de *PM* e colocando-o em *VD*. Esta última posição representa a violação da *deadline*. O eventual conflito estrutural existente entre as transições *TBT* e *TVD* encontra-se resolvido através da definição para ambas de funções de guarda dependentes da marcação de *BT* e implementadas de forma mutuamente exclusiva. O arco que liga *TVD* a *VD* também possui multiplicidade variável para assegurar que a posição *VD* não contém mais do que 1 testemunho.

Admitindo que existe uma mensagem para transmitir, a posição *BT* está marcada com 1 testemunho. A indicação de que o meio de transmissão está livre é simbolizada pela existência de um testemunho na posição *MT*. Estando reunidas

estas duas condições e se não existirem interferências (*IN* não está marcada), a transição imediata *TPR* pode disparar, iniciando assim a transmissão da mensagem. Esta transição tem uma função de guarda que depende de *BT* e de *IN* e têm uma prioridade igual à mensagem que se pretende transmitir. Este aspecto será pormenorizado adiante.

A transmissão da mensagem inicia-se com a marcação de *TM* (e a remoção do testemunho em *MT*), que por sua vez habilita a transição *TTM*. Esta última é uma transição determinística, à qual corresponde a soma do tempo de transmissão da mensagem com o tempo de intermissão. Se não existirem falhas (interferências) durante a transmissão da mensagem a transição *TTM* dispara, removendo o testemunho de *TM* e representando assim uma transmissão com sucesso. O disparo desta transição vai também realizar simultaneamente as seguintes funções: remoção do testemunho de *BT*, simbolizando a transmissão da mensagem, remoção de um eventual testemunho de *VD* (arco de multiplicidade variável), indicando que a deixaram de existir violações da *deadline* e colocação de um testemunho em *MT* indicando que o meio de transmissão está de novo livre. Assume-se que sempre que existe uma produção da mensagem, e esta ainda não foi transmitida, é realizada uma re-escrita da mesma no *buffer* de transmissão.

Se durante a transmissão da mensagem existir uma falha (interferência) a posição *IN* encontra-se marcada (ver modelo da ocorrência das falhas). Definindo uma função de guarda para a transição imediata *TEM* dependente desta posição, permite-se que o disparo desta represente a ocorrência de um erro durante a transmissão da mensagem. Quando isto ocorre, o testemunho é removido de *TM* e colocado em *TE*, desabilitando a transição *TTM* e simbolizando desta forma a interrupção da transmissão e a consequente sinalização do erro. Isto decorre do facto das transições imediatas possuírem sempre prioridade no disparo em relação às temporizadas.

Por fim existem alguns aspectos finais sobre o modelo que devem ser discutidos. Dado que as falhas podem ocorrer em qualquer instante durante a transmissão de uma mensagem, os resultados obtidos com o modelo serão necessariamente não-pessimistas. Um segundo aspecto diz respeito à indefinição sobre o momento em que os erros são sinalizados. É um facto que se uma falha ocorrer em certos campos da trama (ex. Identificador) é possível que o erro possa só ser detectado alguns bits mais tarde. Para modelar correctamente este comportamento seria necessário ter em consideração todos os campos possíveis e também as características das falhas (instante de ocorrência e duração). Embora este comportamento pudesse ser implementado, na maioria dos casos é desnecessário, dado que na esmagadora maioria das situações o erro é imediatamente detectado iniciando-se a sua sinalização no bit imediatamente a seguir. Por fim, se as falhas possuírem uma duração muito curta (ex. 1 bit) é possível que em algumas situações os erros não

sejam nem detectados ou sinalizados (ex. Identificador). No entanto estas são situações marginais que têm um impacto muito reduzido nos resultados finais.

É assim possível concluir que as imprecisões resultantes de se assumir que os erros são imediatamente detectados e imediatamente sinalizados são muito pequenas e podem ser ignoradas. Note-se que estes pressuposto dá origem a resultados ligeiramente pessimistas, o que por sua vez garante uma maior confiança nos mesmos.

6.5.3 Modelo da Sinalização dos Erros

Este modelo foi desenvolvido com o objectivo de descrever o comportamento da rede durante a sinalização dos erros de transmissão (Fig. 6.11).

Para além dos pressupostos definidos previamente, foram além disso considerados adicionalmente os seguintes:

- ♦ Ao assumir-se que as falhas têm uma duração aliada à possibilidade das mesmas ocorrerem durante a transmissão de uma trama de erro, é possível que ocorram comportamentos imprevisíveis (ex. quantas tramas de erro são transmitidas ?), que por sua vez são difíceis de descrever e modelar. Para contornar este problema assumiu-se uma abordagem pragmática, com as seguintes características: durante a duração da interferência (falha) o meio físico é considerado totalmente inacessível (nenhuma transmissão de qualquer natureza é realizada); após o fim da interferência é transmitida uma trama de erro. Este comportamento é equivalente a uma sequência de tramas de erro corrompidas, seguida de uma última trama de erro não corrompida;
- ♦ Os nós da rede nunca atingem o estado de erro-passivo. Como pode ser verificado em [GN02] o tempo médio até este estado ser atingido é tipicamente da ordem das centenas ou milhares de horas nos casos mais comuns, tornando-se apenas relevante quando a taxa de ocorrência das falhas é muito elevada. Além disso, este aspecto é muito dependente das características do tráfego da rede.

Ao não incluir este efeito para além de se reduzir a complexidade do modelo, está-se também a assumir que as tramas de erro são todas do tipo erro-activo. Refira-se contudo, que este aspecto pode também ser incluído no modelo. No capítulo 7, na secção referente aos trabalhos futuros será discutida uma possibilidade de incorporar este funcionamento;

- ♦ O problema da incerteza em relação ao instante da sinalização dos erros é resolvido (minimizado) assumindo que as tramas de erro têm sempre a maior duração possível (20 bits). Desta forma todos os cenários são cobertos;

- ♦ Em (§6.5.2) a duração da transição *TTM* inclui o tempo de intermissão (3 bits). Se uma falha ocorrer durante este período é transmitida uma trama de sobrecarga (*overload*). Dado que esta trama tem a mesma estrutura que uma trama de erro-activo, o modelo pode incorporar ambos os comportamentos sem a necessidade de quaisquer modificações.

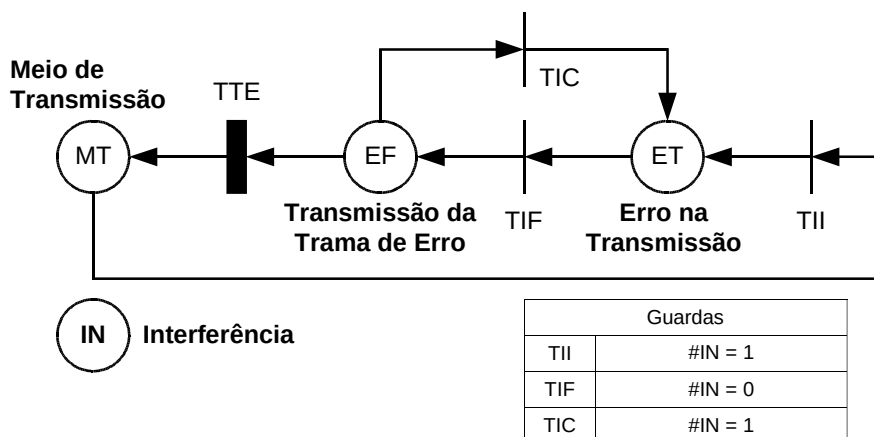


Figura 6.11- Modelo da sinalização dos erros.

Quando a posição *ET* é marcada a transição imediata *TIF* é utilizada para “esperar” pelo fim da interferência, utilizando para isso uma função de guarda que verifica se a posição *IN* não está marcada. Quando esta transição dispara é removido o testemunho de *ET* e a posição *EF* é marcada. Esta posição vai habilitar a transição determinística *TTE* que representa a transmissão de uma trama de erro-activo (ou de sobrecarga) mais o tempo de intermissão. Quando esta transição dispara é removido um testemunho da posição *EF* e a posição *MT* é marcada, o que indica que o meio de transmissão está de novo livre. Depois desta ocorrência a eventual transmissão de uma mensagem inicia-se de acordo com o exposto em (§6.5.2).

Enquanto a trama de erro está a ser transmitida se ocorrer uma falha a transição imediata *TIC* dispara, e o testemunho é removido da posição *EF* e a posição *ET* é de novo marcada. Consequentemente a transição *TTE* é desabilitada e a transmissão da trama de erro é interrompida.

Se durante o *bus idle* (*MT* está marcada) ocorrer uma falha, então inicia-se a transmissão de uma trama de erro. Isto deve-se ao facto da falha dar origem a um bit de *Start of Frame* (SOF) erróneo (Fig. 6.2) que vai dar origem a uma trama de dados virtual (porque na prática não existiu qualquer transmissão). Neste caso a transição imediata *TII* dispara e remove um testemunho da posição *MT* e marca a posição *ET* com um testemunho.

Nos dois últimos cenários descritos, após a marcação da posição *ET* o modelo evolui de acordo com a discussão atrás descrita.

6.5.4 Extensão do Modelo a Várias Mensagens

Pela forma como o modelo foi desenvolvido, a sua extensão para uma análise que contemple a transmissão de n mensagens pode ser realizada de forma muito simples. Para isso é apenas necessário replicar o modelo da transmissão de uma mensagem n vezes, mantendo a posição MT comum a todos eles. Note-se, que a inclusão de uma prioridade na transição TPR incorpora de forma automática o mecanismo de arbitragem do CAN. Assim, se existirem várias mensagens em simultâneo para serem transmitidas, só a que têm maior prioridade é que vai ser transmitida, ou seja, só a transição TPR de maior prioridade é que fica habilitada, podendo assim disparar. Fica assim resolvido um potencial conflito estrutural.

Como nota final refira-se dois aspectos. Todas as transições temporizadas foram implementadas com uma política de memória PRD (PRD - *Preemptive Repeat Different*), e uma política de concorrência do tipo servidor único (*single server*). Com uma pequena modificação, o modelo pode também incorporar o conceito de *instante crítico* referente ao escalonamento de mensagens. Para isso basta apenas considerar que todas as posições PM estão marcadas inicialmente.

6.5.5 Definição das Medidas de Interesse

A probabilidade da *deadline* da mensagem ser violada pode ser obtida através da definição de uma taxa de recompensa (*reward rate*) de valor unitário na posição VD , e do cálculo do valor esperado da recompensa (instantânea) em regime estacionário. Se se estiver interessado no conjunto de todas as mensagens, basta definir a medida como a soma de todas estas taxas de recompensa.

Refira-se que os modelos propostos, pela forma como foram desenvolvidos, podem ser utilizados para obter um número grande de outras medidas, nomeadamente as associadas à capacidade de desempenho (*performability*). Exemplos destas medidas são:

- ♦ Taxa de transmissão de mensagens com sucesso: associar um impulso de recompensa (*impulse reward*) na transição TTM e calcular o valor esperado da recompensa (instantânea) obtida em regime estacionário;
- ♦ Probabilidade de transmissão de tramas de erros: associar uma taxa de recompensa unitária à posição TE e calcular o valor esperado da recompensa (instantânea) obtida em regime estacionário.

6.6 Caso de Estudo

Com o objectivo de validar o modelo proposto foi definido um caso de estudo.

Tal como discutido no capítulo anterior, nunca foi um objectivo deste trabalho realizar uma análise do comportamento específico da rede CAN num cenário de falhas transitórias, mas sim desenvolver modelos que permitissem, de forma eficiente, realizar essa avaliação. Neste sentido, o caso de estudo proposto foca-se exclusivamente na análise das características do modelo, nomeadamente no que diz respeito ao seu desempenho.

Nas próximas secções pretende-se apresentar a forma como os parâmetros foram definidos e também discutir os vários aspectos associados à obtenção da solução do modelo.

6.6.1 Parâmetros do Modelo

Tendo em conta os objectivos previamente definidos, os parâmetros do modelo foram estabelecidos a partir de cenários que se assumiram como típicos das aplicações reais.

6.6.1.1 Definição das Mensagens

A escolha dos parâmetros relacionados com o modelo de transmissão das mensagens (períodos de produção, *deadlines* e duração), é sempre uma opção discutível, para além de que os respectivos valores podem influenciar de forma directa muitas das características do modelo, incluindo o seu desempenho.

Por forma evitar situações deste tipo, optou-se por utilizar conjuntos de mensagens que já tivessem sido alvo de estudos anteriores. Neste sentido, foram consideradas duas alternativas. Por um lado, o conjunto definido em [NSS00] e utilizado como *benchmark* pela Peugeot-Citröen nas redes CAN dos seus veículos, e por outro, o conjunto definido em [TB94] e utilizado pela SAE (*Society of Automotive Engineers*).

Optou-se pelo primeiro conjunto por duas razões: as *deadlines* eram iguais ao período de produção, estando assim de acordo com o pressupostos do modelo, e o número de mensagens é menor, o que permite avaliar de forma mais simples o modelo proposto. No entanto, esta escolha originou alguns problemas devido ao facto do conjunto das mensagens possuir uma taxa de utilização do meio¹⁰³ de

¹⁰³Esta taxa é definida em relação a um conjunto n de mensagens como $\sum_{i=1}^n TTM_i / TPM_i$, em que

transmissão de apenas 21,5%, e desta forma dar origem a valores residuais ao nível das medidas obtidas ([BBN02] demonstra que num cenário deste tipo e mesmo em condições muito pessimistas, a probabilidade de ocorrerem falhas nas *deadlines* é residual (≈ 0)). Para contornar este problema reduziram-se as respectivas *deadlines* (e os períodos de produção) em 3.5 do seu valor, obtendo-se assim uma taxa de utilização de cerca de 75,4%. Com esta redução pretende-se que a uma avaliação se realize num cenário de elevada utilização. Além disso confirmou-se que o conjunto ainda cumpre as respectivas *deadlines* para a velocidade de transmissão utilizada como referência (250Kbits/s).

O conjunto das mensagens utilizado é apresentado na Tabela 6.2. Os respectivos períodos de produção e os tempos de resposta máximos¹⁰⁴ (WCRT – *worst case response time*) estão expressos em bits. Pretende-se desta forma fornecer uma percepção da dimensão relativa dos vários parâmetros do modelo. Utilizou-se como referência uma velocidade de transmissão de 250Kbits/s.

Prioridade (decrecente)	Dados (bytes)	Período (em bits)	Tempo de Resposta Máximo (em bits)
1	8	714	257
2	3	1000	324
3	3	1428	427
4	2	1071	502
5	5	1428	607
6	5	2857	712
7	4	1071	942
8	5	3571	1047
9	4	1428	1397
10	7	7142	1502
11	5	3571	1987
12	1	7141	1990

Tabela 6.2 - Conjunto de mensagens utilizado.

6.6.1.2 Definição dos Cenários de Ocorrência de Falhas

Para os parâmetros do modelo da ocorrência das falhas propuseram-se dois cenários opostos (Tab. 6.3).

TTM_i representa a duração da transmissão da mensagem i , e TPM_i o respectivo período de produção.

¹⁰⁴Obtidos segundo [TB94] e num cenário sem quaisquer falhas.

Cenário	Taxa de Ocorrência (falhas/segundo)	Duração da Falha (bits)
1	500	5
2	50	50

Tabela 6.3 - Cenários de ocorrência das falhas.

O primeiro cenário, corresponde a uma interferência com um nível elevado de ocorrência mas cuja duração é muito curta, enquanto que o segundo cenário corresponde ao caso oposto. A escolha dos valores em si foi apenas motivada por serem um múltiplo de 50 Hz e assim serem próximos do tipo de interferências resultantes de ruído electromagnético existente num ambiente industrial. O objectivo aqui é dispor de dois cenários, de certa forma opostos, que permitam avaliar o comportamento do modelo em duas situações limites.

Os valores assumidos para a taxa de ocorrência merecem um pequeno comentário. Embora os valores utilizados pareçam ser de facto elevados quando encarados de forma absoluta, na prática não o são. Assim como as falhas ocorrem em *bursts*, é natural que durante os mesmos períodos a sua ocorrência seja elevada, decaindo posteriormente para valores muito mais reduzidos. Justifica-se assim a adopção de valores com estas características.

Em relação aos intervalos de ocorrência e de ausência de falhas, assumiu-se em ambos os cenários que estes eram respectivamente 1 minuto e 5 minutos. Note-se que estes valores são totalmente arbitrários. Só uma análise cuidada do ambiente em que o sistema se insere é capaz de fornecer valores credíveis.

6.6.2 Solução do Modelo

Tendo em conta as vantagens oferecidas pelos modelos analíticos, a primeira abordagem passou por avaliar a possibilidade da solução do modelo poder ser obtida através da utilização de técnicas analíticas/numéricas. Contudo, e após vários estudos, verificou-se que só em casos particulares é que seria possível obter uma solução com estas características. Face a estas limitações, optou-se por utilizar técnicas de simulação para obter a respectiva solução.

No sentido de avaliar os motivos que levaram à opção por uma solução baseada em simulação, são discutidos nas secções seguintes os vários estudos realizados.

6.6.2.1 Redes de Petri Discretas Determinísticas e Estocásticas

A partir da análise do modelo, verifica-se que o principal obstáculo à sua resolução analítica está no facto de existirem várias transições determinísticas habilitadas

simultaneamente. Como foi referido no capítulo 2, a existência de uma solução analítica para este tipo de situações só possível em casos muito particulares. Além disso e mesmo neste tipo de casos, as ferramentas que existem presentemente estão limitadas a soluções que consideram apenas existir uma transição deste tipo habilitada em cada instante. A única excepção é a ferramenta **DSPNexpress** [Lin95], que permite obter soluções analíticas com duas transições determinísticas habilitadas simultaneamente.

Tendo em conta que o modelo possui quase exclusivamente transições determinísticas, existe uma possibilidade teórica ao nível da utilização do formalismo subjacente às redes de Petri Discretas Determinísticas e Estocásticas (DDSPN - *Discrete Deterministic and Stochastic Petri Nets*) [ZC96]. De facto, a passagem para um tempo discreto (discretização temporal de passo Δ) pode fazer-se com toda a facilidade ao nível das transições determinísticas. Só ao nível das transições exponenciais é que é cometido um erro de aproximação (são aproximadas por uma distribuição geométrica), que é tanto mais pequeno quanto menor for o passo utilizado na discretização temporal. Esta abordagem torna-se ainda mais aliciante pelo facto de a ferramenta **TimeNet** [ZFG+99] [ZFG+00] suportar este formalismo, bem como pela sua utilização com sucesso para avaliar a exequibilidade do escalonamento de tarefas ordenadas de acordo com o *rate monotonic analysis*, num sistema de tempo-real [ZFH01].

No entanto, e após vários estudos, verificou-se que esta abordagem não era exequível pelos seguintes motivos:

- ◆ **Dimensão do espaço de estados gerado**

Este problema, de certa forma já esperado, tem duas causas. Por um lado, o processo de discretização temporal das transições determinísticas, impõe, para cada transição, a criação de um conjunto de posições adicionais função do passo temporal utilizado. Por outro lado, existe o problema de manter a memória das transições durante a discretização [HPS+00]. Ambas as causas são agravadas pela dimensão da duração das transições determinísticas e pela ausência de factores múltiplos entre estas.

A minimização destes efeitos só é possível através de alterações ao nível do modelo, ou no valor dos parâmetros utilizados. Por exemplo, o passo temporal utilizado na discretização poderia ser aumentado (a duração das transições teria que ser adaptada para reflectir este aspecto) e assim originar um menor número de posições adicionais. Outra alternativa seria considerar que todas as mensagens partilham o mesmo período de produção. Isto permitiria apenas ter uma transição determinística para representar a produção de todas as mensagens;

♦ Processo estocástico subjacente bem definido

Este problema tem origem na possibilidade de existirem várias transições que podem disparar simultaneamente (note-se que o tempo é discreto) [ZC96] [ZCH97]. Caso não seja especificada a sequência de disparo, o processo estocástico subjacente pode ficar definido de forma incorrecta. Este problema pode ser resolvido através da atribuição de prioridades às transições. Para isso, considerou-se atribuir a cada modelo referente a uma mensagem, um conjunto de prioridades distintas, por forma a localizar os conjuntos de conflitos estendidos (ECS) de forma mais simples. Esta tarefa é mais complexa do que a descrição sugere, porque obriga a definir de antemão qual o comportamento probabilístico desejado, o que se torna especialmente fastidioso quando se quer analisar um número elevado de mensagens;

♦ Problemas de funcionamento da ferramenta

Em muitas das situações analisadas, verificou-se que a ferramenta gerava muitos erros de execução. Uma explicação possível (confirmada também por contactos com o grupo que a desenvolveu) é que se trata de uma ferramenta desenvolvida no âmbito de trabalhos de investigação e que nas suas funcionalidades mais avançadas não foi ainda convenientemente depurada de erros.

6.6.2.2 Expansão por Fases Total

A segunda alternativa explorada foi aproximar o comportamento de todas as transições determinísticas através de uma expansão por fases, utilizando para isso várias transições exponenciais em série [MR93]. O principal problema desta abordagem está no facto de ser necessário realizar uma expansão com uma dimensão apreciável para minimizar os efeitos da aproximação. No entanto, o erro cometido pode ainda ser visto de certa forma como positivo. De facto, se for considerado que a produção das mensagens inclui a existência um *jitter*¹⁰⁵ (variação) aleatório, é possível atribuir este efeito à variância da aproximação realizada. Contudo, esta abordagem deve ser realizada de forma cuidada. É o *jitter* que impõem uma variância à aproximação e não o contrário.

Foram avaliadas expansões entre 2 a 5 fases. Apenas nas expansões de pequena dimensão (não mais do que 3 fases) a abordagem definida resultou um modelo analítico, sendo contudo a sua exequibilidade muito dependente do número de mensagens que se pretende analisar¹⁰⁶. Tal como no caso anterior, a dimensão do espaço de estados gerado foi o principal problema. Considerando algumas das restrições indicadas previamente é possível obter resultados satisfatórios,

¹⁰⁵Optou-se por utilizar a terminologia em Inglês por se considerar mais adequada ao contexto.

¹⁰⁶Para além do conjunto de mensagens proposto, foram também avaliados subconjuntos do mesmo.

especialmente devido ao facto de só existir uma transição do tipo *TTM* habilitada de cada vez.

6.6.2.3 Expansão por Fases Parcial

Numa terceira alternativa explorou-se a possibilidade de expandir apenas as transições determinísticas associadas à produção das mensagens, e assim obter um modelo em que apenas uma transição determinística está habilitada de cada vez (*TTM*). O objectivo é duplo. Por um lado minimizar o espaço de estados gerado face à abordagem anterior, e por outro dispor de um modelo analítico capaz de ser resolvido pelas ferramentas existentes que suportem processos Não-Markovianos.

Esta abordagem também só resultou para pequenas expansões (não mais do que 3 fases). Contudo, as razões não são exactamente as mesmas do caso anterior. Embora o espaço de estados gerado fosse perfeitamente razoável (entre algumas centenas e algumas dezenas de milhares de estados, função do número de mensagens a analisar e do tipo de expansão realizado) o cálculo da solução tornou-se extremamente exigente em termos de recursos computacionais (tempo de CPU). Isto deve-se às características dos cálculos que são necessários para resolver este tipo de problemas [Ger00]. De forma geral, este cálculo envolve, para cada transição determinística habilitada, a determinação de várias matrizes exponenciais referentes à cadeia de Markov subordinada à transição e posteriormente a respectiva integração.

6.6.2.4 Simulação

A última alternativa foi utilizar a simulação. A principal vantagem desta abordagem está no facto de não existirem quaisquer limitações estruturais ao nível do modelo que se pretende analisar. Acresce ainda a possibilidade de manter todas as vantagens em utilizar um modelo simulação baseado em redes de Petri. O principal problema está no facto da solução obtida ser uma aproximação, e do tipo de recursos computacionais que são necessários para obter a solução.

Quando se optou por esta abordagem, foram considerados os seguintes requisitos:

- ♦ O modelo desenvolvido tem que ser suficientemente simples para garantir a partida um desempenho do mesmo elevado. Neste contexto, pretende-se executar o maior número de simulações possíveis no mais curto intervalo de tempo;
- ♦ A ferramenta de simulação tem que ser muito eficiente, e se possível incluir mecanismos que permitissem reduzir o número de simulações necessárias.

Ambos os requisitos são muito importantes quando se consideram os objectivos definidos inicialmente. Face aos trabalhos de referência previamente discutidos, o

modelo proposto neste capítulo possui vantagens acrescidas ao nível do rigor com que o comportamento da rede é modelado. Contudo, como a solução do mesmo é obtida através de simulação, este facto acaba por ser uma desvantagem. Uma das formas de contrariar este aspecto, é garantir que a solução do modelo é obtida de forma eficiente. Esta eficiência resulta da acção combinada de ambos os requisitos atrás definidos, daí a necessidade da sua referência.

Pelas características do modelo apresentado o primeiro requisito encontra-se automaticamente satisfeito. Quanto ao segundo, optou-se por utilizar a ferramenta **TimeNet** [ZFG+99] [ZFG+00]. Esta escolha deveu-se essencialmente às características do seu simulador, entre as quais se destacam:

- ♦ Capacidade de executar simulações em ambientes distribuídos e heterogéneos [Kel95]. Além disso, inclui vários aspectos avançados sobre a forma de como as amostras da simulação devem ser recolhidas;
- ♦ Implementação de técnicas de redução da variância [HW81] [Kel94]. A partir de uma análise da estrutura da rede de Petri, são definidas, de forma automática, um conjunto de variáveis de controlo (*control variates*) que permitem obter um estimador com menor variância e assim reduzir o número de simulações;
- ♦ Possibilidade de se realizarem simulações de eventos raros utilizando técnicas divisão (*splitting*) dos percursos amostrais [Kel96];
- ♦ Facilidade no desenvolvimento do modelo, nomeadamente ao nível da definição de funções de guarda e de prioridades das transições.

6.6.3 Caracterização do Desempenho do Modelo

Tal como discutido em (§5.6.2) a utilização de um modelo cuja solução é obtida através de simulação obriga a uma caracterização do seu desempenho. Tendo em conta que os factores que condicionam o desempenho deste modelo são em todo idênticos aos apresentados e discutidos em (§5.6.2), optou-se por não os referir de novo, excepto no que diz respeito ao processo de ocorrência de falhas que agora possui características diferentes:

- ♦ **Taxa de ocorrência das falhas e respectiva duração**

O valor destes parâmetros influencia de forma directa a probabilidade de ocorrer uma violação da *deadline*, dado que introduzem maior, ou menor, nível interferências. A sua influência não está respectivo valor absoluto, mas sim no seu valor relativo face aos restantes parâmetros do modelo, nomeadamente em relação ao período de produção e aos tempos de transmissão das mensagens. Assim, e se forem escolhidas para as falhas taxas de ocorrência

elevadas e com longas durações face aos restantes parâmetros, os tempos de simulação serão necessariamente mais curtos, e vice-versa.

Com o objectivo de analisar o desempenho do modelo de simulação utilizou-se uma abordagem equivalente à discutida no capítulo anterior. Nomeadamente, utilizou-se a mesma métrica para caracterizar o desempenho. Na continuação da mesma, também não foram utilizadas as funcionalidades do ambiente de simulação distribuído disponibilizado pela ferramenta. Todas as simulações foram realizadas numa única máquina, correndo apenas um único processo de simulação.

Os problemas colocados ao nível do desempenho do modelo levaram mesmo a uma ligeira alteração do modelo proposto. A existência de intervalos de tempo em que não ocorrem falhas (segundo o modelo proposto), vai corresponder a um desperdício de tempo de simulação, pois não vão existir quaisquer violações das *deadlines* durante estes períodos (logo não existem amostras das medidas). Em termos da solução isto dá origem ao aumento do tempo de simulação devido à necessidade de obter mais amostras, acrescido pelo facto destas possuírem também uma maior variância.

Por forma minimizar estes efeitos optou-se por remover a parte do modelo referente ao processo modulado de Poisson (posições *BN* e *BF* e transições *TBN* e *TBF*), mantendo-se no entanto as restantes. Contudo, para não alterar as características do modelo realizou-se um ajuste na taxa de ocorrência das falhas, que neste caso é “pesada” pela probabilidade da posição *BF* estar marcada. Dado que *BN* e *BF* representam uma cadeia de Markov com 2 estados o cálculo é trivial. Refira-se que embora se tenha realizado uma alteração estrutural no modelo, os resultados obtidos em regime estacionário são exactamente os mesmos, caso a alteração não tivesse sido realizada.

Como alternativa à abordagem anterior, é possível remover esta parte do modelo sem contudo alterar quaisquer taxas. No caso anterior, a probabilidade obtida é a média entre os cenários em que ocorrem e não ocorrem falhas, enquanto que neste caso se propõe que esta medida seja realizada num cenário onde ocorrem sempre falhas. Como se sabe à partida que quando não ocorrem falhas a probabilidade das *deadlines* serem violadas é zero, utilizando uma abordagem baseada na lei da probabilidade total [Triv02] é possível concluir que a probabilidade da *deadline* ser violada é igual a

$$Prob\{Deadline\ Violada|Ocorrem\ Falhas\} \times Prob\{Ocorrem\ Falhas\} \quad (6.1)$$

em que $Prob\{Deadline\ Violada|Ocorrem\ Falhas\}$ é obtida da simulação realizada no cenário anterior, e $Prob\{Ocorrem\ Falhas\}$ é obtida da cadeia de Markov previamente referida calculando a probabilidade da posição *BF* estar marcada.

Resulta assim num modelo de simulação muito mais eficaz, bastando posteriormente realizar o cálculo acima indicado para ajustar os resultados obtidos. Note-se, que esta abordagem permite estender a utilização destes modelos a cenários onde existem períodos curtos de tempo onde o nível de ocorrência das falhas é muito elevado, seguidos de períodos longos de ausências de falhas, sem quaisquer custos computacionais. Dadas as vantagens inerentes a esta abordagem, esta foi utilizada para obter os resultados dos modelos neste caso de estudo.

6.6.4 Resultados Obtidos

Nesta secção são apresentados resultados obtidos nas simulações para a rede proposta, utilizando uma velocidade de transmissão de referência de 250Kbits/s. As simulações foram realizadas num PC equipado com processador AMD Athlon @ 1145Mhz, 256KB cache, 512MB de RAM, com um sistema operativo Linux (Suse 7.2) kernel 2.4.10.

A Tabela 6.4 apresenta os resultados obtidos de acordo com a métrica definida anteriormente para avaliar o desempenho do modelo. Neste caso, são indicados os tempos de execução reais necessários para executar 1 simulação (uma realização amostral) correspondendo a um intervalo de tempo simulado fixo. Foi utilizada uma simulação terminal, cuja duração corresponde ao intervalo definido.

Cenário	Tempo Real de Simulação - 1 simulação	
	Intervalo simulado – 1s	Intervalo simulado – 10s
1	50 (ms)	480 (ms)
2	58 (ms)	570 (ms)

Tabela 6.4 - Tempo real necessário para executar 1 simulação.

A partir da análise dos resultados verifica-se que a o ganho em relação ao tempo simulado é cerca de 20 vezes. Refira-se, que os tempos apresentados não são tempos de execução no CPU mas sim tempos reais de simulação, com todo o conjunto de processos típicos a serem executados em simultâneo pelo sistema operativo.

Tal como no caso de estudo apresentado no capítulo anterior, a qualidade do modelo proposto em relação ao seu desempenho pode ser avaliada através da relação entre o tempo simulado e o tempo real de simulação. Neste caso, a relação é muito elevada o que permite concluir desde já que o desempenho do modelo se pode classificar de muito bom. Entre os vários factores que contribuem para este bom resultado destacam-se o baixo nível de complexidade do modelo, e a eficiência do simulador utilizado. A pequena diferença existente entre os resultados obtidos pode ser explicada pelo facto da taxa de utilização do meio ser elevada. Desta

forma, a maioria do tempo de simulação é “gasto” na transmissão das próprias mensagens, sendo os efeitos das falhas menos preponderantes. Para este facto também contribui em muito a forma como se processa de transmissão das mensagens no CAN (não-preemptivo e baseado em prioridades).

Por forma comprovar a qualidade efectiva do desempenho do modelo, realizou-se uma simulação completa com o objectivo de determinar as medidas de interesse. (Tab. 6.5). A medida realizada incidiu sobre a probabilidade de violação da *deadline* da mensagem de prioridade 9 (M9). A opção por esta escolha deveu-se ao facto desta mensagem ser entre as várias a que possui o tempo de resposta máximo (WCRT) mais próximo da respectiva *deadline* (Tab. 6.2), e assim a que assume, aparentemente, a maior probabilidade de violação. Os resultados apresentados indicam o tempo real de simulação que foi necessário para obter a medida anterior, no ambiente computacional definido previamente.

De acordo com a medidas de interesse definidas, as simulações foram obtidas em regime estacionário utilizando a técnica da média dos lotes (*batch means*). Neste caso não é necessário indicar quais as dimensões dos lotes e do período transitório porque a ferramenta implementa um conjunto de técnicas avançadas que automaticamente determinam estes parâmetros [HW81] [Zim01].

Cenário	Tempo Real de Simulação – M9 – Simulação Completa Intervalo de confiança: 95% - Máximo erro relativo: 10%
1	39 segundos
2	501 segundos

Tabela 6.5 - Tempo real de uma simulação completa.

A simulação completa foi realizada num contexto que deve ser analisado com algum cuidado, devido ao facto de existirem aspectos que contribuem de forma oposta para os resultados obtidos:

1. O ambiente de computação utilizado possui, na perspectiva dos padrões actuais, um poder de cálculo que pode ser considerado de nível médio em termos comparativos, nomeadamente no que diz respeito ao CPU. Na perspectiva dos resultados, este aspecto pode ser encarado como ligeiramente conservativo;
2. O intervalo de confiança foi definido com um nível de confiança elevado (0.95), e com uma largura relativa muito reduzida (10%), o que implica que o número de simulações realizadas seja comparativamente maior. No primeiro cenário, a situação torna-se ainda mais exigente porque as medidas de interesse tomam valores muito baixos, o que implica que a largura efectiva do intervalo seja muito reduzida, resultando assim num maior número de simulações;

3. O conjunto de mensagens utilizadas possui uma taxa de utilização do meio de transmissão que é elevada, potenciando assim uma elevada interferência entre estas e aumentando desta forma as probabilidades de violação das respectivas *deadlines*. Este aspecto vai assim contribuir para um optimismo dos resultados;
4. A medida de interesse definida apresenta uma característica interessante. Embora a mensagem M9 seja aquela que possui o tempo de resposta mais próximo da respectiva *deadline*, o mecanismo de acesso ao meio do CAN pode influenciar em muito este aspecto, e na prática existir uma outra mensagem que possui uma maior probabilidade de violar a respectiva *deadline*. Desta forma, não é possível de afirmar de antemão que esta corresponde ao caso mais optimista. No entanto para retirar quaisquer dúvidas, realizaram-se outras simulações para as restantes mensagens e comprovou-se (em termos estatísticos) que a mensagem 6 é de facto a que possui a maior probabilidade de violar a respectiva *deadline*, confirmado desta forma o optimismo deste caso;
5. O segundo cenário representa uma taxa de ocorrência de falhas relativamente baixa, pelo que os resultados obtidos devem ser interpretados em relação a este parâmetro de forma positiva. A enorme diferença entre os cenários 1 e 2 resulta do facto da taxa de falhas ter aumentado 10 vezes. A explicação para este comportamento é semelhante à apresentada no capítulo anterior num contexto idêntico.

Partindo da discussão realizada e ponderando os vários aspectos discutidos, é possível concluir, no contexto dos cenários de falhas assumidos, que os resultados obtidos devem ser encarados como valores um pouco optimistas. Contudo, é importante realçar que em termos absolutos os resultados da simulação completa são de facto muito bons, confirmando assim o bom desempenho do modelo. Refira-se também, que a utilização do ambiente de simulação distribuída disponibilizado pela ferramenta pode melhorar de forma dramática os resultados obtidos, enfatizando ainda mais este aspecto.

A possibilidade de dispor de um modelo de simulação juntamente com os bons resultados obtidos ao nível do seu desempenho, permitem antever a possibilidade de incluir um maior nível de detalhe no modelo sem comprometer o respectivo desempenho. Neste contexto, seria interessante analisar o efeito dos contadores de erros REC e TEC em cenários de falhas com elevada taxa de ocorrência, e também a existência de mensagens com erros não detectados.

A utilização da técnica da decomposição comportamental para incorporar os resultados obtidos com este modelo é realizada nos moldes discutidos no capítulo anterior, pelo que se optou por não a descrever de novo.

6.7 Conclusões

Neste capítulo é proposto um modelo para avaliar a confiança no funcionamento de uma rede CAN em cenários de falhas transitórias que ocorrem durante as comunicações.

O modelo desenvolvido resulta de uma abordagem centrada na redução do nível de pessimismo adoptado em análises anteriores, utilizando para isso modelos de falhas mais próximos da realidade e modelando o comportamento da rede de acordo com as suas condições reais de operação.

Embora a solução do modelo seja obtida através de simulação, os resultados obtidos a partir do caso de estudo realizado permitem concluir que o modelo proposto possui um desempenho muito bom a este nível. O contexto em que os resultados foram obtidos, permite antever que a utilização de um ambiente de simulação distribuído pode melhorar de forma dramática os tempos de simulação.

É assim possível concluir que o modelo desenvolvido é rigoroso na abordagem, reduz o nível de pessimismo nas medidas, e é perfeitamente exequível ao nível da solução, utilizando recursos computacionais normais e tempos de execução perfeitamente razoáveis em função do cenário de falhas definido.

Face a este conjunto de aspectos, é possível concluir que o modelo desenvolvido cumpre com os objectivos propostos inicialmente.

Esta página foi intencionalmente deixada em branco

Capítulo 7

Conclusões

O objectivo principal desta dissertação centrou-se no desenvolvimento de modelos estocásticos que permitam realizar uma avaliação da confiança no funcionamento de arquitecturas de controlo distribuídas suportadas por redes de campo. Pretende-se assim contribuir para o processo de validação deste tipo de sistemas.

A avaliação deste tipo de sistemas é naturalmente uma tarefa complexa, devido essencialmente ao seu carácter inter-disciplinar e à natureza distinta dos vários componentes do sistema. Estes têm características muito diversas, possuem diferentes graus de importância, situam-se em níveis funcionais distintos e interagem de forma diferenciada entre si.

Aliada à realidade anterior, verifica-se muitas das vezes um conhecimento limitado sobre o comportamento real do sistema na presença de falhas. Este aspecto pode resultar de dois factores. Por um lado, o sistema pode ainda estar numa fase de especificação e desenvolvimento, e portanto ainda não estarem bem definidas as suas características. Por outro, a informação referente ao comportamento do sistema pode simplesmente ser escassa devido ao desconhecimento do respectivo processo de desenvolvimento, como é o caso de muitas das soluções comerciais existentes.

Face a este conjunto de aspectos, a abordagem a este problema foi realizada de acordo com duas perspectivas complementares. Para o caso dos sistemas dos quais

se possui um conhecimento limitado, procuraram-se desenvolver modelos que permitam uma avaliação preliminar do seu comportamento, e que desta forma possam ser utilizados na tomada de decisões entre propostas de concepção alternativas nas fases desenvolvimento, ou na definição das condições de operação do sistema. Esta foi a perspectiva utilizada nos modelos desenvolvidos no capítulo 3. No caso de sistemas mais complexos e para os quais se dispõe de um maior número de dados, procurou-se desenvolver modelos integrados que incluam diferentes perspectivas sobre o funcionamento do sistema. Esta foi a abordagem utilizada nos capítulos 4, 5 e 6.

Os modelos de falhas permanentes propostos no capítulo 3, são obtidos a partir da identificação dos componentes da rede cuja avaria, neste contexto, vão implicar as maiores consequências para o funcionamento da mesma. As limitações existentes nos dados disponíveis, implicam ao nível da modelação a adopção de uma abstracção de alto nível centrada nos nós e na infra-estrutura de comunicações, bem como a utilização pressupostos simples ao nível da ocorrência das falhas. Os modelos são baseados em cadeias de Markov contínuas, e descrevem o comportamento de duas arquitecturas base, incluindo as avarias em modo-comum e as reparações dos componentes. Além disso, os modelos possuem também uma estrutura genérica. Ambos os aspectos combinados, permitem que estes modelos possam ser utilizados, através da sua composição, para avaliar arquitecturas de controlo mais complexas. Garante-se assim uma gama de aplicabilidade alargada para os mesmos. Para além da solução numérica permitir obter as medidas de interesse, é apresentada uma técnica que permite extrair dos modelos expressões analíticas referentes a essas mesmas medidas. Embora a técnica resulte de uma aproximação, os resultados obtidos possuem um elevado grau de rigor. Desta forma, o processo de avaliação fica facilitado por dois motivos. Por um lado, não é necessário recorrer a ferramentas de modelação específicas para obter os resultados, e por outro, é possível a partir das expressões obtidas identificar os parâmetros que mais influenciam o comportamento da rede e assim simplificar os modelos. O caso de estudo apresentado, embora focado num exemplo específico, permitiu identificar que entre os vários parâmetros dos modelos, os factores de cobertura (nas suas várias vertentes) e as avarias em modo-comum são os que assumem maior importância na definição do comportamento da rede. Os modelos propostos possuem todas as características que lhes permitem ser utilizados com sucesso numa avaliação preliminar. Isto aplica-se quer à capacidade de extensão dos modelos para a avaliação de arquitecturas mais complexas, quer ao nível do rigor que incluem face às restrições existentes. Representam assim um meio útil para a tomada de decisões entre propostas de concepção alternativas ou relacionadas com a operação da rede.

O modelo das falhas transitórias proposto no capítulo 3, resulta de um compromisso entre o rigor e o detalhe com que o comportamento da rede é

representado, e a necessidade de dispor de um modelo analítico. O seu desenvolvimento assentou num aspecto central dos sistemas de controlo suportados por redes de campo, nomeadamente na existência de requisitos de tempo-real associados às comunicações. O modelo proposto resulta da combinação de uma abordagem baseada em argumentos probabilísticos com cadeias de Markov discretas. O seu desenvolvimento foi efectuado de acordo com a estrutura da *stack* de comunicações, podendo desta forma ser caracterizado segundo duas perspectivas. Ao nível da camada de física, o modelo permite avaliar os efeitos de falhas com características diversas sobre a transmissão de tramas, incluindo a existência de meios de transmissão redundantes e de avarias em modo-comum dos mesmos. Neste caso os modelos podem ser considerados muito rigorosos, o que permite a sua utilização com confiança para avaliar redes com características muito diversas. Ao nível das camadas superiores, o comportamento da rede é apenas considerado até à camada de ligação de dados. Além disso e como o objectivo de obter um modelo analítico, o funcionamento específico do protocolo de comunicações definido ao nível desta camada é, em alguns aspectos, modelado utilizando uma abstracção de alto nível, o que implicou utilizar algumas simplificações. Ambos os aspectos combinados acabam por resultar numa aproximação em relação ao comportamento real da rede. No entanto, este tipo de aproximações podem ser consideradas muito razoáveis quando se avalia o comportamento da rede em condições normais de operação. Além disso e em função do tipo de utilização dada aos resultados dos modelos, é ainda possível minimizar os efeitos das aproximações empregues. Se os resultados forem interpretados de forma relativa, ou seja, como uma forma de avaliar comparativamente soluções alternativas, as conclusões obtidas a partir desta análise podem considerar-se bastante credíveis. O caso de estudo apresentado, embora focado num exemplo específico, permitiu concluir, entre vários aspectos, que o período de transmissão das mensagens desempenha um papel fundamental no comportamento da rede. Além disso, também se verificou que a introdução de um meio de transmissão redundante permite aumentar consideravelmente a qualidade das medidas obtidas. Os modelos propostos possuem assim todas as características que lhes permitem ser utilizados com sucesso numa avaliação preliminar focada numa análise comparativa entre soluções alternativas.

Com o objectivo de permitir uma avaliação sobre o comportamento do sistema de controlo suportado pela rede que inclua diferentes perspectivas sobre o funcionamento do mesmo, foram propostos no capítulo 4 vários modelos que permitem realizar uma avaliação integrada. O processo de desenvolvimento destes modelos foi uma tarefa complexa. O carácter multi-disciplinar das várias perspectivas que se pretendem incluir nos modelos obrigou a uma análise cuidada das mesmas com o objectivo de identificar e incluir apenas as mais importantes ao nível do comportamento da rede. Juntamente com este facto, a natureza distinta dos componentes da rede, com diferentes funcionalidades e com interacções

complexas entre si, obrigou à utilização de um formalismo com elevado poder de modelação, concretizado através da utilização de redes de Petri Estocásticas. Contudo, a sua utilização implicou também alguns problemas, nomeadamente ao nível da dimensão do espaço de estados subjacente ao modelo. Neste sentido, uma das componentes da metodologia utilizada no desenvolvimento dos modelos focou-se particularmente na utilização de técnicas e ferramentas de modelação apropriadas para a resolução deste problema. Neste contexto, foi dada especial relevância à utilização da técnica da decomposição comportamental e à utilização da ferramenta Möbius, que implementa uma técnica específica de redução do espaço de estados. Os modelos obtidos resultam de uma abordagem modular baseada numa decomposição estrutural e funcional do sistema, e incluem, de forma integrada, aspectos distintos, tais como: o comportamento dos componentes de *hardware* e de *software* situados em diferentes níveis hierárquicos e com diferentes funções, dependências funcionais e propagação de erros entre componentes, falhas com diferentes características (permanentes, transitórias e em modo-comum), reparações de vários níveis e tipos, e modos de avaria da rede que consideram diferentes níveis de gravidade. Embora os modelos possuam um carácter genérico, a sua estrutura está definida de tal forma que é possível incluir novas funcionalidades nos mesmos, ou simplificar o comportamento de vários mecanismos cujos detalhes individuais se desconhecem, mas que é possível caracterizar em termos de funcionamento conjunto. Ao invés de realizar um caso de estudo particular, procurou-se essencialmente caracterizar a qualidade dos modelos obtidos em função da dimensão do espaço de estados gerado. Esta opção resultou do facto deste ser o principal problema encontrado ao nível da modelação. Os resultados obtidos permitem concluir que, na esmagadora maioria dos casos, o espaço de estados gerado possui uma dimensão totalmente compatível com os recursos computacionais existentes presentemente. Isto permite antever a possibilidade de se introduzir ainda um maior nível de detalhe nos modelos, sem comprometer a respectiva solução. Os modelos possuem assim um elevado nível de rigor e de detalhe, o que permite garantir a sua utilização com sucesso na avaliação de arquitecturas de controlo complexas.

Os modelos de avaliação em cenários de falhas transitórias que ocorrem durante as comunicações, propostos nos capítulos 5 e 6 para as redes Profibus-DP e CAN respectivamente, enquadram-se num objectivo múltiplo. Ao contrário dos modelos desenvolvidos no capítulo 3, estes são muito mais rigorosos e detalhados com a intenção de permitir a sua utilização em avaliações absolutas, ao contrário dos primeiros. O segundo objectivo, prende-se com a importância com que se reveste este tipo de avaliação, função do papel central assumido pela infra-estrutura de comunicações no funcionamento do sistema e conseqüentemente dos efeitos que as falhas na mesma implicam para o processo controlado. Tal como no capítulo 3, o desenvolvimento dos modelos assume a existência de requisitos de tempo-real associados às comunicações. Por fim, pretende-se incorporar os resultados do

modelo no modelo integrado proposto no capítulo 4 utilizando a técnica da decomposição comportamental.

A necessidade de dispor de modelos bastante pormenorizados, com o objectivo de reflectir as condições reais de funcionamento da rede, obrigou à utilização de modelos baseados em redes de Petri Estocásticas. Estas demonstraram ser um formalismo adequado para desenvolver um modelo com estas características. Contudo, o nível de detalhe adoptado implicou que o processo estocástico subjacente ao modelo deixasse de poder ter uma solução analítica, o que obrigou à utilização de técnicas de simulação para a resolução dos modelos. A utilização de ferramentas de modelação de redes de Petri que suportem este tipo de solução veio em muito facilitar esta tarefa. Para além de já incorporarem as técnicas associadas à simulação de eventos discretos, muitas delas de carácter avançado, permitem que o modelo seja validado de forma adequada, utilizando por exemplo as propriedades obtidas do mesmo, além de possibilitarem um desacoplamento completo entre o modelo e a respectiva solução.

Para a rede Profibus-DP, propôs-se um modelo que representa a *stack* de comunicações completa, incluindo os aspectos específicos da implementação DP. Isto inclui para além das mensagens de dados trocadas entre mestres e escravos, as fases de configuração, parametrização e verificação dos escravos. Está-se assim perante um modelo extremamente rigoroso e detalhado, que representa praticamente o funcionamento exacto do Profibus-DP em aplicações reais. Além disso, o modelo inclui todo o tipo de comportamentos da rede em cenários onde ocorrem falhas durante as comunicações. Neste contexto, um dos aspectos importantes diz respeito ao facto de alguns destes cenários não terem sido previstos, ou até mal formulados, em trabalhos desenvolvidos previamente. Ao nível da solução dos modelos, estes foram avaliados através do seu desempenho durante a simulação. A utilização de uma métrica independente para caracterizar o desempenho permitiu que esta avaliação fosse realizada de forma clara e objectiva. Os resultados obtidos permitem concluir que os modelos possuem um bom desempenho, permitindo assim obter tempos de simulação perfeitamente razoáveis em função do cenário de falhas definido. A forma conservadora como os resultados foram obtidos, permite antever que a utilização de um ambiente de simulação distribuído oferecido pelas ferramentas de modelação pode melhorar de forma dramática os tempos de simulação necessários. É assim possível concluir que o modelo desenvolvido é rigoroso na abordagem e perfeitamente exequível ao nível da sua solução sem utilizar recursos computacionais exigentes, permitindo assim a sua utilização numa gama alargada de cenários de avaliação.

Em relação à rede CAN, o modelo proposto procurou essencialmente reduzir o pessimismo das análises desenvolvidas em trabalhos anteriores. Isto foi realizado de duas formas. Por um lado, utilizaram-se modelos de falhas mais próximos da

realidade de um ambiente industrial, nomeadamente incluindo *bursts* de falhas. Por outro lado, a modelação da ocorrência de erros de transmissão e a respectiva sinalização foi realizada de acordo com as condições reais de operação do CAN. Ambos os aspectos combinados vão permitir obter resultados menos pessimistas. Tal como no capítulo 5, a utilização de simulação implicou a utilização de uma métrica para a avaliação do desempenho dos modelos com as mesmas características. Os resultados obtidos, demonstram que o modelo tem um desempenho muito bom, e tal como no caso anterior permite também antever que os tempos de simulação possam melhorar substancialmente através da utilização de um ambiente distribuído. É assim possível concluir que o modelo desenvolvido é rigoroso na abordagem, reduz o nível de pessimismo dos resultados, sendo perfeitamente exequível ao nível da sua solução sem utilizar recursos computacionais exigentes, permitindo assim a sua utilização numa gama alargada de cenários de avaliação.

7.1 Perspectivas de Trabalhos Futuros

O trabalho realizado nesta dissertação permitiu estabelecer as fundações para trabalhos futuros no domínio da avaliação da confiança no funcionamento de arquitecturas de controlo distribuídas, através da utilização de modelos de estocásticos.

No contexto dos trabalhos apresentados nesta dissertação, embora se tenha procurado ser o mais exaustivo possível no desenvolvimento dos modelos, como em qualquer trabalho, existem sempre aspectos que podem ser melhorados. Nesta perspectiva, é possível identificar um conjunto de áreas onde será possível futuramente apresentar contribuições relevantes:

- ♦ O modelo integrado proposto no capítulo 4 pode ser melhorado ao nível dos seus componentes de *software*. Nomeadamente, seria importante definir a componente funcional existente entre tarefas executadas nos nós e a forma como estas interagem. Neste sentido, devem ser sistematizadas, de acordo com a perspectiva anterior, um conjunto de arquitecturas típicas de controlo e desenvolvidos os respectivos modelos de avaliação. Entre as várias características que estes modelos deveriam incluir, salientam-se: a topologia funcional entre tarefas, a propagação de falhas entre tarefas situadas em nós distintos e a possibilidade de uma tarefa estar em diferentes estados funcionais.

É notório que todos estes aspectos fazem aumentar o espaço de estados do modelo, pelo que o seu desenvolvimento deve ser realizado no contexto da

ferramenta Möbius, por forma a tirar partido das técnicas de redução do espaço de estados por esta disponibilizadas. Note-se contudo, que devido às restrições de modelação impostas por esta ferramenta, este é um trabalho complexo que obriga a uma análise considerável ao nível da modelação;

- ♦ O modelo proposto para avaliação da rede Profibus-DP contém em si um conjunto de potencialidades que, se forem devidamente exploradas, permitem obter modelos com maior eficácia ao nível da sua solução.

Um dos aspectos que pode ser explorado, é a eventual possibilidade de obter um modelo analítico que descreva o comportamento da rede. Para conseguir este objectivo é necessário utilizar uma abordagem em duas fases. A primeira, passaria por eliminar do modelo proposto os comportamentos aos quais correspondem as menores probabilidades de ocorrência, obtendo-se assim um modelo estruturalmente mais simples. As expressões deduzidas neste capítulo e associadas às respectivas probabilidades de ocorrência podem (e devem) ser utilizadas para auxiliar esta tarefa. A segunda fase, passa por uma separação do modelo ao nível da camada de ligação de dados, obtendo-se assim dois modelos que descrevem o comportamento respectivamente acima e abaixo desta camada.

O modelo superior representa o tráfego (mensagens) gerado por cada nó da rede, enquanto o inferior descreve o mecanismo de acesso ao meio (passagem do *token*). Como estes modelos não são independentes, é necessário arranjar um forma de os desacoplar.

Ao nível do modelo superior e partindo do modelo de simulação, é possível caracterizar o tráfego gerado por cada nó e representar este comportamento através modelo capaz de permitir uma solução analítica. Este modelo pode, por exemplo, ser obtido através da técnica de expansão por fases da função distribuição. Embora se trate de uma aproximação, a natureza estocástica do problema potencia que os resultados obtidos tenham um bom nível de rigor.

Ao nível do modelo inferior (passagem do *token*), a obtenção de um modelo analítico é mais simples de realizar. Admitindo-se por exemplo, que o *token* está durante um tempo constante em cada nó, então a solução é imediata (só se tem uma transição determinística habilitada de cada vez). Contudo, esta pode, eventualmente, ser uma aproximação um pouco grosseira, pelo que é necessário utilizar uma abordagem semelhante à proposta para o modelo anterior.

Esta abordagem sofre contudo de um problema intrínseco: é necessário executar o modelo de simulação previamente para estimar os parâmetros do modelo analítico. A solução deste problema, passa por uma tentativa de caracterização das funções distribuição (mensagens e passagem do *token*) para um conjunto representativo de cenários. Ou seja, ter um único modelo em que seja possível estabelecer uma relação funcional entre os seus parâmetros e

a situação avaliada, sem contudo ter que executar o modelo de simulação. Pretende-se assim que, partindo de um cenário qualquer, seja possível determinar previamente (de forma aproximada, é claro) o seu comportamento estocástico, com base em cenários tipo pré-avaliados.

No sentido de permitir uma caracterização mais eficiente destes cenários, o autor está presentemente envolvido num projecto de investigação onde está a ser implementada uma estrutura para a injeção de falhas em redes Profibus-DP. Isto vai permitir obter os parâmetros destes modelos de forma mais exacta e sem tem que recorrer directamente ao modelo de simulação;

- ♦ O modelo desenvolvido para o CAN apresenta características que potenciam obtenção de uma solução analítica. Embora da discussão realizada no capítulo 6 tenha ficado patente a dificuldade de obter uma solução deste tipo, a utilização de uma outra abordagem permite antever essa possibilidade.

Esta abordagem passa pela utilização de técnicas iterativas para a obtenção da solução, através da utilização da ferramenta SPNP [HTT02]. Dada a natureza simétrica do modelo da transmissão de uma mensagem aliada à sua simplicidade, é possível desenvolver um modelo (com características semelhantes) que possa ser utilizado desta forma. Embora a ferramenta proposta não suporte a utilização de transições determinísticas, implementa de forma automática técnicas de expansão por fases que permitem realizar a respectiva aproximação. Note-se, que devido à natureza iterativa da solução em que cada modelo é resolvido de forma independente, não se coloca o problema da explosão do espaço de estados.

Esta abordagem foi também utilizada com sucesso ao nível dos sistemas de comunicação baseados em *polling* [Hav99], o que permite assim antever boas possibilidades de sucesso neste caso concreto.

Num contexto próximo, seria interessante incluir outros aspectos como por exemplo o comportamento dos contadores TEC e REC e das mensagens com erros não detectados. Os contadores poderiam ser incluídos de forma indirecta, através de um conjunto de transições exponenciais que representam o tempo médio (esperado) entre estadias nos vários estados dos contadores. Estes tempos podem ser directamente obtidos a partir de outros estudos já realizados. As mensagens com erros não detectados, podem ser incluídas a partir de um abordagem probabilística ao problema.

Num contexto não directamente associado aos modelos propostos, existem perspectivas muito interessantes ao nível da avaliação do comportamento de redes utilizadas em veículos automóveis. Sendo estas redes um componente central dos sistemas de segurança crítica utilizados a este nível, torna-se imprescindível analisar o seu comportamento na presença de falhas, principalmente do tipo

transitório e que ocorrem durante as comunicações. O carácter periódico associado ao funcionamento de muitas delas (ex. *FlexRay* [Fle04] e *Byteflight* [PBG99]), permite antever a possibilidade de dispor de soluções analíticas. Nos casos mais complexos poder-se-á explorar as soluções baseadas em simulação, mas agora utilizando as disponibilidades de simulação distribuída oferecidas pelas várias ferramentas de modelação disponíveis.

Esta página foi intencionalmente deixada em branco

Anexo 1

Distribuição Bernoulli	
Parâmetros	$p, \quad 0 \leq p \leq 1$ – probabilidade de sucesso
Espaço amostral	$S_X = \{0, 1\}$
Função de massa	$f_X(k) = \begin{cases} 1-p & \text{se } k=0 \\ p & \text{se } k=1 \\ 0 & \text{restantes} \end{cases}$
Função distribuição	$F_X(k) = \begin{cases} 1-p & \text{se } k < 0 \\ p & \text{se } k < 1 \\ 1 & \text{restantes} \end{cases}$
Momentos	$E[X] = p \quad V[X] = p \cdot (1-p)$
Comentários	A variável aleatória X pode ser interpretada como a indicação do sucesso de um determinado evento.

Distribuição Binomial	
Parâmetros	$p, \quad 0 \leq p \leq 1$ – probabilidade de sucesso $n, \quad n \geq 0$ – número de experiências
Espaço amostral	$S_X = \{0, 1, \dots, n\}$
Função de massa	$f_X(k) = \binom{n}{k} \cdot p^k \cdot (1-p)^{n-k}$
Função distribuição	$F_X(k) = \sum_{i=0}^k \binom{n}{i} \cdot p^i \cdot (1-p)^{n-i}$
Momentos	$E[X] = n \cdot p \quad V[X] = n \cdot p \cdot (1-p)$
Comentários	A variável aleatória X pode ser interpretada como o número de sucessos em n experiências de Bernoulli.

Anexo 1

Distribuição Geométrica	
Parâmetros	$p, \quad 0 \leq p \leq 1$ – probabilidade de sucesso
Espaço amostral	$S_X = \{0, 1, \dots\}$
Função de massa	$f_X(k) = (1-p)^{n-k} \cdot p$
Função distribuição	$F_X(k) = 1 - (1-p)^k$
Momentos	$E[X] = \frac{1}{p} \quad V[X] = \frac{1-p}{p^2}$
Comentários	A variável aleatória X pode ser interpretada como o número de insucessos antes do primeiro sucesso numa sequência de experiências de Bernoulli.

Distribuição de Poisson	
Parâmetros	$\lambda, \quad \lambda \geq 0$ – taxa de ocorrência do evento por unidade de tempo
Espaço amostral	$S_X = \{0, 1, \dots\}$
Função de massa	$f_X(k) = \frac{\lambda^k}{k!} \cdot e^{-\lambda}$
Função distribuição	$F_X(k) = \sum_{i=0}^k \frac{\lambda^i}{i!} \cdot e^{-\lambda}$
Momentos	$E[X] = \lambda \quad V[X] = \lambda$
Comentários	A variável aleatória X pode ser interpretada como o número de eventos que ocorrem numa unidade de tempo, quando o intervalo de tempo entre eventos sucessivos é definido por uma distribuição de exponencial com parâmetro λ .

Distribuição Uniforme	
Parâmetros	$a, \quad a \geq 0$ – limite inferior $b, \quad b \geq a$ – limite superior
Função densidade	$f_X(t) = \begin{cases} \frac{1}{b-a} & a \leq t \leq b \\ 0 & \text{restantes} \end{cases}$
Função distribuição	$F_X(t) = \begin{cases} 0 & \text{se } 0 \leq t < a \\ \frac{t-a}{b-a} & \text{se } a \leq t < b \\ 1 & \text{se } t \geq b \end{cases}$
Momentos	$E[X] = \frac{a+b}{2} \quad V[X] = \frac{(b-a)^2}{12}$

Distribuição Exponencial	
Parâmetros	λ , $\lambda \geq 0$ – parâmetro de escala
Função densidade	$f_X(t) = \lambda \cdot e^{-\lambda t}$
Função distribuição	$F_X(t) = 1 - e^{-\lambda t}$
Momentos	$E[X] = \frac{1}{\lambda}$ $V[X] = \frac{1}{\lambda^2}$
Comentários	Quando o intervalo de tempo entre eventos sucessivos possui uma distribuição exponencial com taxa λ , o número de eventos que ocorrem num intervalo $]0, t]$ tem uma função distribuição de Poisson com parâmetro $\lambda \cdot t$.

Distribuição de Erlang	
Parâmetros	λ , $\lambda \geq 0$ – parâmetro de escala r , $r > 0$ – parâmetro de aspecto
Função densidade	$f_X(t) = \frac{t^{r-1} \cdot \lambda^r}{(r-1)!} \cdot e^{-\lambda t}$
Função distribuição	$F_X(t) = 1 - e^{-\lambda t} \cdot \sum_{i=0}^{r-1} \frac{(t \cdot \lambda)^i}{i!}$
Momentos	$E[X] = \frac{r}{\lambda}$ $V[X] = \frac{r}{\lambda^2}$
Comentários	A variável aleatória X pode ser interpretada como a soma de r funções distribuição exponenciais independentes de parâmetro λ . Quando r tende para ∞ a distribuição degenera numa distribuição determinística.

Distribuição Hipo-Exponencial	
Parâmetros	$\lambda_1, \lambda_2, \dots, \lambda_r$, $\lambda_i \geq 0$ – parâmetros de escala
Função densidade	$f_X(t) = \sum_{i=1}^r a_i \cdot \lambda_i \cdot e^{-\lambda_i t}$ $a_i = \prod_{\substack{j=1 \\ j \neq i}}^r \frac{\lambda_j}{\lambda_j - \lambda_i}$
Função distribuição	$F_X(t) = 1 - \sum_{i=1}^r a_i \cdot e^{-\lambda_i t}$
Momentos	$E[X] = \sum_{i=1}^r \frac{r}{\lambda_i}$ $V[X] = \sum_{i=1}^r \frac{r}{\lambda_i^2}$
Comentários	A variável aleatória X pode ser interpretada como a soma de r funções distribuição exponenciais independentes de parâmetro λ_i .

Distribuição Hiper-Exponencial	
Parâmetros	$\lambda_1, \lambda_2, \dots, \lambda_r, \lambda_i \geq 0$ – parâmetros de escala $\alpha_1, \alpha_2, \dots, \alpha_r, \alpha_i > 0$ $\sum_{i=1}^r \alpha_i = 1$ – parâmetros de aspecto
Função densidade	$f_X(t) = \sum_{i=1}^r \alpha_i \cdot \lambda_i \cdot e^{-\lambda_i t}$
Função distribuição	$F_X(t) = \sum_{i=1}^r \alpha_i \cdot (1 - e^{-\lambda_i t})$
Momentos	$E[X] = \sum_{i=1}^r \frac{\alpha_i}{\lambda_i} \quad V[X] = 2 \cdot \sum_{i=1}^r \frac{\alpha_i}{\lambda_i^2} - \left(\sum_{i=1}^r \frac{\alpha_i}{\lambda_i} \right)^2$
Comentários	A variável aleatória X pode ser interpretada como a soma de r funções distribuição exponenciais independentes de parâmetro λ_i , pesadas através das probabilidades α_i .

Distribuição de Weibull	
Parâmetros	$\lambda, \lambda \geq 0$ – parâmetro de escala $\alpha, \alpha > 0$ – parâmetro de aspecto
Função densidade	$f_X(t) = \alpha \cdot \lambda \cdot t^{\alpha-1} \cdot e^{-\lambda t^\alpha}$
Função distribuição	$F_X(t) = 1 - e^{-\lambda t^\alpha}$
Momentos	$E[X] = \left(\frac{1}{\lambda} \right)^{\left(\frac{1}{\alpha} \right)} \cdot \Gamma\left(1 + \frac{1}{\alpha}\right) \quad V[X] = \left(\frac{1}{\lambda^2} \right)^{\left(\frac{2}{\alpha} \right)} \cdot \left[\Gamma\left(1 + \frac{2}{\alpha}\right) - \left[\Gamma\left(1 + \frac{1}{\alpha}\right) \right]^2 \right]$ $\Gamma(\alpha) = \int_0^\infty x^{\alpha-1} \cdot e^{-x} dx \quad \alpha > 0$

Distribuição t-Student	
Parâmetros	$n, n > 0$ – graus de liberdade
Função densidade	$f_X(t) = \frac{\Gamma((n+1)/2)}{\sqrt{n \cdot \pi} \cdot \Gamma(n/2)} \cdot \left(1 + \frac{t^2}{n}\right)^{-(n+1)/2}$
Função distribuição	$F_X(t) = \begin{cases} 1 - \frac{1}{2} \cdot I_x(n/2, 1/2) & t > 0 \\ \frac{1}{2} \cdot I_x(n/2, 1/2) & t \leq 0 \end{cases} \quad x = \frac{1}{1 + t^2/n} \quad I_x(a, b) = \frac{B_x(a, b)}{B(a, b)}$ $B_x(a, b) = \int_0^x t^{(a-1)} \cdot (1-t)^{(b-1)} dt$

Anexo 2

Sendo o principal objectivo deste estudo realizar uma análise genérica, procurou-se definir os vários parâmetros do modelo através de critérios próximos da realidade de operação da rede:

- ♦ $n=15$, correspondendo a uma rede com uma dimensão apreciável;
- ♦ $m=3$, correspondendo a uma redundância razoável;
- ♦ $\lambda_n = \begin{cases} 1 \text{ ano} \\ 8 \text{ meses} \end{cases}$, tendo em conta que os nós são os elementos mais complexos do sistema a sua taxa de avarias será necessariamente a maior. Neste caso admitiram-se dois valores típicos por forma a analisar qual a importância deste parâmetro;
- ♦ $\lambda_b = \begin{cases} 1.5 \text{ anos} \\ 3 \text{ anos} \end{cases}$, o meio físico transmissão e os *transceivers* dos nós são relativamente simples (especialmente o primeiro), consequentemente foi admitida uma menor taxa de avarias (em comparação com λ_n). Também aqui foram admitidos dois valores pelas mesmas razões anteriores;
- ♦ $\mu=2$ horas , tendo em conta o tipo de componentes utilizados neste tipo de rede a sua reparação e/ou substituição pode ser considerada rápida;
- ♦ $\mu_g=6$ horas , este valor foi assumido tendo em conta uma reparação e/ou substituição de vários componentes da rede;
- ♦ $c_n \in [0.5 \dots 1.0]$, a consideração deste intervalo foi baseada no princípio de que existem ou que se pretendem introduzir mecanismos de tolerância a falhas minimamente eficazes.

Devido às características dos modelos avaliados, alguns dos parâmetros definidos, nomeadamente a dimensão da rede e a redundância entre nós, foram ligeiramente sobre-dimensionados. O objectivo é apenas fazer realçar (numericamente) os vários factores associados os mesmos, caso contrário a análise poderia ser difícil realização. Desta forma, os parâmetros do modelo devem ser também interpretados neste contexto.

A confiança no funcionamento da rede foi analisada através de duas medidas: a indisponibilidade assintótica ($\bar{A}$) e o tempo médio até falhar ($MTTF$). Tendo em conta a dependência existente entre os valores destas medidas e os parâmetros utilizados, procurou-se apresentar os resultados de forma normalizada através da definição de medidas relativas, utilizando para isso várias referências. Estas últimas foram escolhidas para representar as piores situações ($\bar{A}$ elevados e $MTTF$ reduzidos), implicando desta forma uma avaliação relativa que resulta na obtenção de resultados optimistas. O objectivo foi assim assegurar, que em condições normais de funcionamento da rede as medidas relativas nunca vão ultrapassar estes valores.

$$\begin{aligned}\bar{A}_{\text{Relativa}} &= \frac{\bar{A} - \bar{A}_{\text{Caso mais desfavorável}}}{\bar{A}_{\text{Caso mais desfavorável}}} * 100\% \\ \bar{A}_{\text{Comparativa}} &= \frac{\bar{A}_{\text{Caso 1}} - \bar{A}_{\text{Caso 2}}}{\bar{A}_{\text{Caso 2}}} * 100\% \\ MTTF_{\text{Relativo}} &= \frac{MTTF - MTTF_{\text{Caso mais desfavorável}}}{MTTF_{\text{Caso mais desfavorável}}} * 100\% \\ MTTF_{\text{Comparativo}} &= \frac{MTTF_{\text{Caso 1}} - MTTF_{\text{Caso 2}}}{MTTF_{\text{Caso 2}}} * 100\% \\ Erro_{\text{Relativo}} &= \left| \frac{Valor_{\text{calculado}} - Valor_{\text{aproximado}}}{Valor_{\text{calculado}}} \right| * 100\%\end{aligned}\tag{1}$$

Além destes, e como forma de apresentar uma ordem de grandeza das medidas em causa, são também apresentados os respectivos valores absolutos.

A2.1 Modelo mR : Conclusões

De forma global (Fig. A2.1, A2.2, A2.3 e A2.4) é possível concluir que para as medidas realizadas ($\bar{A}$ e $MTTF$) a expressão obtida para λ_e constitui uma óptima aproximação.

Nos gráficos relativos foi considerado que o caso mais desfavorável correspondia à situação em que $c_n=0$ (total ausência de mecanismos de tolerância a falhas), mantendo todos os restantes parâmetros constantes, com excepção daqueles que se pretendem analisar.

A análise da evolução de $\bar{A} = f(c_n)$ (Fig. A2.1) e de $MTTF = f(c_n)$ (Fig. A2.2) com λ_b e λ_n como parâmetros fixos, permite concluir em relação aos seguintes parâmetros¹⁰⁷:

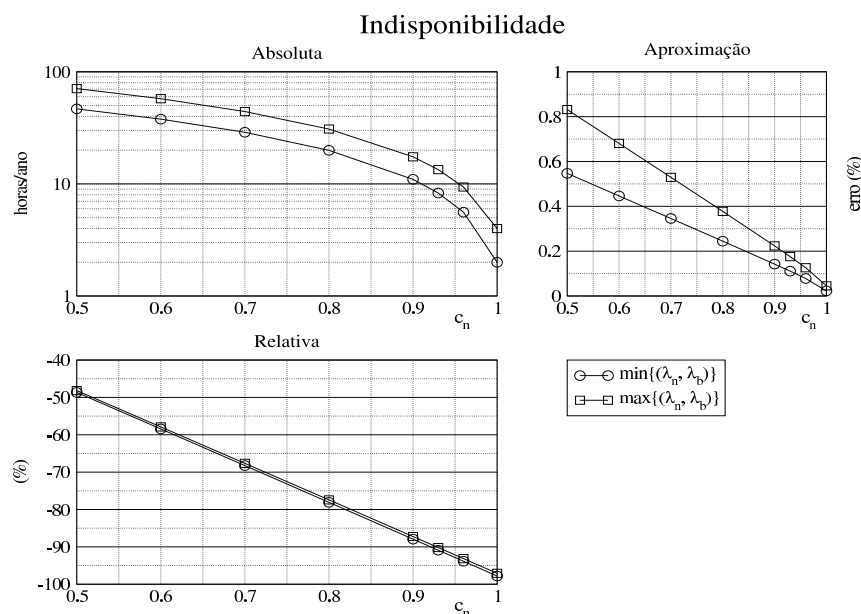


Figura A2.1 - Modelo mR: indisponibilidade (c_n)

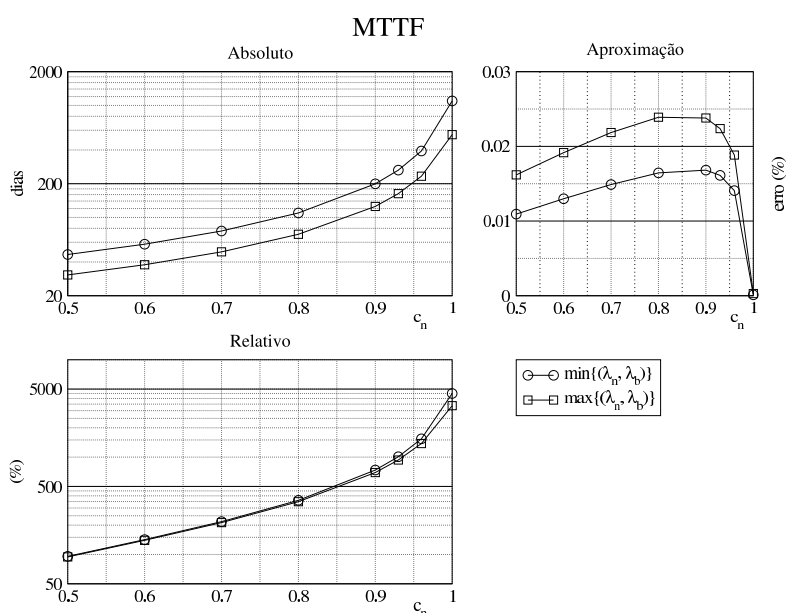


Figura A2.2 - Modelo mR: MTTF (c_n)

107As referências utilizadas no gráfico relativo foram: ($c_n=0$, $\min\{(\lambda_n, \lambda_b)\}$) e ($c_n=0$, $\max\{(\lambda_n, \lambda_b)\}$), respectivamente para $\min\{(\lambda_n, \lambda_b)\}$ e $\max\{(\lambda_n, \lambda_b)\}$.

- ♦ c_n : o factor de cobertura possui um papel fundamental na evolução das medidas obtidas. Para $c_n \in [0.5 \dots 0.9]$ a evolução é praticamente linear. Contudo, para $c_n \in [0.9 \dots 1.0]$ a evolução é mais significativa, chegando a valores próximos de uma ordem de grandeza. O primeiro intervalo corresponde ao caso típico de $\lambda_n \gg \lambda_b$, em que as medidas dependem quase exclusivamente do termo $\bar{c}_n n \lambda_n$ (daí o seu comportamento quase linear). No segundo intervalo, como o valor de c_n é muito elevado o termo anterior é fortemente reduzido, sendo o valor das medidas apenas limitado por λ_b . É importante verificar que para uma determinada variação percentual de c_n (admitindo que $c_n \geq 0.5$) a variação de $\bar{c}_n$ é muito superior (ex: $c_n = 0.7 \rightarrow 0.8 \equiv \Delta \approx 14\% \Leftrightarrow \bar{c}_n = 0.3 \rightarrow 0.2 \equiv \Delta \approx -50\%$!), provocando assim um efeito de “ampliação” do parâmetro $\bar{c}_n$, efeito este que é tanto maior, quanto maior o valor de c_n ;
- ♦ (λ_b, λ_n) : a sua influência é ligeira, sendo atribuída essencialmente à variação de λ_n , dado que este último possui um peso mais elevado, quer em termos absolutos, quer em termos de contribuição para o valor final de λ_e .

A análise da evolução de $\bar{A} = f(\lambda_n)$ (Fig. A2.3) com λ_b e c_n como parâmetros fixos, permite concluir em relação aos seguintes parâmetros¹⁰⁸:

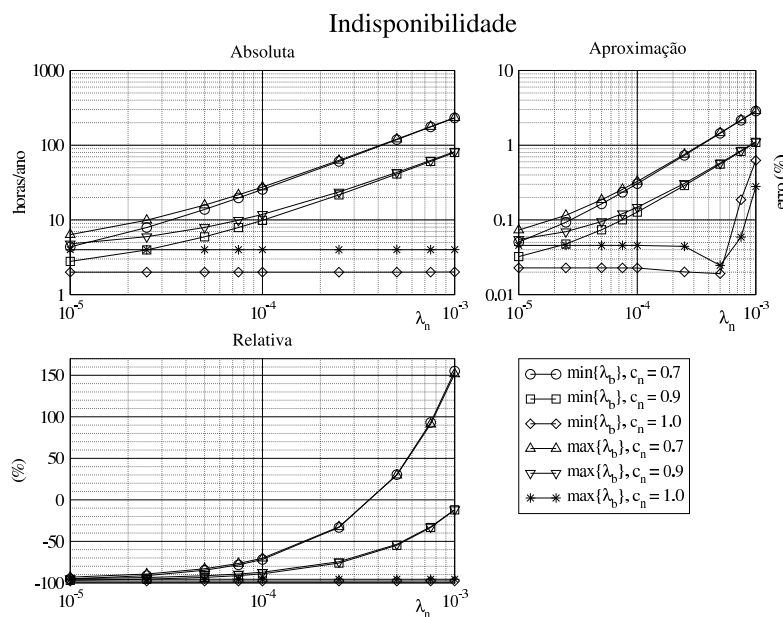


Figura A2.3 -Modelo mR: indisponibilidade (λ_n)

¹⁰⁸As referências utilizadas no gráfico relativo foram: $(c_n = 0, \min\{\lambda_b\}, \min\{\lambda_n\})$ e $(c_n = 0, \max\{\lambda_b\}, \min\{\lambda_n\})$, respectivamente para $\min\{\lambda_b\}$ e $\max\{\lambda_b\}$.

- ♦ λ_n : para valores reduzidos, a sua influência é ligeira e comparável aos restantes parâmetros. Isto corresponde a uma situação de equilíbrio entre λ_b e $\bar{c}_n n \lambda_n$. Para valores médios/elevados a sua influência é preponderante, correspondendo a uma situação de total predominância de $\bar{c}_n n \lambda_n$ sobre λ_b , implicando assim que as medidas sejam apenas condicionadas por λ_n e c_n ;
- ♦ (c_n, λ_b) : dos dois parâmetros, o mais importante é sem dúvida c_n . A sua influência acentua-se para valores de λ_n médios/elevados, resultando do facto de que pequenas variações de c_n (à qual correspondem variações grandes de $\bar{c}_n$) implicarem variações significativas nas medidas.

A análise da evolução de $\bar{A} = f(\lambda_b)$ (Fig. A2.4) com λ_b e c_n como parâmetros fixos, permite concluir em relação aos seguintes parâmetros¹⁰⁹:

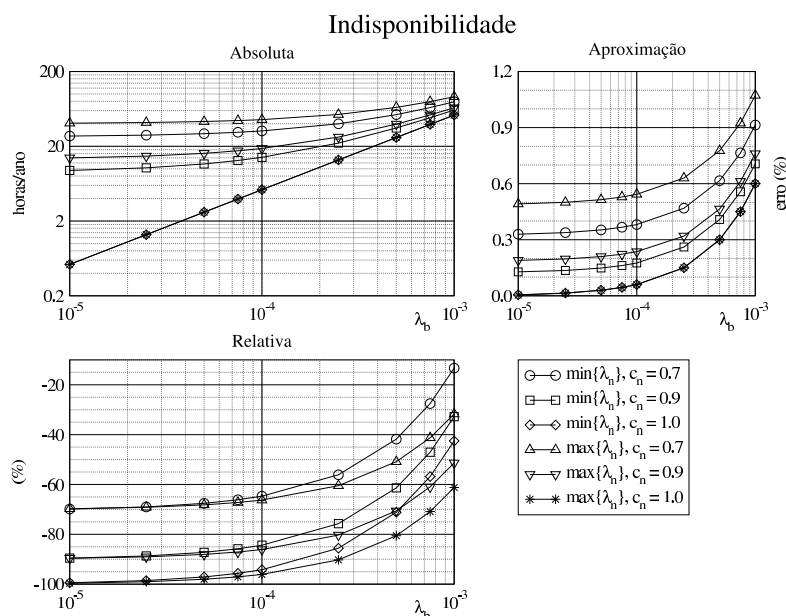


Figura A2.4 - Modelo mR: indisponibilidade (λ_b)

- ♦ λ_b : para valores reduzidos/médios a sua influência é praticamente nula, correspondendo a uma situação de total predominância de $\bar{c}_n n \lambda_n$ sobre λ_b . Isto implica que apenas λ_n e c_n condicionem as medidas. Para valores elevados, existe um certo equilíbrio entre λ_b e $\bar{c}_n n \lambda_n$, correspondendo assim uma influência proporcional de todos os parâmetros;
- ♦ (c_n, λ_n) : a sua influência faz-se sentir, essencialmente, para valores reduzidos/baixos de λ_b . Neste caso o efeito de c_n é superior ao de λ_n devido essencialmente ao efeito de “ampliação” de $\bar{c}_n$ ser superior à variação de λ_n ;

¹⁰⁹As referências utilizadas no gráfico relativo foram: ($c_n = 0$, $\min\{\lambda_b\}$, $\min\{\lambda_n\}$) e ($c_n = 0$, $\min\{\lambda_b\}$, $\max\{\lambda_n\}$), respectivamente para $\min\{\lambda_n\}$ e $\max\{\lambda_n\}$.

Comparando, de forma geral, a evolução de $\bar{A}=f(\lambda_n)$ com $\bar{A}=f(\lambda_b)$, conclui-se que esta última é muito mais ligeira (especialmente para $c_n \leq 0.9$) do que a primeira, reforçando assim a importância do factor $\bar{c}_n n \lambda_n$ em relação a quaisquer outros.

A2.2 Modelo mD : Conclusões

De forma global (Fig. A2.5, A2.6, A2.7 e A2.8) é possível concluir que para as medidas realizadas ($\bar{A}$ e $MTTF$) a expressão obtida para λ_e constitui uma óptima aproximação.

Nos gráficos relativos foi considerado que o caso mais desfavorável correspondia à situação em que $c_n=0$ (total ausência de mecanismos de tolerância a falhas), mantendo todos os restantes parâmetros constantes, com excepção para aqueles que se pretendem analisar. Os gráficos comparativos apresentados, referem-se à comparação com o modelo **mR** para o mesmo conjunto de parâmetros.

A análise da evolução comparativa de $\bar{A}=f(c_n)$ (Fig. A2.5) e $MTTF=f(c_n)$ (Fig. A2.6) com λ_b e λ_n como parâmetros fixos, permite concluir em relação aos seguintes parâmetros¹¹⁰:

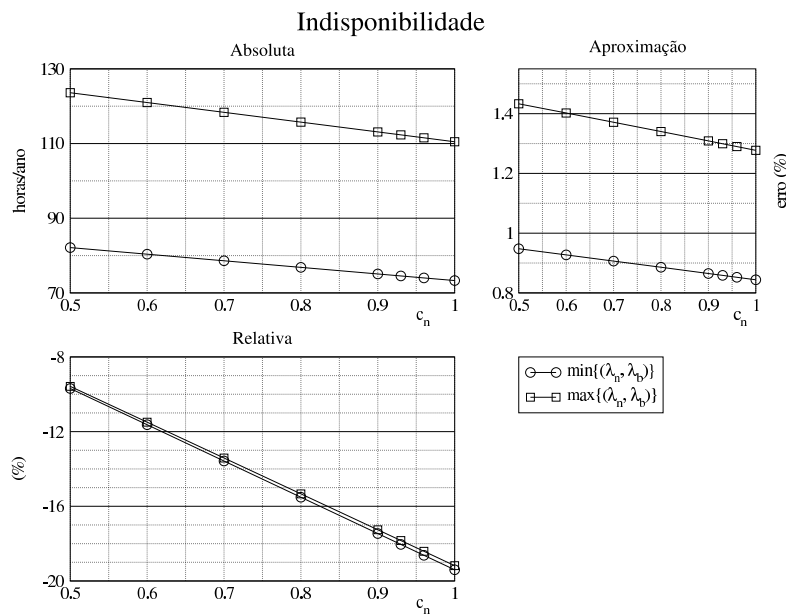


Figura A2.5 - Modelo mD: indisponibilidade (c_n)

¹¹⁰As referências utilizadas no gráfico relativo foram: ($c_n=0$, $\min\{(\lambda_n, \lambda_b)\}$) e ($c_n=0$, $\max\{(\lambda_n, \lambda_b)\}$), respectivamente para $\min\{(\lambda_n, \lambda_b)\}$ e $\max\{(\lambda_n, \lambda_b)\}$.

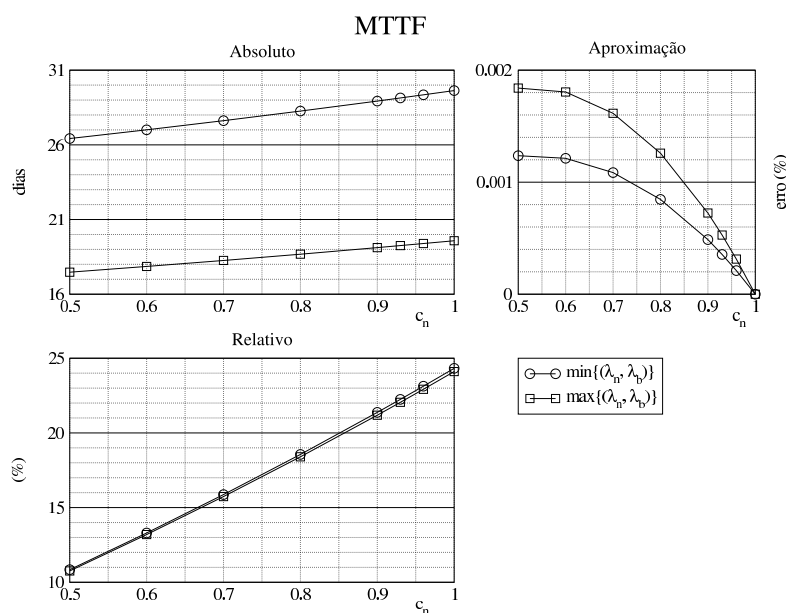


Figura A2.6 - Modelo mD: MTTF (c_n)

- ♦ c_n : a sua influência é reduzida, dado que da totalidade das falhas existentes apenas uma pequena parte das mesmas ($\bar{c}_n m \lambda_n$) se encontra coberta;
- ♦ (λ_b, λ_n) : a sua influência é reduzida/média, sendo atribuída essencialmente a λ_n . Este efeito deve-se, quer à amplitude de variação de λ_n face a λ_b , quer à fracção importante de falhas que não são cobertas ($(n-m)\lambda_n$).

A análise da evolução de $\bar{A} = f(\lambda_n)$ (Fig. A2.7) com λ_b e c_n como parâmetros fixos, permite concluir em relação aos seguintes parâmetros¹¹¹:

- ♦ λ_n : a sua influência é muito forte, traduzindo-se praticamente através de uma quase total dependência. Este efeito resulta do facto de que a maioria das falhas ($(n-m)\lambda_n$ e $\bar{c}_n m \lambda_n$) serem uma função de λ_n , e de que os termos anteriores serem largamente dominantes face aos restantes (λ_b);
- ♦ (c_n, λ_b) : a sua influência é muito reduzida, sendo atribuída essencialmente a c_n . Este efeito deve-se à existência de um conjunto, reduzido, de falhas cobertas ($\bar{c}_n m \lambda_n$);

¹¹¹As referências utilizadas no gráfico relativo foram: ($c_n=0$, $\min\{\lambda_b\}$, $\min\{\lambda_n\}$) e ($c_n=0$, $\max\{\lambda_b\}$, $\min\{\lambda_n\}$), respectivamente para $\min\{\lambda_b\}$ e $\max\{\lambda_b\}$.

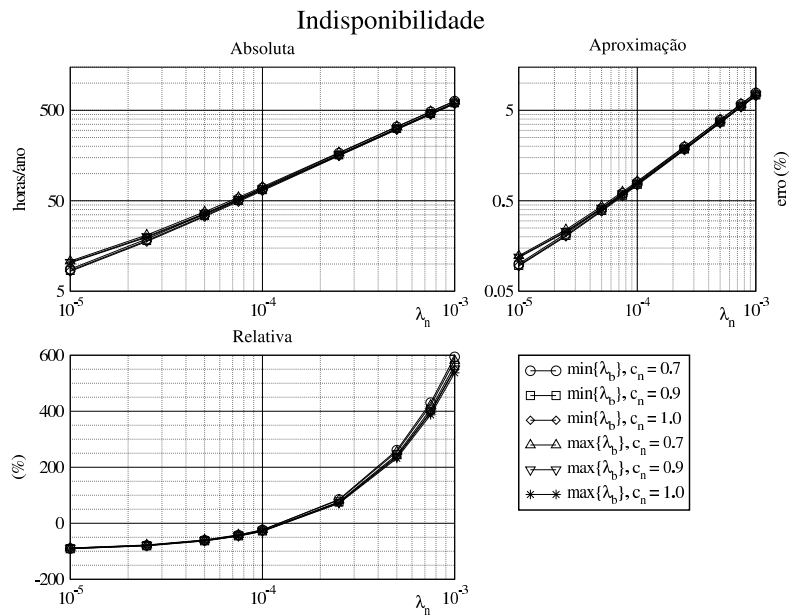


Figura A2.7 - Modelo mD: indisponibilidade (λ_n)

A análise da evolução de $\bar{A} = f(\lambda_b)$ (Fig. A2.8) com λ_b e c_n como parâmetros fixos, permite concluir em relação aos seguintes parâmetros¹¹²:

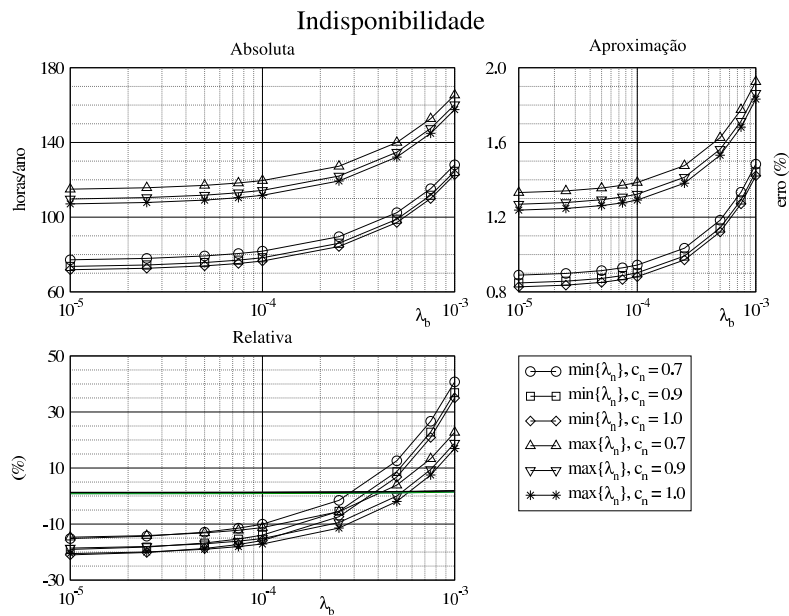


Figura A2.8 - Modelo mD: indisponibilidade (λ_b)

¹¹²As referências utilizadas no gráfico relativo foram: ($c_n=0$, $\min\{\lambda_b\}$, $\min\{\lambda_n\}$) e ($c_n=0$, $\min\{\lambda_b\}$, $\max\{\lambda_n\}$), respectivamente para $\min\{\lambda_n\}$ e $\max\{\lambda_n\}$.

- ♦ λ_b : para valores reduzidos/médios a sua influência é praticamente nula, dado que a importância deste parâmetro é muito reduzida face aos restantes. Para valores elevados torna-se mais importante, correspondendo a um equilíbrio/predominância de λ_b sobre os restantes;
- ♦ (c_n, λ_n) : a sua influência é significativa, fazendo-se sentir em toda a gama de valores de λ_b . Dos dois, o mais importante é sem dúvida λ_n , devido à existência de um parte significativa de falhas não cobertas;
- ♦ Comparando este modelo com o anterior (modelo **mR**), conclui-se que este apresenta um comportamento de pior qualidade, devido essencialmente ao facto de que o número de nós em que falhas podem ser cobertas neste caso é de apenas m , enquanto que no anterior é de n .

A2.3 Modelo mRD : Conclusões

De forma global (Fig. A2.10, A2.11, A2.12 e A2.13) é possível concluir que para as medidas realizadas ($\bar{A}$ e $MTTF$), a expressão obtida para λ_e constitui uma óptima aproximação.

A análise da evolução de $\bar{A} = f(c_b)$ (Fig. A2.9) com λ_n , λ_b e c_n como parâmetros fixos, permite concluir que o efeito de c_b é, na maioria dos casos ligeiro. Este comportamento é previsível, dado que c_b só afecta λ_b , cujo valor absoluto é, na esmagadora maioria dos casos, bastante inferior às restantes taxas. Este parâmetro só assume especial importância no caso de c_n ser muito elevado ($c_n \gg 0.9$), dado que nesta situação a taxa de avarias é apenas limitada por λ_b , cujo o efeito pode ainda ser minorado através do factor de cobertura do meio físico (c_b).

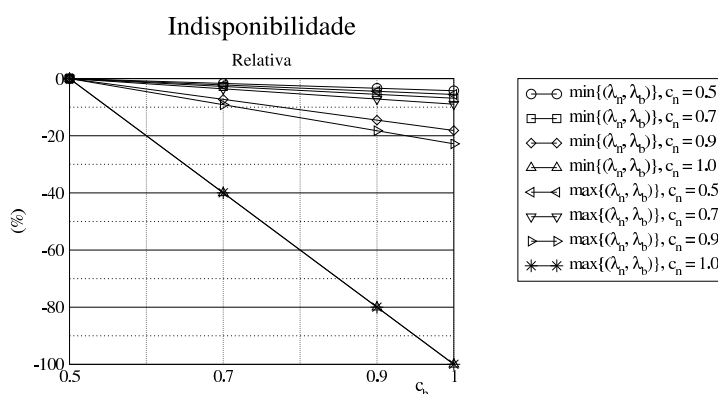


Figura A2.9 - Modelo mRD: indisponibilidade (c_b)

Ao contrário das secções anteriores onde era analisado o comportamento individual de cada modelo, as conclusões relativas a este modelo foram obtidas através da comparação dos respectivos resultados com o modelo **mR**. Pretende-se assim avaliar os efeitos da introdução de um meio físico redundante. Neste sentido, e partindo das conclusões já obtidas para c_b , assumiu-se na avaliação comparativa que $c_b=0.9$. Este pressuposto tem apenas como objectivo, permitir que a comparação entre os modelo **mRD** e **mR** se realize de forma o mais simples possível. Além disso, o valor considerado é de facto pessimista (os valores serão na prática tipicamente maiores), resultando assim numa avaliação comparativa com a mesma natureza.

Os gráficos comparativos apresentados, referem-se à comparação do modelo **mRD** (caso 1) com o modelo **mR** (caso 2), mantendo os mesmos valores dos restantes parâmetros.

A análise da evolução comparativa de $\bar{A}=f(c_n)$ (Fig. A2.10) e $MTTF=f(c_n)$ (Fig. A2.11) com λ_b e λ_n como parâmetros fixos, permite concluir em relação aos seguintes parâmetros:

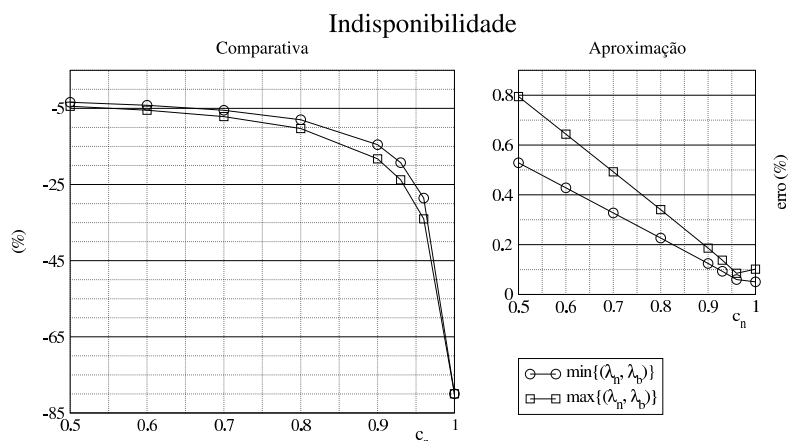


Figura A2.10 - Modelo **mRD** vs. **mR**: indisponibilidade (c_n)

- ♦ c_n : a introdução de um segundo meio físico de transmissão não se traduz por um aumento significativo da qualidade de funcionamento (tipicamente um aumento até 10/15 %), excepto para os casos em que $c_n > 0.96$ em que a evolução é muito significativa. Comparando os modelos, verifica-se que a diferença se situa, fundamentalmente, nos termos λ_b e $2\bar{c}_b\lambda_b$, com este último sempre inferior ao primeiro (desde que $c_b > 0.5$). É simples de verificar,

que só para valores muito elevados de c_n é que estes termos se tornam significativos, impondo o parâmetro c_b a diferença entre ambos. Assim, e nas condições anteriores, só é justificada a introdução de um segundo meio de transmissão se $c_b > 0.5$;

- ♦ (λ_b, λ_n) : a influência destes factores é reduzida, fazendo-se sentir apenas para valores de c_n elevados. Isto corresponde à situação em que estes parâmetros se tornam, ainda que ligeiramente, significativos.

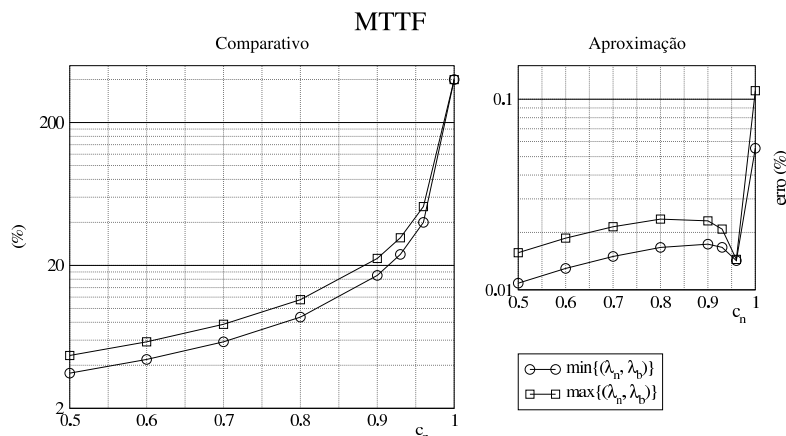


Figura A2.11 - Modelo mRD vs. mR: MTTF (c_n)

A análise da evolução comparativa de $\bar{A} = f(\lambda_n)$ (Fig. A2.12) com λ_b e c_n como parâmetros fixos, permite concluir em relação aos seguintes parâmetros:

- ♦ λ_n : a sua influência só é significativa para valores reduzidos/médios, correspondendo à situação em que os primeiros termos das expressões, λ_b e $2\bar{c}_b\lambda_b$ respectivamente, possuem um papel mais importante face aos restantes. Tendo em conta que o termo em **mRD** possui um valor mais reduzido, verifica-se uma melhoria da qualidade medidas na rede dual. Para valores elevados as expressões são equivalentes ($\approx \bar{c}_n n \lambda_n$), implicando assim a ausência de diferenças significativas;
- ♦ (c_n, λ_b) : a influência destes parâmetros é importante, especialmente para valores reduzidos/médios de λ_n , sendo o efeito de c_n dominante face a λ_b . Este último, resulta do facto de c_n surgir, em ambos os casos associado a λ_n ($\bar{c}_n n \lambda_n$) e do consequente efeito de “ampliação” de $\bar{c}_n$.

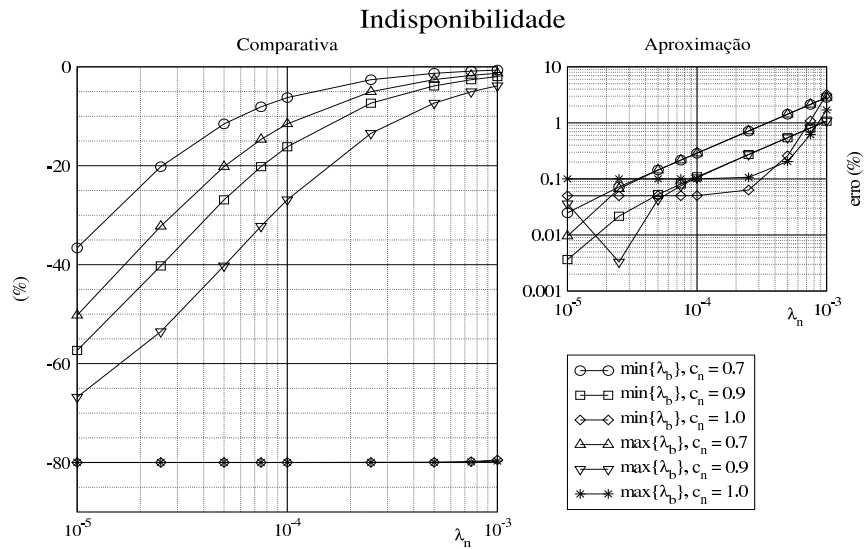


Figura A2.12 - Modelo mRD vs. mR: indisponibilidade (λ_n)

A análise da evolução comparativa de $\bar{A} = f(\lambda_b)$ (Fig. A2.13) com λ_n e c_n como parâmetros fixos, permite concluir em relação aos seguintes parâmetros:

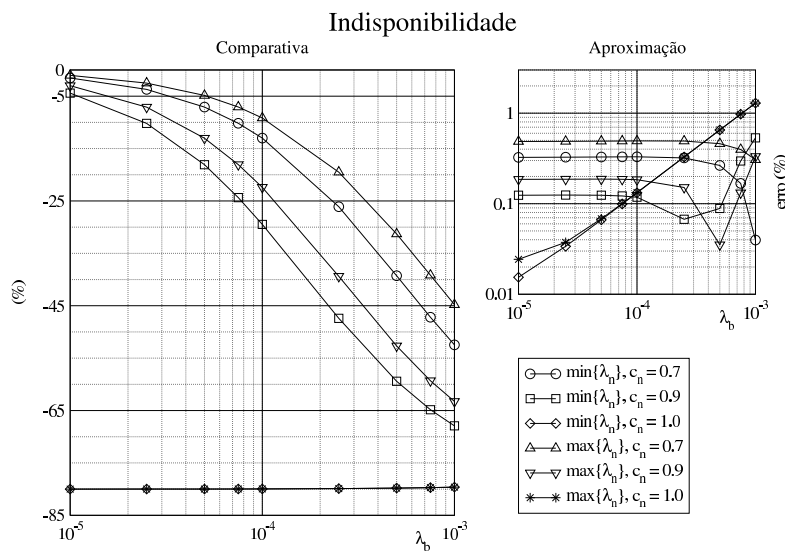


Figura A2.13 - Modelo mRD vs. mR: indisponibilidade (λ_b)

- λ_b : a sua influência só é significativa para valores médios/elevados. Isto corresponde à situação em que os primeiros termos das expressões, λ_b e $2\bar{c}_b\lambda_b$ respectivamente, possuem um papel mais importante face aos restantes. Tendo em conta que o termo em **mRD** possui um valor mais

reduzido, verifica-se uma melhoria da qualidade medidas na rede dual. Para valores reduzidos, as expressões são equivalentes ($\approx \bar{c}_n n \lambda_n$), implicando assim a ausência de diferenças significativas;

- ♦ (c_n, λ_n) : a influência destes parâmetros é importante, especialmente para valores médios/elevados de λ_b , sendo o efeito de c_n dominante face a λ_b . Este último, resulta do facto de c_n surgir em ambos os casos associado a λ_n ($\bar{c}_n n \lambda_n$) e do consequente efeito de “ampliação” de $\bar{c}_n$.

A2.4 Modelo mRCC : Conclusões

Neste caso pretendeu-se apenas analisar os efeitos resultantes das falhas em modo-comum. Neste sentido, foram definidos adicionalmente os seguintes parâmetros de acordo com os critérios discutidos previamente:

- ♦ $\lambda_n = \min\{\lambda_n\}$ e $\lambda_b = \min\{\lambda_b\}$;
- ♦ $\lambda_{cn}^2 = \begin{cases} 5\% \lambda_n \\ 2,5\% \lambda_n \end{cases}$, assumindo uma percentagem típica de λ_n . Neste caso admitiram-se dois valores por forma a analisar qual a importância deste parâmetro;
- ♦ $\lambda_{cn3} = \begin{cases} 0,5\% \lambda_n \\ 0,25\% \lambda_n \end{cases}$, assumindo uma redução de uma ordem de grandeza face a λ_{cn}^2 , tendo em conta uma muito menor probabilidade de ocorrência. Também aqui foram admitidos dois valores pelas mesmas razões anteriores;
- ♦ $\lambda_{cnb}^1 = \begin{cases} 5\% \lambda_b \\ 2,5\% \lambda_b \end{cases}$, assumindo uma percentagem típica de λ_b . Também aqui foram admitidos dois valores pelas mesmas razões anteriores;
- ♦ $\lambda_{cnb}^2 = 0,5\% \lambda_b$, assumindo uma redução de uma ordem de grandeza face a λ_{cn}^2 , tendo em conta uma muito menor probabilidade de ocorrência.

De forma global (Fig. A2.14, A2.15, A2.16, A2.17 e A2.18) é possível concluir que para as medidas realizadas ($\bar{A}$ e $MTTF$) a expressão obtida para λ_e (apenas os termos mais relevantes), constitui uma boa aproximação, com a excepção para alguns conjuntos de valores, não típicos, de λ_{cn2} , λ_{cn3} e λ_{cnb1} . Esta diferença resulta do facto dos termos utilizados nas aproximações deixarem de ser os mais relevantes para os valores utilizados, implicando assim a consideração de um número maior de termos¹¹³.

¹¹³Verificou-se experimentalmente que a consideração de um número maior de termos resultava numa redução do erro de aproximação.

Anexo 2

Tal como na secção anterior, as conclusões relativas a este modelo foram obtidas através da comparação dos respectivos resultados com o modelo **mR**. Os gráficos comparativos apresentados, referem-se à comparação do modelo **mRCC** (caso 1) com o modelo **mR** (caso 2), mantendo os mesmos valores dos restantes parâmetros.

A análise da evolução de $\bar{A} = f(c_n)$ (Fig. A2.14) e $MTTF = f(c_n)$ (Fig. A2.15) com λ_{cn}^2 , λ_{cn}^3 e λ_{cnb}^1 como parâmetros fixos, permite concluir em relação aos seguintes parâmetros¹¹⁴:

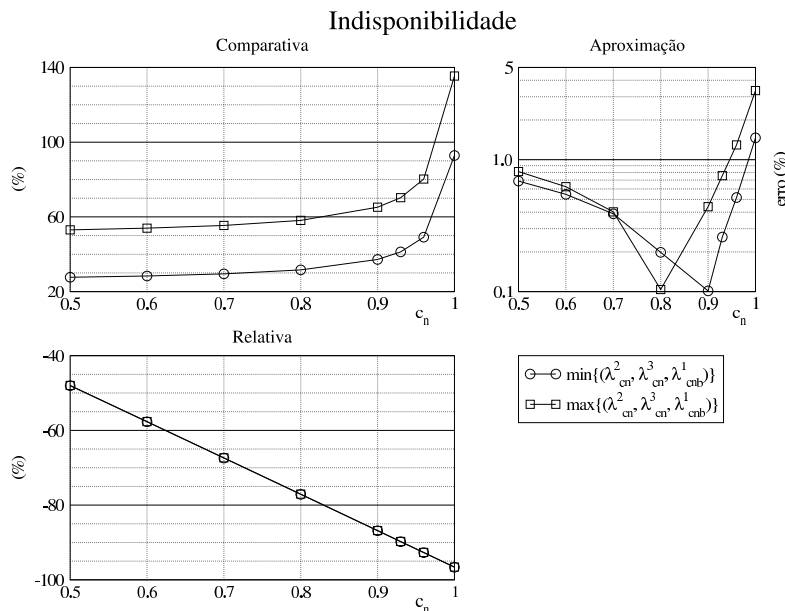


Figura A2.14 - Modelo **mRCC** vs. **mR**: indisponibilidade (c_n)

¹¹⁴As referências utilizadas no gráfico relativo foram: ($c_n=0$, $\min\{(\lambda_{cn}^2, \lambda_{cn}^3, \lambda_{cnb}^1)\}$) e ($c_n=0$, $\max\{(\lambda_{cn}^2, \lambda_{cn}^3, \lambda_{cnb}^1)\}$), respectivamente para $\min\{(\lambda_{cn}^2, \lambda_{cn}^3, \lambda_{cnb}^1)\}$ e $\max\{(\lambda_{cn}^2, \lambda_{cn}^3, \lambda_{cnb}^1)\}$.

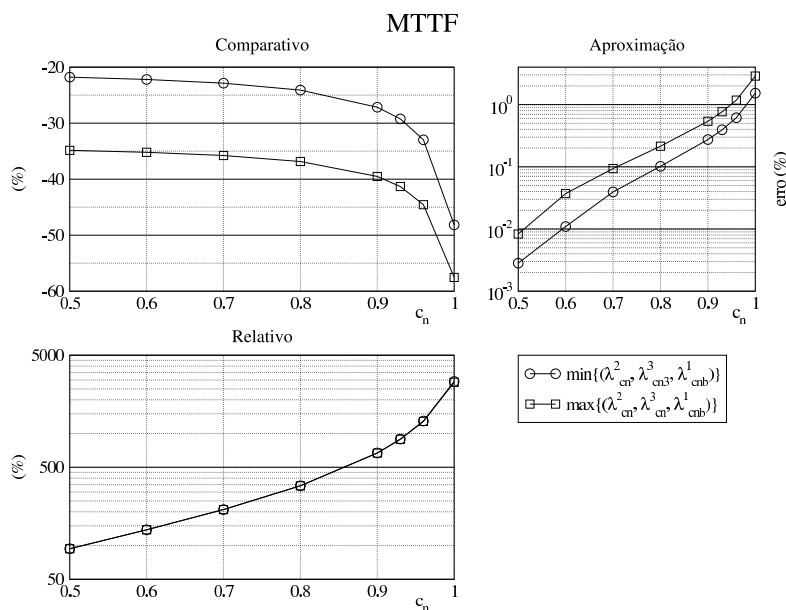


Figura A2.15 - Modelo mRCC vs. mR: MTTF (c_n)

◆ Comparativa

- c_n : a influência deste parâmetro só se torna significativa para valores superiores a 0.9, podendo analisar-se este comportamento através das expressões associadas a ambos os modelos. Assim, verifica-se que a segunda expressão (**mRCC**) pode ser decomposta na primeira (**mR**), acrescida das componentes em modo-comum, o que resulta numa imediata redução da qualidade das medidas. Para valores de $c_n \in [0.5 \dots 0.9]$ o comportamento é essencialmente constante, dado que ambas as expressões são equivalentes, a menos de um factor de escala. Para valores de $c_n > 0.9$ a primeira expressão vai aproximar-se de λ_b , e a segunda de $\lambda_b + n\lambda_{cnb}^1 + \binom{n}{2}\lambda_{cnb}^2$. Ou seja, ficam reduzidas à falhas não-cobertas que envolvem o meio de transmissão, com as do segundo caso muito superiores à primeira devido essencialmente ao efeito de $\binom{n}{2}$;
- $(\lambda_{cn}^2, \lambda_{cn}^3, \lambda_{cnb}^1)$: a influência destes parâmetros é moderada, resultando essencialmente de se ter considerado a sua existência no modelo **mRCC**, face à respectiva ausência no modelo **mR**, e da respectiva variação associada. O efeito das falhas em modo-comum, resulta, como seria de esperar, numa diminuição redução na qualidade das medidas obtidas. A necessidade de considerar os seus efeitos pode ser justificada se c_n tomar valores elevados (> 0.9), e as respectivas taxas associadas corresponderem a uma percentagem mínima das falhas normais, que pode ser deduzida a

partir da relação entre as duas expressões ($c_n > 0.9$) associadas a ambos os modelos;

◆ Relativa

- c_n : a sua influência traduz-se por um aumento quase linear das medidas. Analisando a expressão associada, verifica-se que quase todos os factores estão associados directamente a c_n ou $\bar{c}_n$ (excepto uma pequena minoria, que neste caso são constantes), resultando assim num comportamento deste tipo;
- $(\lambda_{cn}^2, \lambda_{cn}^3, \lambda_{cnb}^1)$: a sua influência é praticamente nula, dado que os factores associados a estes parâmetros são neste caso inferiores aos associados às falhas ditas normais. Além disso a sua variação, em termos absolutos, é muito reduzida.

A análise da evolução de $\bar{A} = f(\lambda_{cn}^2)$ (Fig. A2.16) com c_n , λ_{cn}^3 e λ_{cnb}^1 como parâmetros fixos, permite concluir em relação aos seguintes parâmetros¹¹⁵:

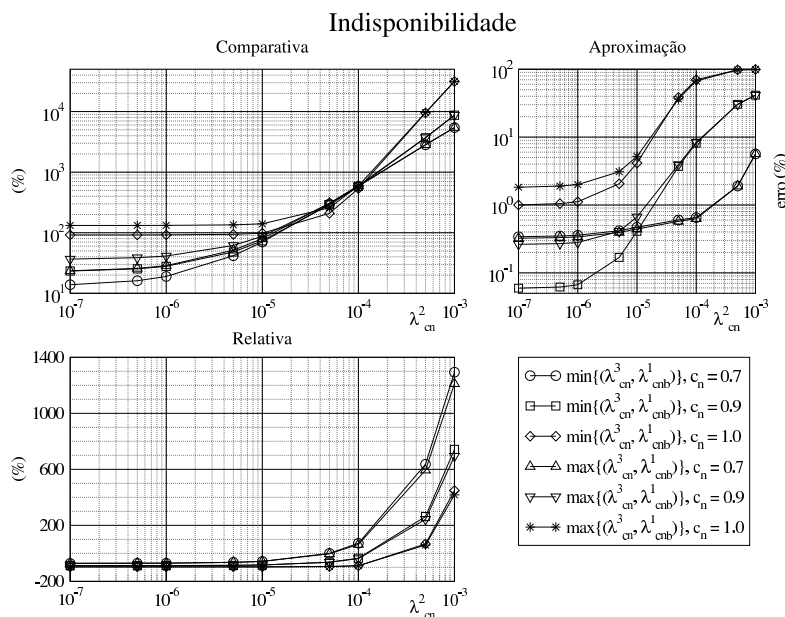


Figura A2.16 - Modelo mRCC vs. mR: indisponibilidade (λ_{cn}^2)

¹¹⁵As referências utilizadas no gráfico relativo foram: ($c_n=0$, $\min\{\lambda_{cn}^2\}$, $\min\{(\lambda_{cn}^3, \lambda_{cnb}^1)\}$) e ($c_n=0$, $\min\{\lambda_{cn}^2\}$, $\max\{(\lambda_{cn}^3, \lambda_{cnb}^1)\}$), respectivamente para $\min\{(\lambda_{cn}^3, \lambda_{cnb}^1)\}$ e $\max\{(\lambda_{cn}^3, \lambda_{cnb}^1)\}$.

♦ **Comparativa**

- λ_{cn}^2 : o seu efeito só se torna significativo para valores médios/elevados, chegando a atingir várias ordens de grandeza. Este parâmetro só surge no modelo **mRCC**, verificando-se que surge sempre associado a factores do tipo $\binom{i}{2}$, que normalmente tomam valores muito elevados. Resulta assim num efeito multiplicador de λ_{cn}^2 , que se torna significativo face aos restantes para valores médios/elevados do mesmo;
- c_n : a sua influência é moderada, resultando essencialmente do facto deste parâmetro surgir associado a um conjunto de termos adicionais no modelo **mRCC**. Estes termos estão associados a factores do tipo $\bar{c}_n \binom{i}{2} \lambda_{cn}^j$, verificado-se assim um efeito de ampliação combinado de c_n , devido à variação significativa de $\bar{c}_n$, e de $\binom{i}{j}$, que normalmente toma valores elevados sobre λ_{cn}^j ;
- $(\lambda_{cn}^3, \lambda_{cnb}^1)$: a sua influência é ligeira, sendo apenas significativa para valores reduzidos de λ_{cn}^2 . Ambos surgem apenas no modelo **mRCC**, sendo λ_{cn}^3 o mais importante, dado surgir associado a factores mais significativos face a λ_{cnb}^1 , $\binom{n}{3}$ e n respectivamente.

♦ **Relativa**

- λ_{cn}^2 : a sua influência só é significativa para valores elevados, dado que embora esteja associado a factores do tipo $\binom{i}{j}$, que resultam num forte efeito de ampliação, o valor de λ_{cn}^2 é bastante reduzido face aos restantes parâmetros;
- c_n : a sua influência é apenas significativa para valores elevados de λ_{cn}^2 , resultando num efeito moderado. Isto deve-se ao facto deste parâmetro estar associado a λ_{cn}^2 ;
- $(\lambda_{cn}^3, \lambda_{cnb}^1)$: a sua influência é muito reduzida, devido ao facto dos factores a que estão associados serem muito pouco significativos face aos restantes.

A análise da evolução de $\bar{A} = f(\lambda_{cn}^3)$ (Fig. A2.17) com c_n , λ_{cn}^2 e λ_{cnb}^1 como parâmetros fixos, permite concluir em relação aos seguintes parâmetros¹¹⁶:

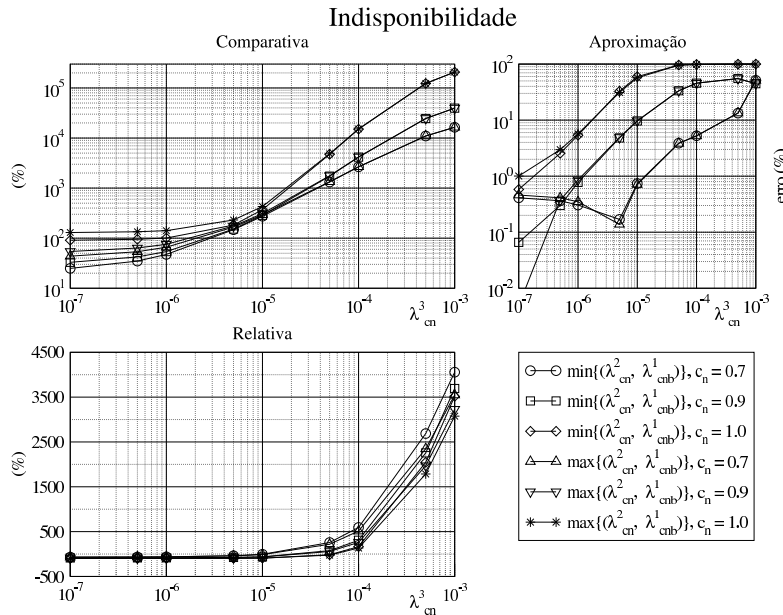


Figura A2.17 - Modelo mRCC vs. mR: indisponibilidade (λ_{cn}^3)

♦ Comparativa:

- λ_{cnb}^1 : o seu efeito só se torna significativo para valores médios/elevados, chegando a atingir várias ordens de grandeza. Este parâmetro só surge no modelo **mRCC**, verificando-se que surge sempre associado a factores do tipo $\binom{n}{3}$, que normalmente tomam valores muito elevados. Resulta assim num forte efeito multiplicador de λ_{cn}^3 que se torna significativo, face aos restantes, para valores médios/elevados do mesmo;
- c_n : a sua influência é moderada, resultando essencialmente do facto deste parâmetro surgir associado a um conjunto de termos adicionais no modelo **mRCC**. Estes termos estão associados a factores do tipo $\bar{c}_n \binom{i}{j} \lambda_{cn}^j$, verificando-se assim um efeito de ampliação combinado de c_n , devido à variação significativa de $\bar{c}_n$, e de $\binom{i}{j}$, que normalmente toma valores elevados, sobre λ_{cn}^j ;

¹¹⁶As referências utilizadas no gráfico relativo foram: ($c_n=0$, $\min\{\lambda_{cn}^3\}$, $\min\{(\lambda_{cn}^2, \lambda_{cnb}^1)\}$) e ($c_n=0$, $\min\{\lambda_{cn}^3\}$, $\max\{(\lambda_{cn}^2, \lambda_{cnb}^1)\}$), respectivamente para $\min\{(\lambda_{cn}^2, \lambda_{cnb}^1)\}$ e $\max\{(\lambda_{cn}^2, \lambda_{cnb}^1)\}$.

- $(\lambda_{cn}^2, \lambda_{cn}^3)$: a sua influência é ligeira, sendo apenas significativa para valores reduzidos de λ_{cnb}^1 . Ambos surgem apenas no modelo **mRCC**, sendo o seu efeito equilibrado.

♦ **Relativa:**

- λ_{cnb}^1 : a sua influência só é significativa para valores elevados, dado que está apenas associado a factores do tipo $(n-j)$, resultando num ligeiro efeito de ampliação. Além disso, λ_{cnb}^1 é bastante reduzido face aos restantes parâmetros, implicando que só para valores elevados o seu efeito se faça sentir;
- c_n : a sua influência é apenas significativa para valores elevados de λ_{cnb}^1 , resultando num efeito moderado. Isto deve-se ao facto deste parâmetro estar associado a λ_{cnb}^1 ;
- $(\lambda_{cn}^2, \lambda_{cn}^3)$: a sua influência é muito reduzida, devido ao facto dos factores a que estão associados serem muito pouco significativos face aos restantes.

A análise da evolução de $\bar{A} = f(\lambda_{cnb}^1)$, (Fig. A2.18) com c_n , λ_{cn}^2 e λ_{cn}^3 como parâmetros fixos, permitem concluir em relação aos seguintes parâmetros¹¹⁷:

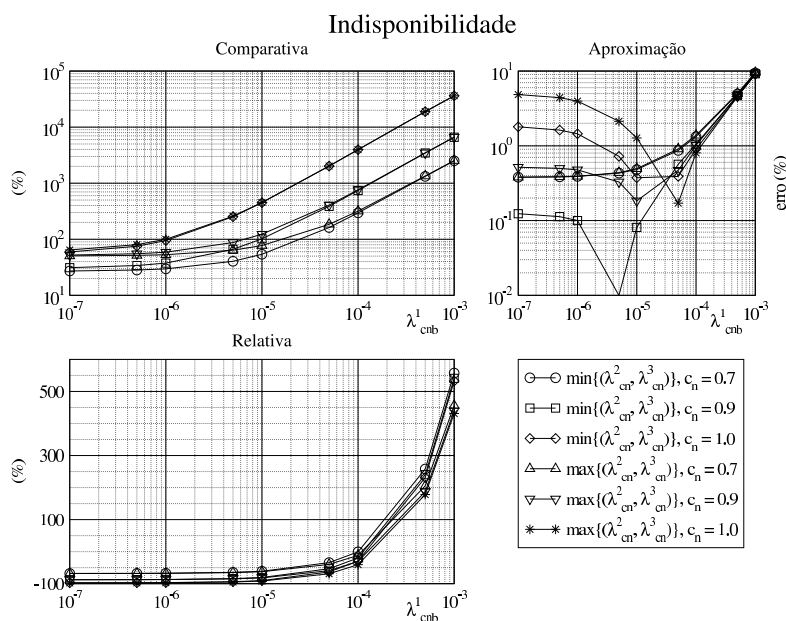


Figura A2.18 - Modelo mRCC vs. mR: indisponibilidade (λ_{cnb}^1)

¹¹⁷As referências utilizadas no gráfico relativo foram: $(c_n=0, \min\{\lambda_{cn}^3\}, \min\{(\lambda_{cn}^2, \lambda_{cnb}^1)\})$ e $(c_n=0, \min\{\lambda_{cn}^3\}, \max\{(\lambda_{cn}^2, \lambda_{cnb}^1)\})$, respectivamente para $\min\{(\lambda_{cn}^2, \lambda_{cnb}^1)\}$ e $\max\{(\lambda_{cn}^2, \lambda_{cnb}^1)\}$.

♦ Comparativa

- λ_{cnb}^1 : o seu efeito só se torna significativo para valores médios/elevados, chegando a atingir várias ordens de grandeza. Este parâmetro só surge no modelo **mRCC**, verificando-se que aparece sempre associado a factores do tipo $(n-j)$, que embora não tomem valores elevados, não se encontram, de forma geral, condicionados pelo factor de cobertura (c_n). Resulta assim um efeito multiplicador de λ_{cnb}^1 , que se torna significativo face aos restantes para valores médio/elevados do mesmo. É interessante verificar que em termos comparativos, o efeito de λ_{cnb}^1 é superior a λ_{cn}^2 e equivalente a λ_{cn}^3 . Embora λ_{cnb}^1 esteja associado a factores nitidamente inferiores, face a λ_{cn}^2 e λ_{cn}^3 , é necessário considerar que este parâmetro se encontra associado a falhas não-cobertas, ao contrário dos restantes;
- c_n : a sua influência é apenas significativa para valores médios/elevados de λ_{cnb}^1 , resultando num efeito importante. Para valores de c_n elevados, como é o caso, a redução de λ_e , é mais significativa no modelo **mR** do que no modelo **mRCC**, dado que neste último ainda existem um número considerável de falhas não-cobertas. Consequentemente a evolução comparativa é mais acentuada;
- $(\lambda_{cn}^2, \lambda_{cn}^3)$: a sua influência é muito reduzida, dado que estão associados a factores muito pouco significativos face aos restantes, além de que a sua variação em termos absolutos é reduzida;

♦ Relativa

- λ_{cnb}^1 : a sua influência só é significativa para valores elevados, dado que estão associados a factores muito pouco significativos face aos restantes;
- c_n : a sua influência é praticamente nula, devendo-se ao facto das falhas cobertas constituírem a esmagadora maioria;
- $(\lambda_{cn}^2, \lambda_{cn}^3)$: a sua influência é apenas significativa para valores elevados de λ_{cnb}^1 , resultando num efeito muito ligeiro. Este efeito está essencialmente associado à variação destes parâmetros.

A2.5 Modelo mRDCC: Conclusões

Neste caso pretendeu-se apenas analisar os efeitos resultantes das falhas em modo-comum utilizando um meio de transmissão redundante. Neste sentido, foram definidos adicionalmente os seguintes parâmetros de acordo com os critérios discutidos previamente:

- ♦ $\lambda_n = \min\{\lambda_n\}$ e $\lambda_b = \min\{\lambda_b\}$;
- ♦ $\lambda_{cn}^2 = \begin{cases} 5\% \lambda_n \\ 2,5\% \lambda_n \end{cases}$, de acordo com os pressupostos assumidos no modelo **mRCC**;
- ♦ $\lambda_{cn}^3 = \begin{cases} 0,5\% \lambda_n \\ 0,25\% \lambda_n \end{cases}$, de acordo com os pressupostos assumidos no modelo **mRCC**;
- ♦ $\lambda_{cnb}^1 = \begin{cases} 5\% \lambda_b \\ 2,5\% \lambda_b \end{cases}$, de acordo com os pressupostos assumidos no modelo **mRCC**;
- ♦ $\lambda_{cnb}^2 = 0,5\% \lambda_b$, de acordo com os pressupostos assumidos no modelo **mRCC**;
- ♦ $\lambda_{cnbb}^1 = 0,5\% \lambda_b$, assumindo uma percentagem de λ_b , neste caso fixa por se considerar que o seu efeito é muito reduzido;
- ♦ $\lambda_{cbb} = 1\% \lambda_b$, assumindo uma percentagem de λ_b , neste caso fixa por se considerar que o seu efeito é muito reduzido;
- ♦ $c_b = 0.9$, de acordo com os pressupostos assumidos no modelo **mRCC**;

De forma global (Fig. A2.20, A2.21, A2.22, A2.23 e A2.24) é possível concluir que para as medidas realizadas $\bar{A}$ e $MTTF$, a expressão obtida para λ_e (apenas os termos mais relevantes) constitui uma boa aproximação, excepto para alguns conjuntos de valores, não típicos, de λ_{cn}^2 , λ_{cn}^3 e λ_{cnb}^1 . Esta diferença resulta do facto dos termos utilizados nas aproximações deixarem de ser os mais relevantes para os valores utilizados, implicando assim a consideração de um número maior de termos¹¹⁸.

Tal como em casos anteriores, as conclusões relativas a este modelo foram obtidas através da comparação dos respectivos resultados com o modelo **mRCC**. Os gráficos comparativos apresentados, referem-se à comparação do modelo **mRDCC** (caso 1) com o modelo **mRCC** (caso 2), mantendo os mesmos valores dos restantes parâmetros.

¹¹⁸Verificou-se que a consideração de um número maior de termos resultava numa redução do erro de aproximação.

A análise da evolução de $\bar{A}=f(c_{nb})$ (Fig. A2.19) com λ_{cn}^2 , λ_{cn}^3 e λ_{cnb}^1 como parâmetros fixos, permite concluir que o efeito de c_{nb} é na maioria dos casos ligeiro. Este comportamento deve-se ao facto de c_{nb} estar apenas associado a λ_{cnb}^j , cujos valores absolutos, tendo em conta os respectivos factores multiplicativos, são, na esmagadora maioria dos casos, bastante inferiores às restantes taxas. Este parâmetro só assume especial importância no caso de c_n e c_b serem elevados ($c_n, c_b \gg 0.9$), dado que nesta situação a taxa de avarias é apenas limitada pelas taxas associadas a falhas não cobertas λ_{cnbb}^1 e λ_{cnbb} , que possuem valores equivalentes.

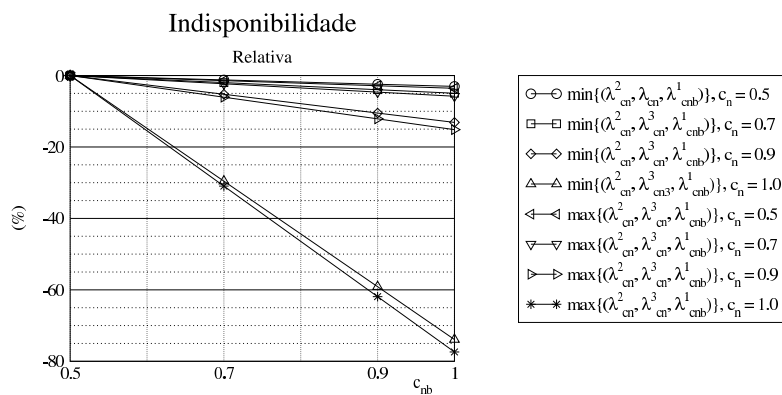


Figura A2.19 - Modelo mRDCC: indisponibilidade (c_{nb})

Neste sentido, assumiu-se na avaliação comparativa que a probabilidade de cobrir uma falha em modo-comum entre um meio físico de transmissão e um nó pode ser significativa (existe um outro meio físico alternativo), mas contudo deve ser inferior a c_b (manteve-se o valor previamente assumido $c_b=0.9$), dado que envolve um elemento de maior complexidade, considerando-se assim que $c_{nb}=0.7$. Este pressuposto tem apenas como objectivo, permitir que a comparação entre os modelo mRDCC e mRCC se realize de forma o mais simples possível. Além disso, o valor considerado é de facto pessimista (os valores práticos são tipicamente maiores), resultando assim numa avaliação comparativa com a mesma natureza.

A análise da evolução comparativa de $\bar{A} = f(c_n)$ (Fig. A2.20) e $MTTF = f(c_n)$ (Fig. A2.21), com λ_{cn}^2 , λ_{cn}^3 e λ_{cnb}^1 como parâmetros fixos, permite concluir em relação aos seguintes parâmetros:

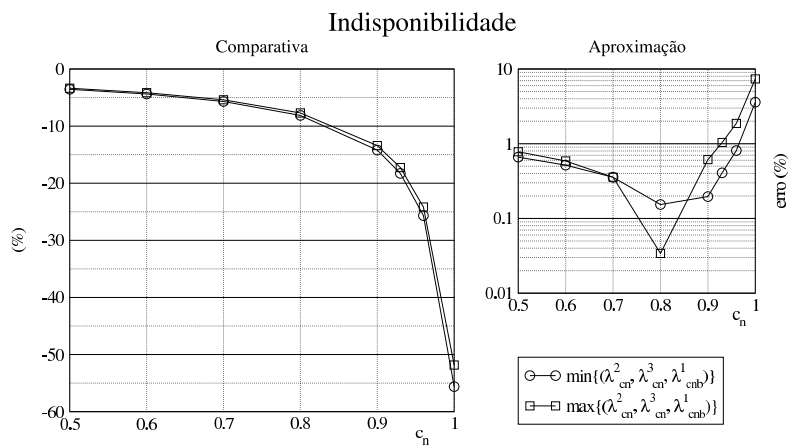


Figura A2.20 - Modelo mRDCC vs. mRCC: indisponibilidade (c_n)

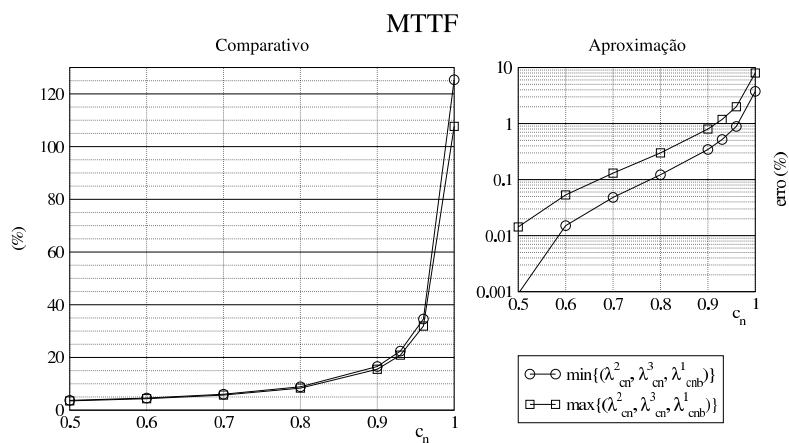


Figura A2.21 - Modelo mRDCC vs. mRCC: MTTF (c_n)

- ♦ c_n : o seu aumento conduz, naturalmente a uma melhoria da qualidade das medidas obtidas. Contudo, a introdução de um segundo meio físico de transmissão não se traduz por um aumento significativo das mesmas, resultando tipicamente em aumentos entre 5...20%. No entanto, para $c_n \geq 0.96$, a evolução é muito significativa. Comparando as expressões de (mRCC) com (mRDCC), verifica-se que para $c_n \in [0.5 \dots 0.96]$ são razoavelmente aproximadas, com uma ligeira superioridade para a primeira, sendo a diferença existente resultante essencialmente de existirem falhas ($\lambda_b, \lambda_{cnb}^1, \lambda_{cnb}^2$) que são cobertas no segundo caso e não no primeiro. Contudo, para $c_n \geq 0.96$ ambas as expressões se reduzem, praticamente, às falhas não-cobertas, que no primeiro caso (mRCC) são bastante superiores ao segundo (mRDCC), originando assim diferenças significativas;
- ♦ $(\lambda_{cn}^2, \lambda_{cn}^3, \lambda_{cnb}^1)$: a sua influência é reduzida, dado que estes parâmetros em ambas as expressões estão associados a termos iguais, com uma excepção para λ_{cnb}^1 , mas cuja diferença é mínima. É possível também concluir que só devem ser considerados os efeitos das falhas em modo-comum se todos os factores de cobertura forem elevados, em especial c_n que deverá ter valores muito próximos da unidade, e se as respectivas taxas corresponderem a uma percentagem mínima das falhas normais, que pode ser deduzida a partir da relação entre as expressões (considerando $c_n \geq 0.96$) associadas a ambos os modelos.

A análise da evolução comparativa de $\bar{A} = f(\lambda_{cn}^2)$ (Fig. A2.22) com c_n , λ_{cn}^3 e λ_{cnb}^1 como parâmetros fixos, permitem concluir em relação aos seguintes parâmetros:

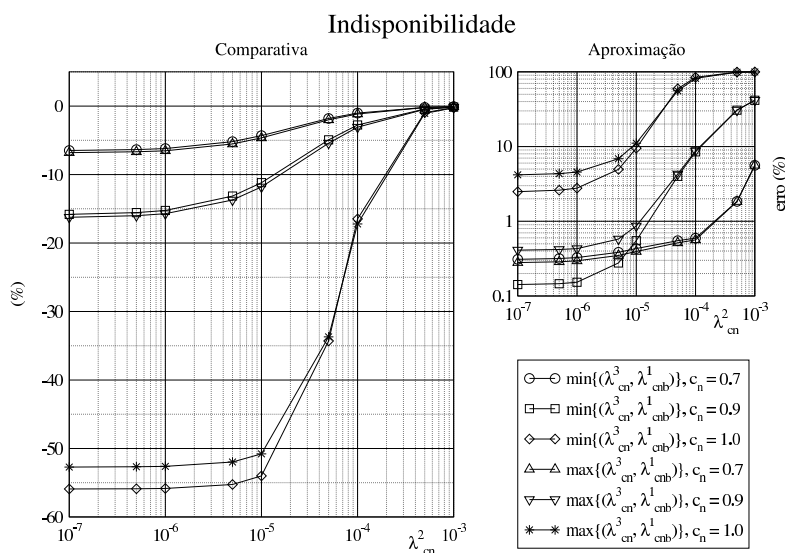


Figura A2.22 - Modelo mRDCC vs.mRCC: indisponibilidade (λ_{cn}^2)

- ♦ λ_{cn}^2 : para valores reduzidos/médios a influência é muito ligeira, dado que neste caso o parâmetro não assume valores importantes face aos restantes, e que em ambas as expressões (**mRDCC** e **mRCC**) dependem de forma equivalente deste parâmetro. Para valores elevados, ambas as expressões tendem para o mesmo factor: $\sim \bar{c}_n \left(\frac{n}{2}\right) \lambda_{cn}^2$, daí a convergência de valores. Em qualquer dos casos verifica-se sempre um aumento da qualidade das medidas obtidas devido à existência de um número maior de falhas cobertas no caso dual;
- ♦ c_n : este parâmetro assume um papel importante quando toma valores elevados ($c_n > 0.9$), com λ_{cn}^2 reduzido/médio. Este comportamento deve-se ao facto de em ambos os casos os termos associados c_n serem os mesmos, e os termos não abrangidos por c_n serem em (**mRCC**) superiores a (**mRDCC**);
- ♦ $(\lambda_{cn}^3, \lambda_{cnb}^1)$: a influência destes parâmetros é muito reduzida, porque λ_{cn}^3 está associado a um termo equivalente em ambos os casos. Além disso, λ_{cnb}^1 só assume importância nos situações em que $c_n \rightarrow 1$ e λ_{cn}^2 é reduzido, dado que só nestes casos é que seu efeito é significativo.

A análise da evolução comparativa de $\bar{A} = f(\lambda_{cn}^3)$ (Fig. A2.23) com c_n , λ_{cn}^2 e λ_{cnb}^1 como parâmetros fixos, permite concluir em relação aos seguintes parâmetros:

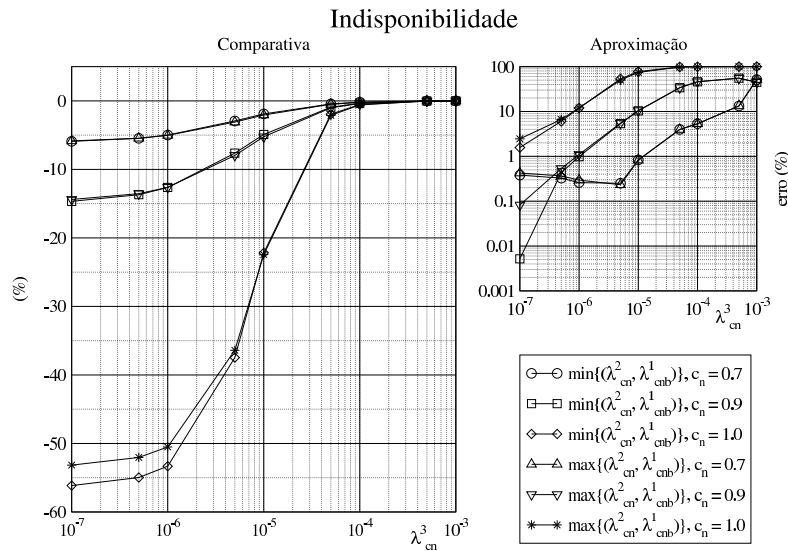


Figura A2.23 - Modelo **mRDCC** vs. **mRCC**: indisponibilidade (λ_{cn}^3)

- λ_{cn}^3 : para valores reduzidos a influência é muito ligeira, dado que, neste caso não assume valores importantes face aos restantes. Além disso, ambas as expressões (**mRCC** e **mRDCC**) dependem de forma equivalente deste parâmetro. Para valores médios/elevados, ambas as expressões tendem para o mesmo factor: $\sim \bar{c}_n \binom{n}{3} \lambda_{cn}^3$, daí a convergência de valores. Em qualquer dos casos verifica-se sempre um aumento qualidade das medidas, devido à existência de um factor de cobertura (c_{nb}) no caso dual;
- c_n : este parâmetro assume um papel importante quando toma valores elevados ($c_n > 0.9$), com λ_{cn}^3 reduzidos/médios. Este comportamento deve-se ao facto de em ambos os casos os termos associados c_n serem os mesmos, e os termos não abrangidos por c_n serem em (**mRCC**) superiores a (**mRDCC**);
- $(\lambda_{cn}^2, \lambda_{cnb}^1)$: a influência destes parâmetros é muito reduzida, porque λ_{cn}^2 está associado a um termo equivalente em ambos os casos, e λ_{cnb}^1 só assume importância nas situações em que $c_n \rightarrow 1$ e λ_{cn}^3 é reduzido, dado que só nestas casos é que o seu efeito é significativo.

A análise da evolução comparativa de $\bar{A} = f(\lambda_{cnb}^1)$ (Fig. A2.24) com c_n , λ_{cn}^2 e λ_{cn}^3 como parâmetros fixos, permite concluir em relação aos seguintes parâmetros:

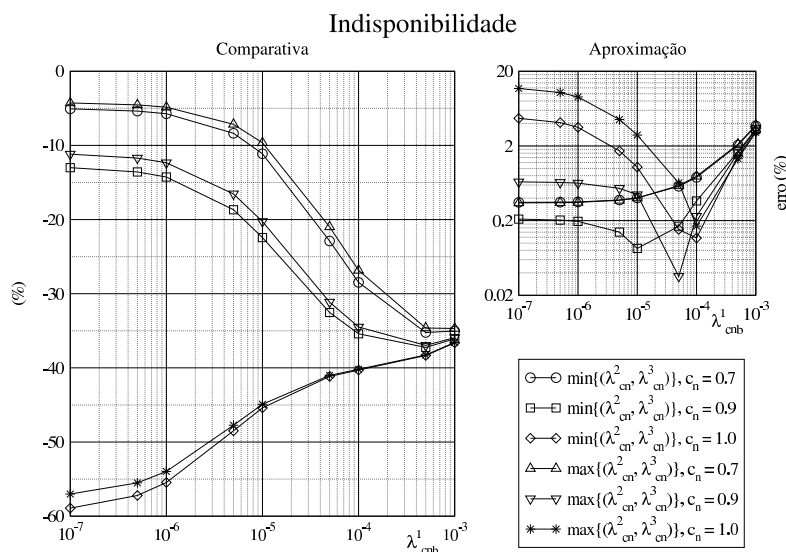


Figura A2.24 - Modelo mRDCC vs.mRCC: indisponibilidade (λ_{cnb}^1)

- ♦ λ_{cnb}^1 : para valores reduzidos a influência é muito ligeira, dado que neste caso não assume valores importantes face aos restantes. Além disso, ambas as expressões (**mRCC** e **mRDCC**) dependem de forma aproximada deste parâmetro. Para valores médios/elevados existe uma convergência para uma

relação constante, dado que convergem respectivamente para $n\lambda_{cnb}^1$ e para $2\bar{c}_{nb}\lambda_{cnb}^1$. Em qualquer dos casos, verifica-se sempre um aumento da qualidade das medidas, devido à existência de um factor de cobertura (c_{nb}) no caso dual;

- ♦ c_n : este parâmetro assume um papel importante para valores elevados ($c_n > 0.9$), com λ_{cn}^3 reduzido/médio. Este comportamento deve-se ao facto de em ambos os casos os termos associados c_n serem os mesmos, e de os termos não abrangidos por c_n serem em (**mRCC**) superiores a (**mRDCC**);
- ♦ $(\lambda_{cn}^2\lambda_{cn}^3)$: a influência destes parâmetros é muito reduzida, porque λ_{cn}^2 está associado a um termo equivalente, em ambos os casos, e λ_{cn}^3 só assume importância nos casos em que $c_n \rightarrow 1$ e λ_{cnb}^1 é reduzido, dado que só nestas situações é que o seu efeito é significativo.

Esta página foi intencionalmente deixada em branco

Anexo 3

Timed Activity:	t_msg_slv_r
Deterministic Distribution Parameters	Value /* Deve incluir o TSDR*/ TRMSG+TSDRE
Activation Predicate	1
Reactivation Predicate	0
Case Distributions	case 1 p1 case 2 p2 case 3 p3

Input Gate:	i_msg_s
Predicate	/* Resposta à mensagem */ ((BUS->Mark() == B_PFD) (BUS->Mark() == B_PMS) (BUS->Mark() == B_PDG) (BUS->Mark() == B_PPR) (BUS->Mark() == B_PCF)) && (NSA->Mark() == ts->Mark())

Input Gate:	i_msg_slv_r
Predicate	/* Inicio da resposta */ (s_slv_r->Mark() > 0)

Output Gate:	o_msg_r_e
Function	/* Resposta com erros */ BUS->Mark() = B_RER; s_slv_r->Mark() = 0;

Anexo 3

Output Gate:	o_msg_r_ok
Function	<pre> /* Qual o tipo de pedido enviado pelo mestre */ switch(s_slv_r->Mark()) { /* Mensagem */ case (B_PMS): { BUS->Mark() = B_RMS; break; } /* Diagnóstico */ case (B_PDG): { BUS->Mark() = B_RDG; break; } /* Parametros */ case (B_PPR): { BUS->Mark() = B_RPR; break; } /* Configuração */ case (B_PCF): { BUS->Mark() = B_RCF; break; } /* FDL */ case (B_PFD): { BUS->Mark() = B_RFD; } } s_slv_r->Mark() = 0; </pre>

Output Gate:	o_msg_r_ok_d
Function	<pre> /* Resposta correcta com pedido de diagnóstico */ BUS->Mark() = B_RMSD; s_slv_r->Mark() = 0; </pre>

Output Gate:	o_rst
Function	<pre> /* Reset dos escravos */ passive->Mark() = 1; ts->Mark() = RST_A->Mark(); if (RST_IN->Mark() < NEE) { /* Continua */ RST_OUT->Mark() = RST_IN->Mark() +1; } else { /* Pára */ /* Reset dos mestres */ RST_OUT->Mark() = 1; } RST_IN->Mark() = 0; </pre>

Output Gate:	o_slv_r
Function	<pre> /* Resposta enviada correctamente */ /* Qual o tipo de pedido */ switch(BUS->Mark()) { /* Mensagem */ case (B_PMS): { TRMSG = TRMS; p1 = (1.0 - p_e_msg_r)*(1.0-p_diag); p2 = (1.0 - p_e_msg_r)*(p_diag); p3 = p_e_msg_r; break; } /* Diagnóstico */ case (B_PDG): { TRMSG = TRDG; p1 = (1.0 - p_e_diag_r); p2 = 0.0; p3 = (p_e_diag_r); break; } /* Parâmetros */ case (B_PPR): { TRMSG = TRPR; p1 = (1.0 - p_e_prm_r); p2 = 0.0; p3 = (p_e_prm_r); break; } /* Configuração */ case (B_PCF): { TRMSG = TRCF; p1 = (1.0 - p_e_cfg_r); p2 = 0.0; p3 = (p_e_cfg_r); break; } /* FDL */ case (B_PFD): { TRMSG = TRFD; p1 = (1.0 - p_e_fdl_r); p2 = 0.0; p3 = p_e_fdl_r; break; } } /* Inicio da resposta */ s_slv_r->Mark() = BUS->Mark(); BUS->Mark() = B_INR; </pre>

Instantaneous Activity:	t_da_rest
Case Distributions	<pre> case 1 if (LMP->Mark() == 1) return 1.0; else return (1.0 - p_e_dup); case 2 if (LMP->Mark() == 1) return 0.0; else return (p_e_dup); </pre>

Instantaneous Activity:	t_e_da
Case Distributions	<pre> case 1 (1- p_st_1 - p_st_2 - p_st_3) case 2 p_st_1 case 3 p_st_2 case 4 p_st_3 </pre>

Anexo 3

Instantaneous Activity:	t_e_sa
Case Distributions	case 1 (1- p_st_1 - p_st_2 - p_st_3) case 2 p_st_1 case 3 p_st_2 case 4 p_st_3

Instantaneous Activity:	t_sa_da
Case Distributions	case 1 (1.0 - p_e_sa_da) case 2 p_e_sa_da

Instantaneous Activity:	t_sa_rest
Case Distributions	case 1 (1.0 - p_e_dup) case 2 p_e_dup

Output Gate:	o_DA1
Function	<pre>/* Erro em DA */ if (NSA->Mark() == AD1) { /* Não é erro */ BUS->Mark() = B_TOKL; } else { BUS->Mark() = B_TED; NSA->Mark() = AD1; }</pre>

Output Gate:	o_DA2
Function	<pre>/* Erro em DA */ if (NSA->Mark() == AD2) { /* Não é erro */ BUS->Mark() = B_TOKL; } else { BUS->Mark() = B_TED; NSA->Mark() = AD2; }</pre>

Output Gate:	o_DA3
Function	<pre>/* Erro em DA */ if (NSA->Mark() == AD3) { /* Não é erro */ BUS->Mark() = B_TOKL; } else { BUS->Mark() = B_TED; NSA->Mark() = AD3; }</pre>

Output Gate:	o_DA_g
Function	<pre> /* Erro em DA */ BUS->Mark() = B_TED; if (LMP->Mark() == 0) { NSA->Mark() = 128; } else { NSA->Mark() = 129; } </pre>

Output Gate:	o_SA1
Function	<pre> /* Erro em SA */ if (PSA->Mark() == AD1) { /* Não é erro */ BUS->Mark() = B_TOKL; } else { BUS->Mark() = B_TES; PSA->Mark() = AD1; } </pre>

Output Gate:	o_SA2
Function	<pre> /* Erro em SA */ if (PSA->Mark() == AD2) { /* Não é erro */ BUS->Mark() = B_TOKL; } else { BUS->Mark() = B_TES; PSA->Mark() = AD2; } </pre>

Output Gate:	o_SA3
Function	<pre> /* Erro em SA */ if (PSA->Mark() == AD3) { /* Não é erro */ BUS->Mark() = B_TOKL; } else { BUS->Mark() = B_TES; PSA->Mark() = AD3; } </pre>

Output Gate:	o_SA_g
Function	<pre> /* Erro em SA */ BUS->Mark() = B_TES; if (LMP->Mark() == 0) { PSA->Mark() = 128; } else { PSA->Mark() = 129; } </pre>

Output Gate:	o_da_dp
Function	<pre> /* Erro duplo no envio do token para uma outra estação */ BUS->Mark() = B_TED; NSA->Mark() = 128; </pre>

Output Gate:	o_sa_dp
Function	<pre> /* Erro duplo no envio do token para uma outra estação */ BUS->Mark() = B_TES; PSA->Mark() = 128; </pre>

Anexo 3

Timed Activity:	t_GUD
Deterministic Distribution Parameters	Value G*TTR
Activation Predicate	1
Reactivation Predicate	0

Timed Activity:	t_IDLE
Deterministic Distribution Parameters	Value TIDLE
Activation Predicate	1
Reactivation Predicate	0

Timed Activity:	t_ST
Deterministic Distribution Parameters	Value ST
Activation Predicate	1
Reactivation Predicate	0
Case Distributions	case 1 (1.0-p_e_st) case 2 p_e_st

Timed Activity:	t_TOF
Exponential Distribution Parameters	Rate TOF
Activation Predicate	(none)
Reactivation Predicate	(none)

Timed Activity:	t_TT
Deterministic Distribution Parameters	Value TT
Activation Predicate	1
Reactivation Predicate	0
Case Distributions	case 1 p_eg case 2 p_el_nd case 3 p_el_d case 4 (1.0 - p_eg - p_el_nd - p_el_d)

Timed Activity:	t_TTH
Deterministic Distribution Parameters	Value TTH
Activation Predicate	1
Reactivation Predicate	0

Timed Activity:	t_fdl_mr
Deterministic Distribution Parameters	Value /* Deve incluir o T_SDR*/ TRFD+TSDRM
Activation Predicate	1
Reactivation Predicate	0
Case Distributions	case 1 (1-p_e_fdl_r) case 2 p_e_fdl_r

Timed Activity:	t_msg
Deterministic Distribution Parameters	Value TMSG
Activation Predicate	1
Reactivation Predicate	0
Case Distributions	case 1 if (msg_tp->Mark() == T_DAT) return (1.0-p_e_msg_p); if (msg_tp->Mark() == T_DIG) return (1.0-p_e_diag_p); if (msg_tp->Mark() == T_PRM) return (1.0-p_e_prm_p); if (msg_tp->Mark() == T_CFG) return (1.0-p_e_cfg_p); if (msg_tp->Mark() == T_NAK) return (1.0); if (msg_tp->Mark() == T_FDL) return (1.0-p_e_fdl_p); case 2 if (msg_tp->Mark() == T_DAT) return (p_e_msg_p); if (msg_tp->Mark() == T_DIG) return (p_e_diag_p); if (msg_tp->Mark() == T_PRM) return (p_e_prm_p); if (msg_tp->Mark() == T_CFG) return (p_e_cfg_p); if (msg_tp->Mark() == T_NAK) return (0.0); if (msg_tp->Mark() == T_FDL) return (p_e_fdl_p);

Timed Activity:	t_msg_st
Deterministic Distribution Parameters	Value ST
Activation Predicate	1
Reactivation Predicate	0

Timed Activity:	t_slv1_p
Deterministic Distribution Parameters	Value TSLV1P
Activation Predicate	(none)
Reactivation Predicate	(none)

Anexo 3

Timed Activity:	t_slv_CI
Deterministic Distribution Parameters	Value TCIN
Activation Predicate	1
Reactivation Predicate	0

Timed Activity:	t_slv_MS
Deterministic Distribution Parameters	Value TMSI
Activation Predicate	1
Reactivation Predicate	0

Timed Activity:	t_tto
Deterministic Distribution Parameters	Value /* Slot Time */ 6*ST+2*RST_A->Mark()*ST
Activation Predicate	1
Reactivation Predicate	0

Input Gate:	b_bea
Predicate	/* Atualização das próximas estações */ (BUS_UP->Mark() == 4) && (BUS->Mark() == B_LAS)
Function	BUS_UP->Mark() = 0;

Input Gate:	i_bpe
Predicate	/* Atualização das próximas estações */ (BUS_DW->Mark() == 3) && ((BUS->Mark() == B_LMP) (BUS->Mark() == B_LAS))
Function	BUS_DW->Mark() = 0;

Input Gate:	i_fdl_ir
Predicate	/* Resposta a FDL do Mestre */ (BUS->Mark() == B_PFD) && (NSA->Mark() == ts->Mark()) && (offline->Mark() == 0)

Input Gate:	i_gup
Predicate	/* Inicia o GUD */ (offline->Mark() == 0) && (listen->Mark() == 0) && (gap_s->Mark() == 1)
Function	gap_s->Mark() = 0;

Input Gate:	i_lbe
Predicate	/* Leiura do bus de uma estação que não possui o token */ (BUS_DW->Mark() == 1) && (ck_pass->Mark() == 0)
Function	BUS_DW->Mark() = 0;

Input Gate:	i_lbm
Predicate	/* Leiura do bus de uma estação que possui o token */ (BUS_DW->Mark() == 1) && (ck_pass->Mark() == 1)
Function	BUS_DW->Mark() = 0;

Input Gate:	i_msg_ir
Predicate	/* Inicio da resposta - necessário para parar ST */ (msg_w_r->Mark() == 1) && (BUS->Mark() == B_INR)
Function	msg_w_r->Mark() = 0;

Input Gate:	i_msg_p
Predicate	/* Transmissão de pedido de mensagem */ (msg_p_s->Mark() == 1)
Function	msg_p_s->Mark() = 0; msg_r->Mark()++; if (msg_tp->Mark() == T_FDL) { msg_ct->Mark() = 2; }

Input Gate:	i_msg_r_e
Predicate	/* Resposta recebida com erros */ (BUS->Mark() == B_RER) && (msg_f_r->Mark() == 1)
Function	msg_f_r->Mark() = 0;

Input Gate:	i_msg_r_ok
Predicate	/* Resposta OK */ ((BUS->Mark() == B_MFD) (BUS->Mark() == B_RFD) (BUS->Mark() == B_RMS) (BUS->Mark() == B_RDG) (BUS->Mark() == B_RMSD) (BUS->Mark() == B_RPR) (BUS->Mark() == B_RCF)) && (msg_f_r->Mark() == 1)
Function	msg_f_r->Mark() = 0;

Input Gate:	i_ore
Predicate	/* Retira estações do anel */ (BUS_DW->Mark() == 5) && (BUS->Mark() == B_OR) && (use->Mark() == 0)
Function	BUS_DW->Mark() = 0;

Input Gate:	i_orm
Predicate	/* Retira estações do anel até ao que possui o token */ (BUS_DW->Mark() == 5) && (BUS->Mark() == B_OR) && (use->Mark() == 1)
Function	BUS_DW->Mark() = 0; /* Fim de ciclo */ /* Teste eventual de erros */ BUS->Mark() = B_FIM;

Input Gate:	i_pass
Predicate	(pass->Mark() == 1)
Function	pass->Mark() = 0; tk_r_ct->Mark()++;

Anexo 3

Input Gate:	i_slv1_p
Predicate	(1==1)

Input Gate:	i_slv_CI
Predicate	/* Inicio Control Interval */ (offline->Mark() == 0) && (s_slv_CI->Mark() == 1)
Function	s_slv_CI->Mark() = 0;

Input Gate:	i_slv_MS
Predicate	/* Inicio do MIn Slave Interval */ (offline->Mark() == 0) && (s_slv_MS->Mark() == 1)
Function	s_slv_MS->Mark() = 0;

Input Gate:	i_tk_e
Predicate	/* Erro no token */ ((BUS->Mark() == B_TOKL) (BUS->Mark() == B_TES) (BUS->Mark() == B_TED)) && (ck_pass->Mark() == 1)

Input Gate:	i_tk_me
Predicate	/* Teste eventual de erros */ ((tk_el_ct->Mark() == 2) (tk_eg_ct->Mark() == 2)) && (ck_pass->Mark() == 1) && ((BUS->Mark() == B_TD) (BUS->Mark() == B_FIM))
Function	BUS->Mark() = 0; ck_pass->Mark() = 0;

Input Gate:	i_to
Predicate	/* Timeout após perda do token */ ((BUS->Mark() == B_TLL) (BUS->Mark() == B_TLG)) && (offline->Mark() == 0)

Output Gate:	o_GUD
Function	/* GUD expirou */ /* Existem estações intermédias ? */ if (gap_l->Mark() > 0) { /* Sim */ gap_e->Mark() = 1; /* Inicializa contador de mensagens FDL */ fdl_ct->Mark() = 0; } else { /* Não */ /* Re-inicializa GUD */ gap_s->Mark() = 1; }

Output Gate:	o_bea
Function	<pre> /* É o mestre que possui o token ? */ if (pass->Mark() == 1) { /* Sim */ ns->Mark() = NSA->Mark(); NSA->Mark() = 0; BUS->Mark() = 0; } else { /* Não */ /* Actualiza a próxima estação */ if (active->Mark() == 1) { ns->Mark() = NSA->Mark(); NSA->Mark() = ts->Mark(); } /* Continua */ BUS_DW->Mark() = 4; } </pre>

Output Gate:	o_bpe
Function	<pre> /* É o mestre que possui o token ? */ if (pass->Mark() == 1) { /* Sim */ switch (BUS->Mark()) { /* Simples actualização do LAS global */ case (B_LAS): { /* Actualiza próximo mestre */ ps->Mark() = PSA->Mark(); PSA->Mark() = 0; /* Continua no sentido oposto */ NSA->Mark() = ts->Mark(); BUS_DW->Mark() = 4; break; } /* Limpeza */ case (B_LMP): { /* Limpa situações transitórias */ tk_sa->Mark() = 0; tk_da->Mark() = 0; LMP->Mark() = 0; BUS->Mark() = 0; break; } } } else { /* Não */ /* Qual o modo */ switch (BUS->Mark()) { /* Simples actualização do LAS global */ case (B_LAS): { /* É um mestre intermédio ? */ if (active->Mark() == 1) { /* Actualiza próxima estação */ ps->Mark() = PSA->Mark(); PSA->Mark() = ts->Mark(); } break; } /* Limpeza */ case (B_LMP): { /* Limpa situações transitórias */ tk_sa->Mark() = 0; tk_da->Mark() = 0; break; } } /* Continua para a próxima */ BUS_UP->Mark() = 3; } </pre>

Anexo 3

Output Gate:	o_ck_time
Function	<pre> /* Testa condições para a transmissão de mensagens */ AUX->Mark() = 0; BUS->Mark() = 0; NSA->Mark() = 0; msg_tp->Mark() = 0; /* O TTH não expirou OU expirou e ainda não enviou 1 mensagem E não enviou FDL */ if ((tth_e->Mark() == 0) (tth_e->Mark() == 1) && (msg_ct->Mark() == 0)) && (msg_ct->Mark() != 2)) { /* Sim */ /* O Min Slave Interval expirou ? */ if (slv_MS_e->Mark() == 1) { /* Sim */ /* O Control Interval expirou ? */ if (slv_CI_e->Mark() == 1) { /* Sim */ /* Inicia Control Interval */ slv_CI_e->Mark() = 0; s_slv_CI->Mark() = 1; /* Mensagem de estado do mestre */ msg_tp->Mark() = T_NAK; msg_p_s->Mark() = 1; TMSG=TNAK; TIDLE = TID2; } else { /* Não */ /* Qual o escravo */ if (slv_ct->Mark() == 1) { /* Primeiro */ NSA->Mark() = SLV_1->Mark(); /* Qual o estado */ switch (slv1_op->Mark()) { /* Dados */ case (DAT): { /* Sim */ msg_tp->Mark() = T_DAT; msg_p_s->Mark() = 1; TMSG = TPMS; TIDLE = TID1; break; } /* Diagnostico normal */ case (DNR): { /* Sim */ msg_tp->Mark() = T_DIG; msg_p_s->Mark() = 1; TMSG = TPDG; TIDLE = TID1; break; } /* Diagnóstico inicial */ case (DIN): { msg_tp->Mark() = T_DIG; msg_p_s->Mark() = 1; TMSG = TPDG; TIDLE = TID1; break; } /* Diagnóstico final */ case (DFI): { msg_tp->Mark() = T_DIG; msg_p_s->Mark() = 1; TMSG = TPDG; TIDLE = TID1; break; } } } } } } </pre>

Output Gate:	o_ck_time (Continuação)
Function	<pre> /* Parametros */ case (PRM): { msg_tp->Mark() = T_PRM; msg_p_s->Mark() = 1; TMSG = TPRR; TIDLE = TID1; break; } /* Configuração */ case (CFG): { msg_tp->Mark() = T_CFG; msg_p_s->Mark() = 1; TMSG = TPCF; TIDLE = TID1; break; } } /* Terminou ? */ if (slv_ct->Mark() == (SLV+1)) { /* Inicia Min Slave Interval */ slv_MS_e->Mark() = 0; s_slv_MS->Mark() = 1; slv_ct->Mark() = 1; } } /* Não existem mensagens E GUP terminou E holding time não expirou */ if ((msg_tp->Mark() == 0) && (gap_e->Mark() == 1) && (tth_e->Mark() == 0)) { /* FDL */ /* Chegou ao fim ? */ if (fdl_ct->Mark() == gap_l->Mark()) { /* Sim */ /* Inicializa GUP */ gap_e->Mark() = 0; gap_s->Mark() = 1; fdl_ct->Mark() = 0; } else { /* Não */ /* Determina o endereço da estação a enviar o FDL */ if (ts->Mark() + fdl_ct->Mark() <= HSA) { NSA->Mark() = ts->Mark() + fdl_ct->Mark() + 1; } else { NSA->Mark() = HSA - ts->Mark() - fdl_ct->Mark(); } /* Utiliza AUX para "enviar" o próximo mestre */ AUX->Mark() = ns->Mark(); PSA->Mark() = ts->Mark(); msg_tp->Mark() = T_FDL; msg_p_s->Mark() = 1; TMSG = TPDF; TIDLE = TID2; } } } /* Transmite o token ? */ if (msg_tp->Mark() == 0) { /* Sim */ msg_ct->Mark() = 0; msg_tp->Mark() = 0; msg_r->Mark() = 0; trr_r->Mark() = 0; tth_e->Mark() = 0; tth_s->Mark() = 0; a_pass->Mark() = 1; } </pre>

Output Gate:	o_eg
Function	<pre> /* Erro grave */ tk_eg_ct->Mark()++; tk_el_ct->Mark()++; ck_pass->Mark() = 1; BUS->Mark() = B_TD; PSA->Mark() = 0; NSA->Mark() = 0; </pre>

Anexo 3

Output Gate:	o_el_d
Function	<pre>/* Erro ligeiro detectado */ tk_eg_ct->Mark() = 0; tk_el_ct->Mark()++; ck_pass->Mark() = 1; BUS->Mark() = B_TD; PSA->Mark() = 0; NSA->Mark() = 0;</pre>

Output Gate:	o_el_nd
Function	<pre>/* Erro ligeiro não detectado */ tk_eg_ct->Mark() = 0; tk_el_ct->Mark()++; /* Actualizações locais */ ck_pass->Mark() = 1; /* Envia para uma outra estação */ NTK_OUT->Mark() = 1; BUS->Mark() = B_TND; NSA->Mark() = ns->Mark(); PSA->Mark() = ts->Mark();</pre>

Output Gate:	o_fdl_ir
Function	<pre>/* Inicia resposta */ BUS->Mark() = B_INR; s_fdl_r->Mark() = 1;</pre>

Output Gate:	o_fdl_r_e
Function	<pre>/* Erro na resposta */ BUS->Mark() = B_RER;</pre>

Output Gate:	o_fdl_r_ok
Function	<pre>/* Inserir no anel ? */ if (listen->Mark() == 1) { /* Sim */ /* Actualizações locais */ listen->Mark() = 0; active->Mark() = 1; trr_a->Mark() = 0; trr_r->Mark() = 0; tk_sa->Mark() = 0; tk_da->Mark() = 0; gap_s->Mark() = 1; /* Actualiza LAS */ ps->Mark() = PSA->Mark(); ns->Mark() = AUX->Mark(); /* Informa a estação que inseriu */ NSA->Mark() = ts->Mark(); PSA->Mark() = 0; AUX->Mark() = 0; BUS->Mark() = B_MFD; } else { /* Não */ /* Informa a estação */ BUS->Mark() = B_RFD; }</pre>

Output Gate:	o_lbe
Function	<pre> /* Leitura do token recebido */ /* Está operaconal ? */ if (offline ->Mark() == 0) { /* Sim */ /* Token correcto recebido do mestre anterior e dirigido ao próprio OU já aceite sem problemas*/ if (((BUS->Mark() == B_TOK) && (PSA->Mark() == ps->Mark()) && (NSA->Mark() == ts->Mark()) && (active->Mark() == 1)) ((BUS->Mark() == B_TPC) && (AUX->Mark() ==0))) { /* Sim */ /* Foi aceite ? */ if (BUS->Mark() == B_TOK) { /* Não */ /* Token passado correctamente */ active->Mark() = 0; use->Mark() = 1; BUS->Mark() = B_TPC; } tk_sa->Mark() = 0; tk_da->Mark() = 0; } else { /* Não */ /* É o primeiro token e está relacionado com a estação ? */ if ((tk_sa->Mark() == 0) && (tk_da->Mark() == 0) && ((PSA->Mark() == ts->Mark()) && (listen->Mark() == 1)) ((NSA->Mark() == ts->Mark()) && (active->Mark() == 1)) ((PSA->Mark() == ts->Mark()) && (active->Mark() == 1)) (NSA->Mark() == PSA->Mark()) && (active->Mark() == 1) && (BUS->Mark() != B_TOK))) { /* Sim */ tk_sa->Mark() = PSA->Mark(); tk_da->Mark() = NSA->Mark(); /* É necessário para fazer "limpeza" */ LMP->Mark() = 1; } else { /* Não */ /* Está em Listen e o token tem o SA da estação ? */ if ((listen->Mark() == 1) && (PSA->Mark() == ts->Mark())) { /* Sim */ /* Segundo com o mesmo SA e não dirigido ao próprio ? */ if (tk_sa->Mark() == PSA->Mark()) { /* Sim */ /* Julga que existe uma outra estação com o mesmo SA /* Vai para offline */ listen->Mark() = 0; offline->Mark() = 1; } else { /* Guarda token */ tk_sa->Mark() = PSA->Mark(); tk_da->Mark() = NSA->Mark(); } } } } } */ </pre>

Anexo 3

Output Gate:	o_lbe
Function	<pre> /* Activo ? */ if (active->Mark() == 1) { /* Sim */ /* Token com o SA da estação e não dirigido ao próprio ? */ if ((PSA->Mark() == ts->Mark()) && (NSA->Mark() != ts->Mark()) (NSA->Mark() != PSA->Mark())) { /* Sim */ /* É o segundo igual */ if (tk_sa->Mark() == PSA->Mark()) { /* Sim */ /* Julga que existe uma outra estação /* Vai para listen */ active->Mark() = 0; listen->Mark() = 1; /* Perde o LAS */ ns->Mark() = 0; ps->Mark() = 0; /* GAP */ gap_s->Mark() = 0; gap_e->Mark() = 0; gap_l->Mark() = 0; fdl_ct->Mark() = 0; /* TTR */ trr_r->Mark() = 0; trr_a->Mark() = 0; } else { /* Guarda token */ tk_sa->Mark() = PSA->Mark(); tk_da->Mark() = NSA->Mark(); } } /* Token dirigido ao próprio ? */ if ((NSA->Mark() == ts->Mark()) && (NSA->Mark() != PSA->Mark())) { /* Sim */ /* Com a mesma origem ? */ if ((tk_sa->Mark() == PSA->Mark()) && (tk_da->Mark() == NSA->Mark())) { /* Sim */ /* Aceita o token */ active->Mark() = 0; use->Mark() = 1; /* Actualiza LAS */ ps->Mark() = PSA->Mark(); /* Token enviado correctamente ? */ if (BUS->Mark() == B_TOK) { /* Sim */ AUX->Mark() = 0; } else { /* Não */ /* Saida de mestres do anel AUX->Mark() = 1; } /* Token passado correctamente */ BUS->Mark() = B_TPC; } else { /* Guarda token */ tk_sa->Mark() = PSA->Mark(); tk_da->Mark() = NSA->Mark(); } } } </pre>

Output Gate:	o_lbe (Continuação)
Function	<pre> /* Token (errado) com SA igual DA */ if ((NSA->Mark() == PSA->Mark()) && (BUS->Mark() != B_TOK)) { /* Sim */ /* Igual ao anterior */ if ((tk_sa->Mark() == PSA->Mark()) && (tk_da->Mark() == NSA->Mark())) { /* Sim */ /* Coloca-o fora do anel */ /* Atualizações locais */ tk_r_ct->Mark() = 0; tk_el_ct->Mark() = 0; tk_eg_ct->Mark() = 0; active->Mark() = 0; listen->Mark() = 1; tk_sa->Mark() = 0; tk_da->Mark() = 0; /* GAP */ gap_s->Mark() = 0; gap_e->Mark() = 0; gap_l->Mark() = 0; fdl_ct->Mark() = 0; /* Perde o LAS */ ns->Mark() = 0; ps->Mark() = 0; /* TTR */ trr_r->Mark() = 0; trr_a->Mark() = 0; } else { /* Guarda token */ tk_sa->Mark() = PSA->Mark(); tk_da->Mark() = NSA->Mark(); } } } } } } /* Se o token foi aceite */ if (use->Mark() == 1) { /* Inicialização do TTH */ if (trr_a->Mark() == 0) { /* Primeira vez */ TTH = TTR; tth_s->Mark() = 1; } else { /* Restantes */ TTH= trr_a->Mark() + TTR - LastActionTime; /* TTH expirou ? */ if (TTH <= 0) { /* Sim */ tth_e->Mark() = 1; } else { /* Inicializa TTH */ tth_s->Mark() = 1; } } /* Guarda tempo de recepção do token */ trr_r->Mark() = TTH; trr_a->Mark() = LastActionTime; /* Inicialização de mensagens */ msg_ct->Mark() = 0; TIDLE = TID1; /* Calcula numero estações no GAPL */ if ((ns->Mark() - ts->Mark()) > 0) { gap_l->Mark() = ns->Mark() - ts->Mark() - 1; } else { gap_l->Mark() = HSA - ts->Mark() + ns->Mark(); } } } /* Continua para o próximo mestre */ BUS_UP->Mark() = 1; </pre>

Anexo 3

Output Gate:	o_lbm
Function	<pre> /* Leitura do token recebido */ /* Token transmitido correctamente ou aceite por outra estação */ if ((BUS->Mark() == B_TOK) (BUS->Mark() == B_TPC)) { /* Sim */ /* Passado ao próprio OU aceite correctamente por outra estação ? */ if ((BUS->Mark() == B_TOK) && (NSA->Mark() == ts->Mark()) ((BUS->Mark() == B_TPC) && (AUX->Mark() == 0))) { /* Sim */ /* Actualizações locais */ tk_eg_ct->Mark() = 0; tk_el_ct->Mark() = 0; tk_r_ct->Mark() = 0; tk_sa->Mark() = 0; tk_da->Mark() = 0; /* Mudança de estado */ ck_pass->Mark() = 0; /* É a própria estação */ if (NSA->Mark() == ts->Mark()) { /* Sim */ use->Mark() = 1; /* Inicialização do TTH */ if (trr_a->Mark() == 0) { /* Primeira vez */ TTH = TTR; tth_s->Mark() = 1; } else { /* Restantes */ TTH= trr_a->Mark() + TTR - LastActionTime; /* TTH expirou ? */ if (TTH <= 0) { /* Sim */ tth_e->Mark() = 1; } else { /* Inicializa TTH */ tth_s->Mark() = 1; } } /* Guarda tempo de recepção do token */ trr_r->Mark() = TTH; trr_a->Mark() = LastActionTime; /* Inicialização de mensagens */ msg_ct->Mark() = 0; TIDLE = TID1; /* Calcula numero estações no GAPL */ if ((ns->Mark() - ts->Mark()) > 0) { gap_l->Mark() = ns->Mark() - ts->Mark() - 1; } else { gap_l->Mark() = HSA - ts->Mark() + ns->Mark(); } } else { /* Não */ active->Mark() = 1; } BUS->Mark() = B_FIM; } } </pre>

Output Gate:	o_lbm (Continuação)
Function	<pre> else { /* Necessário retirar mestres do anel ? */ if (AUX->Mark() == 1) { /* Sim */ /* Outra estação aceitou o token */ AUX->Mark() = 0; BUS->Mark() = B_OR; BUS_UP->Mark() = 5; } else { /* Não */ /* Nenhuma estação aceitou o token */ /* Fim de ciclo */ /* Teste eventual de erros */ BUS->Mark() = B_FIM; } } } else { /* Não */ /* SA igual a DA não dirigido ao próprio */ if ((NSA->Mark() == PSA->Mark()) && (NSA->Mark() != ts->Mark())) { /* Sim */ /* É o primeiro token ? */ if ((tk_sa->Mark() == 0) && (tk_da->Mark() == 0)) { /* Sim */ tk_sa->Mark() = PSA->Mark(); tk_da->Mark() = NSA->Mark(); /* É necessário para fazer "limpeza" */ LMP->Mark() = 1; } else { /* Não */ /* SA igual DA e igual ao anterior ? */ if ((tk_sa->Mark() == PSA->Mark()) && (tk_da->Mark() == NSA->Mark())) { /* Sim */ /* Coloca-o fora do anel */ /* Atualizações locais */ tk_r_ct->Mark() = 0; tk_el_ct->Mark() = 0; tk_eg_ct->Mark() = 0; ck_pass->Mark() = 0; active->Mark() = 0; listen->Mark() = 1; tk_sa->Mark() = 0; tk_da->Mark() = 0; /* GAP */ gap_s->Mark() = 0; gap_e->Mark() = 0; gap_l->Mark() = 0; fdl_ct->Mark() = 0; /* Perde o LAS */ ns->Mark() = 0; ps->Mark() = 0; /* TTR */ trr_r->Mark() = 0; trr_a->Mark() = 0; /* Perda global do token */ BUS->Mark() = B_TLG; } else { /* Não */ /* Guarda token */ tk_sa->Mark() = PSA->Mark(); tk_da->Mark() = NSA->Mark(); /* Fim de ciclo */ /* Teste eventual de erros */ BUS->Mark() = B_FIM; } } } } </pre>

Anexo 3

Output Gate:	o_lbm (Continuação)	
Function	<pre>else { /* Necessário retirar mestres do anel ? */ if (AUX->Mark() == 1) { /* Sim */ /* Outra estação aceitou o token */ AUX->Mark() = 0; BUS->Mark() = B_OR; BUS_UP->Mark() = 5; } else { /* Não */ /* Nenhuma estação aceitou o token */ /* Fim de ciclo */ /* Teste eventual de erros */ BUS->Mark() = B_FIM; } } else { /* Fim de ciclo */ /* Teste eventual de erros */ BUS->Mark() = B_FIM; } } NSA->Mark() = 0; PSA->Mark() = 0;</pre>	

Output Gate:	o_msg_p_ok
Function	<pre> /* Pedido enviado sem erros */ /* Qual o tipo de mensagem */ switch (msg_tp->Mark()) { /* Dados */ case (T_DAT): { /* Pedido enviado à estação */ BUS->Mark() = B_PMS; msg_w_r->Mark() = 1; break; } /* Diagnóstico */ case (T_DIG): { /* Pedido enviado à estação */ BUS->Mark() = B_PDG; msg_w_r->Mark() = 1; break; } /* Parametros */ case (T_PRM): { /* Pedido enviado à estação */ BUS->Mark() = B_PPR; msg_w_r->Mark() = 1; break; } /* Configuração */ case (T_CFG): { /* Pedido enviado à estação */ BUS->Mark() = B_PCF; msg_w_r->Mark() = 1; break; } /* Sem confirmação */ case (T_NAK): { /* Passa à próxima mensagem */ msg_r->Mark() = 0; use->Mark() = 1; break; } /* FDL */ case (T_FDL): { /* Pedido enviado à estação */ BUS->Mark() = B_PFD; msg_w_r->Mark() = 1; break; } } </pre>

Output Gate:	o_msg_r_e
Function	<pre> /* Mensagem com pedido ou resposta com erros*/ BUS->Mark() = 0; /* Qual o tipo de mensagem */ switch (msg_tp->Mark()) { /* Dados */ case (T_DAT): { /* Número máximo de transmissões ? */ if (msg_r->Mark() == 2) { /* Sim */ msg_r->Mark() = 0; use->Mark() = 1; BUS->Mark() = 0; NSA->Mark() = 0; /* Próximo escravo */ slv_ct->Mark()++; } else { /* Retransmissão */ msg_p_s->Mark() = 1; } break; } } </pre>

Anexo 3

Output Gate:	o_msg_r_e (Continuação)
Function	<pre> /* Diagnóstico */ case (T_DIG): { /* Número máximo de transmissões ? */ if (msg_r->Mark() == 2) { /* Sim */ msg_r->Mark() = 0; use->Mark() = 1; BUS->Mark() = 0; NSA->Mark() = 0; /* Próximo escravo */ slv_ct->Mark()++; } else { /* Retransmissão */ msg_p_s->Mark() = 1; } break; } /* Parametros */ case (T_PRM): { /* Número máximo de transmissões ? */ if (msg_r->Mark() == 2) { /* Sim */ msg_r->Mark() = 0; use->Mark() = 1; BUS->Mark() = 0; NSA->Mark() = 0; /* Próximo escravo */ slv_ct->Mark()++; } else { /* Retransmissão */ msg_p_s->Mark() = 1; } break; } /* Configuração */ case (T_CFG): { /* Número máximo de transmissões ? */ if (msg_r->Mark() == 2) { /* Sim */ msg_r->Mark() = 0; use->Mark() = 1; BUS->Mark() = 0; NSA->Mark() = 0; /* Próximo escravo */ slv_ct->Mark()++; } else { /* Retransmissão */ msg_p_s->Mark() = 1; } break; } /* FDL */ case (T_FDL): { /* Número máximo de transmissões ? */ if (msg_r->Mark() == 2) { /* Sim */ msg_r->Mark() = 0; use->Mark() = 1; BUS->Mark() = 0; NSA->Mark() = 0; /* Próximo escravo */ fdL_ct->Mark()++; } else { /* Retransmissão */ msg_p_s->Mark() = 1; } } } </pre>

Output Gate:	o_msg_r_ok
Function	<pre> /* Resposta correcta */ msg_r->Mark() = 0; /* Qual o tipo de mensagem */ switch (msg_tp->Mark()) { /* Dados */ case (T_DAT): { /* Qual o escravo */ if (NSA->Mark() == SLV_1->Mark()) { /* Deadline */ slv1_m->Mark() = 0; slv1_dl->Mark() = 0; slv1_op->Mark() = DAT; /* Existe pedido de diagnóstico */ if (BUS->Mark() == B_RMSD) { /* Sim */ slv1_op->Mark() = DNR; } } /* Próximo escravo */ slv_ct->Mark()++; break; } /* Diagnóstico */ case (T_DIG): { /* Qual o escravo */ if (NSA->Mark() == SLV_1->Mark()) { /* Qual o estado */ switch (slv1_op->Mark()) { /* Diagnóstico inicial */ case (DIN): { /* Parametros */ slv1_op->Mark() = PRM; break; } /* Diagnóstico final */ case (DFI): { /* Dados */ slv1_op->Mark() = DAT; break; } /* Diagnóstico normal */ case (DNR): { /* Dados */ slv1_op->Mark() = DAT; break; } } } /* Próximo escravo */ slv_ct->Mark()++; break; } /* Parametros */ case (T_PRM): { /* Qual o escravo */ if (NSA->Mark() == SLV_1->Mark()) { /* Configuração */ slv1_op->Mark() = CFG; } /* Próximo escravo */ slv_ct->Mark()++; break; } } </pre>

Anexo 3

Output Gate:	o_msg_r_ok (Continuação)
Function	<pre> /* Configuração */ case (T_CF6): { /* Qual o escravo */ if (NSA->Mark() == SLV_1->Mark()) { /* Diagnóstico final */ slv1_op->Mark() = DFI; } /* Próximo escravo */ slv_ct->Mark()++; break; } /* FDL */ case (T_FDL): { /* Qual o tipo de resposta */ switch (BUS->Mark()) { /* Resposta normal*/ case (B_RFD): { /* Próximo escravo */ fdl_ct->Mark()++; break; } /* Mestre inserido */ case (B_MFD): { /* Novo mestre no anel */ /* Era o único ? */ if (ns->Mark() == ts->Mark()) { /* Sim */ ns->Mark() = NSA->Mark(); ps->Mark() = NSA->Mark(); } else { /* Não */ ns->Mark() = NSA->Mark(); } /* Actualiza GAPL */ gap_l->Mark() = fdl_ct->Mark(); fdl_ct->Mark() = 0; gap_e->Mark() = 0; gap_s->Mark() = 1; break; } } break; } /* Verificar novas condições */ BUS->Mark() = 0; NSA->Mark() = 0; use->Mark() = 1; </pre>

Output Gate:	o_off
Function	<pre> /* Saída do estado de offline */ offline->Mark() = 0; listen->Mark() = 1; /* Escravos em diagnóstico inicial*/ slv_MS_e->Mark() = 1; s_slv_CI->Mark() = 1; slv_ct->Mark() = 1; slv1_op->Mark() = DIN; </pre>

Output Gate:	o_ok
Function	<pre> /* Token OK */ tk_el_ct->Mark() = 0; tk_eg_ct->Mark() = 0; ck_pass->Mark() = 1; BUS->Mark() = B_TOK; NSA->Mark() = ns->Mark(); PSA->Mark() = ts->Mark(); BUS_UP->Mark() = 1; </pre>

Output Gate:	o_ore
Function	<pre> /* Retira estação do anel */ /* É um mestre activo ? */ if (active->Mark() == 1) { /* Coloca-o fora do anel */ /* Actualizações locais */ active->Mark() = 0; listen->Mark() = 1; /* Perde o LAS */ ns->Mark() = 0; ps->Mark() = 0; /* GAP */ gap_s->Mark() = 0; gap_e->Mark() = 0; gap_l->Mark() = 0; fdl_ct->Mark() = 0; /* TTR */ trr_r->Mark() = 0; trr_a->Mark() = 0; } /* Continua para o próximo mestre */ BUS_UP->Mark() = 5; </pre>

Output Gate:	o_p
Function	<pre> /* É necessário "limpeza" ? */ if (LMP->Mark() == 1) { /* Sim */ BUS->Mark() = B_LMP; BUS_UP->Mark() = 3; } /* Passagem do token */ pass->Mark() = 1; /* DEBUG use->Mark() = 1; trr_a->Mark() = 0; TTH = TTR; tth_s->Mark() = 1; msg_ct->Mark() = 0; TIDLE = TID1; */ </pre>

Anexo 3

Output Gate:	o_sl
Function	<pre> /* Fim do slot time */ /* Número máximo de transmissões ? */ if (tk_r_ct->Mark() == 3) { tk_eg_ct->Mark() = 0; tk_el_ct->Mark() = 0; tk_r_ct->Mark() = 0; /* Procura o próximo mestre */ /* É o primeiro ? */ if (ns->Mark() == MAD1->Mark()) { /* Sim */ ns->Mark() = MAD2->Mark(); } else { /* É o segundo */ if (ns->Mark() == MAD2->Mark()) { /* Sim */ /* Unica estação no anel */ ns->Mark() = ts->Mark(); ps->Mark() = ts->Mark(); } } NSA->Mark() = 0; PSA->Mark() = 0; BUS->Mark() = 0; /* Passagem do token ao próximo mestre */ a_pass->Mark() = 1; } else { /* Não */ /* Continua a tentar */ NSA->Mark() = 0; PSA->Mark() = 0; BUS->Mark() = 0; pass->Mark() = 1; } </pre>

Output Gate:	o_sl_e
Function	<pre> /* Erros durante o Slot Time */ tk_el_ct->Mark() = 0; tk_eg_ct->Mark() = 0; tk_r_ct->Mark() = 0; active->Mark() = 1; /* Perda local do Token */ BUS->Mark() = B_TLL; NSA->Mark() = 0; PSA->Mark() = 0; </pre>

Output Gate:	o_slv1_p
Function	<pre> /* Produção de mensagens */ /* A deadline expirou */ if (slv1_m->Mark() == 1) { slv1_dl->Mark() = 1; } else { slv1_m->Mark() = 1; } </pre>

Output Gate:	o_slv_MR
Function	<pre> /* Timer expirou */ slv_MS_e->Mark() = 1; slv_ct->Mark() = 1; </pre>

Output Gate:	oTkLe
Function	<pre> /* Erro no token */ if (BUS->Mark() == B_TOKL) { /* Erro que não devia ser contabilizado */ tk_el_ct->Mark() = 0; tk_eg_ct->Mark() = 0; BUS->Mark() = B_TOK; } else { BUS->Mark() = B_TND; } BUS_UP->Mark() = 1; </pre>

Output Gate:	oTkMe
Function	<pre> /* Existiram erros 2 consecutivos */ BUS->Mark() = 0; NSA->Mark() = 0; PSA->Mark() = 0; /* Coloca-o fora do anel */ /* Atualizações locais */ tk_r_ct->Mark() = 0; tk_sa->Mark() = 0; tk_da->Mark() = 0; /* GAP */ gap_s->Mark() = 1; gap_e->Mark() = 0; gap_l->Mark() = 0; fdl_ct->Mark() = 0; /* TTR */ trr_r->Mark() = 0; trr_a->Mark() = 0; /* Que tipo de erro */ if (tk_eg_ct->Mark() == 2) { /* Graves */ offline->Mark() = 1; /* Atualiza LAS local */ ns->Mark() = 0; ps->Mark() = 0; /* Perda do token local */ BUS->Mark() = B_TLL; } else { /* Ligeiros */ listen->Mark() = 1; /* Perda do token local */ BUS->Mark() = B_TLL; } tk_eg_ct->Mark() = 0; tk_el_ct->Mark() = 0; </pre>

Output Gate:	oTo
Function	<pre> /* Re-inicializa Mestre */ if (BUS->Mark() == B_TLG) { /* Re-inicialização do anel */ /* Unico mestre no anel */ ns->Mark() = ts->Mark(); ps->Mark() = ts->Mark(); } tth_e->Mark() = 1; BUS->Mark() = 0; listen->Mark() = 0; active->Mark() = 0; use->Mark() = 1; </pre>

Esta página foi intencionalmente deixada em branco

| Bibliografia

- [AAC+90] J. Arlat, M. Aguera, Y. Crouzet, J-C. Fabre, E. Martins, D. Powell, **Experimental Evaluation of the Fault-Tolerance of an Atomic Multicast Protocol**, *IEEE Transactions on Reliability*, Vol. 39, No. 4, pp. 455-467, 1990.
- [AB90] N. Audsley, A. Burns, **Real-Time System Scheduling**, *Technical Report YCS-134*, University of York, 1990.
- [ABD+95] N. Audsley, A. Burns, R. Davis, K. Tindell, A. Wellings, **Fixed Priority Pre-emptive Scheduling: A Historical Perspective**, *Real-Time Systems*, Vol. 8, pp. 173-198, 1995.
- [ACC+93] J. Arlat, A. Costes, Y. Crouzet, J-C. Laprie, D. Powell, **Fault Injection and Dependability Evaluation of Fault-Tolerant Systems**, *IEEE Transactions on Computers*, Vol. 42, No.8, pp. 913-923, 1993.
- [ACK+03] J. Arlat, Y. Crouzet, J. Karlsson, P. Folkesson, E. Fuchs, G. Leber, **Comparison of Physical and Software-Implemented Fault Injection**, *IEEE Transactions on Computers*, Vol. 52, No.9, pp. 1115-1133, 2003.
- [AD00] J. Andrews, S. Dunnett, **Event-Tree Analysis Using Binary Decision Diagrams**, *IEEE Transactions on Reliability*, Vol. 49, No. 2, pp. 230-238, 2000.
- [AGH02] Ö. Askerdal, M. Gäfvert, M. Hiller, N. Suri, **A Control Teory Approach for Analysing the Effects of Data Errors in Safety-Critical Control Systems**, *Proceedings of the Pacific Rim International Sysmposium on Dependable Computing, PRDC'02*, 2002.
- [AH02] A. Adamyan, D. He, **Failure and Safety Assessment of Systems using Petri Nets**, *Proceedings of the IEEE International Conference on Robotics & Automation*, 2002.
- [AHL87] H. Ammar, Y. Huang, R. Liu, **Hierarchical Models for Systems Reliability, Mantainability, and Availability**, *IEEE Transactions on Circuits and Systems*, Vol. CAS-34, No. 6, pp. 629-638, 1987.
- [AI89] H. Ammar, S. Islam, **Time Scale Decomposition of Generalized Stochastic Petri Net Models**, *IEEE Transactions on Software Engineering*, Vol. 15, No. 6, pp. 809-820, 1989.
- [AJK+00] J. Arlat, T. Jarbouï, K. Kanoun, D. Powell, **Dependability Assessment of GUARDS Instances**, *Proceedings of the 4th International Computer Performance and Dependability Symposium*, 2000.

Bibliografia

- [AK02] C. Almeida, K. Kanoun, **Stepwise Construction and Refinement of Dependability Models**, *Proceedings of the International Conference on Dependable Systems and Networks, DSN'02*, 2002.
- [AKL90] J. Arlat, K. Kanoun, J.-C. Laprie, **Dependability Modeling and Evaluation of Software Fault-Tolerant Systems**, *IEEE Transactions on Computers*, Vol. 39, No.4, pp. 504-512, 1990.
- [ALR00] A. Avizienis, J.-C. Laprie, B. Randell, **Dependability of computer systems: fundamental concepts terminology and examples**, LAAS Technical Report, 2000.
- [Alv03] M. Alves, **Real-Time Communications over Hybrid Wired/Wireless PROFIBUS-Based Networks**, Tese de Doutorado, Faculdade de Engenharia da Universidade do Porto, 2003.
- [APF02] L. Almeida, P. Pedreiras, J. Fonseca, **The FTT-CAN Protocol: Why and How**, *IEEE Transactions on Industrial Electronics*, Vol. 49, No. 6, pp. 1189-1201, 2002.
- [ASB+03] A. Ademaj, H. Sivencrona, G. Bauer, J. Torin, **Evaluation of Fault Handling of the Time-Triggered Architecture with Bus and Star Topology**, *Proceedings of the International Conference on Dependable Systems and Networks*, 2003.
- [Ata93] Y. Atamna, **RTPS a Tool for Stochastic Timed Petri Net**, *LAAS Report 93280*, 1993.
- [ATF+02] L. Almeida, E. Tovar, J. Fonseca, F. Vasques, **Schedulability Analysis of Real-Time Traffic in WorldFIP Networks: An Integrated Approach**, *IEEE Transactions on Industrial Electronics*, Vol. 49, No. 5, pp. 1165-1174, 2002.
- [Atw86] C. Atwood, **The Binomial Failure Rate Common Cause Model**, *Technometrics*, Vol. 28, No. 2, pp. 139-148, 1986.
- [AV98] M. Alves, F. Vasques, **Confiança no Funcionamento: Conceitos Básicos e Terminologia**, Relatório Técnico, 1998.
- [BA91] R. Billinton, R. Allan, **Reliability Evaluation of Engineering Systems: Concepts and Techniques**, Plenum Press, 1991.
- [BAA+93] C. Béounes, M. Aguéra, J. Arlat, S. Bachmann, J.-E. Doucet, K. Kanoun, J.-C. Laprie, S. Metge, J. Souza, D. Powell, P. Spiesser, **Surf-2: A Program for Dependability Evaluation of Complex Hardware and Software Systems**, *Proceedings of the 23rd International Fault-Tolerant Computing Symposium*, 1993.
- [BAD+04] R. Bono, R. Alexander, A. Dorman, Y. Kim, J. Reisdorf, **Analysing Reliability - A Simple Yet Regorous Approach**, *IEEE Transactions on Industry Applications*, Vol. 40, No. 4, pp. 950-957, 2004.
- [Bal95] G. Balbo, **On the Success of Stochastic Petri Nets**, *Proceedings of the 6th International Workshop on Petri Nets and Performance Models, PNPm'95*, 1995.
- [Ban01] T. Bangemann, **Management of Distributed Automation Systems Based on Web Technologies**, *IFAC International Conference on Fieldbus Systems and their Applications - FeT01*, 2001.

-
- [BAN04] I. Boster, A. Burns, G. Rodríguez-Navas, **Comparing Real-time Communication under Electromagnetic Interference**, *Proceedings of the 16th Euromicro Conference on Real-Time Systems*, 2004.
- [BB03] I. Boster, A. Burns, **An Analysable Bus-Guardian for Event-Triggered Communication**, *Proceedings of the 24th IEEE International Real-Time Systems Symposium*, 2003.
- [BBB02] I. Broster, G. Bernat, A. Burns, **Weakly Hard Real-Time Constraints on Controller Area Network**, *Proceedings of the 14th Euromicro Conference on Real-Time Systems, ECRTS'02*, 2002.
- [BBL01] G. Bernat, A. Burns, A. Llamosí, **Weakly Hard Real-Time Systems**, *IEEE Transactions on Computers*, Vol. 50, No. 4, pp. 308-321, 2001.
- [BBN02] I. Boster, A. Burns, G. Rodríguez-Navas, **Probabilistic Analysis of CAN with Faults**, *Proceedings of the 23th Real-Time System Symposium*, 2002.
- [BCF+87] G. Balbo, G. Chiola, G. Franceschinis, G. Roet, **On the efficient construction of the tangible reachability graph of generalized stochastic Petri nets**, *Proceedings 2nd International Workshop on Petri Nets and Performance Models*, pp. 85-92, 1987.
- [BCN+00] J. Banks, J. Carson, B. Nelson, D. Nicol, **Discrete-Event System Simulation**, 3rd Edition, Prentice Hall, 2000.
- [BCP02] G. Bernat, A. Colin, S. Petters, **WCET Analysis of Probabilistic Hard-Real Time Systems**, *Proceedings of the 23rd IEEE Real-Time Systems Symposium, RTSS'02*, 2002.
- [BCS69] W. Bouricius, W. Carter, P. Schneider, **Reliability Modeling Techniques for Self-Repairing Computer Systems**, *Proceedings of the 24th Annual ACM National Conference*, 1969.
- [Ben94] N. Lopez-Benitez, **Dependability Modeling and Analysis of Distributed Systems**, *IEEE Transactions on Software Engineering*, Vol. 20, No. 5, pp. 345-352, 1994.
- [BES+96] F. Brasileiro, P. Ezhilchelvan, S. Shrivastava, N. Speirs, S. Tao, **Implementing Fail-Silent Nodes for Distributed Systems**, *IEEE Transactions on Computers*, Vol. 45, No. 11, pp. 1226-1238, 1996.
- [BFL+01] A. Bondavalli, A. Fantechi, D. Latella, L. Simoncini, **Design Validation of Embedded Dependable Systems**, *IEEE Micro*, September, pp. 52-62, 2001.
- [BFS+03] . Bucci, A. Fedeli, L. Sassoli, E. Vicario, **Modeling flexible real-time systems with preemptive time petri nets**, *Proceedings of the Euromicro Conference on Real-Time Systems, ECRTS'03*, 2003.
- [BH02] A. Bobbio, A. Horváth, **Petri Nets with Discrete Phase Type Timming: A Bridge Between Stochastic and Functional Analysis**, *Electronic Notes in Theoretical Computer Science*, Vol. 52, No. 3, 2002.
- [BHT02] A. Bobbio, A. Horváth, M. Telek, **PhFit: A General Phase-type Fitting Tool**, *Proceedings of the International Conference on Dependable Systems and Networks, DSN'02*, 2002.
-

Bibliografia

- [BHT04] A. Bobbio, A. Horváth, M. Telek, **The Scale Factor: A New Degree of Freedom in Phase Type Approximation**, *Performance Evaluation*, Vol. 56 (1-4), pp. 121-144, 2004.
- [BKP+95] A. Bobbio, V. Kulkarni, A. Puliafito, M. Telek, K. Trivedi, **Preemptive Repeat Identical Transitions in Markov Regenerative Stochastic Petri Nets**, *6th International Conference on Petri Nets and Performance Models*, PNPM95, pp. 113-122, 1995.
- [BM99] L. Bello, O. Mirabella, **A Fault Tolerance Analysis of Profibus Systems by Means of Generalized Stochastic Petri Nets**, *Proceedings of the 25th IEEE International Conference on Industrial Electronics, Control and Instrumentation*, IECON'99, 1999.
- [BNS+01] A. Bondavalli, M. Nelli, L. Simoncini, G. Mongardi, **Hierarchical Modelling of Complex Control Systems: Dependability Analysis of a Railway Interlocking**, *International Journal of Computer Systems Science & Engineering*, 2001.
- [BO97] J. Barrenscheen, G. Otte, **Analysis of the Physical CAN Bus Layer**, *Proceedings of the 4th International CAN Conference*, 1997.
- [Boy99] S. Boyer, **SCADA: Supervisory Control and Data Acquisition**, ISA, 1999.
- [BP95] J. Bowles, C. Peláez, **Application of Fuzzy Logic to Reliability Engineering**, *Proceedings of the IEEE*, Vol. 83, No. 3, March, 1995.
- [BPB03] P. Marcelo, M. Poli, M. Barretto, **OPC and CORBA in Manufacturing Execution Systems: A Review**, *IEEE International Conference on Emerging Technologies and Factory Automation*, ETFA'03, 2003.
- [BPS+99] A. Burns, S. Punnekkat, L. Strigini, D. Wright, **Probabilistic Scheduling Guarantess for Fault-Tolerant Real-Time Systems**, *Proceedings of Dependable Computing for Critical Applications*, DCCA'99, 1999.
- [BPT+98] A. Bobbio, A. Puliafito, M. Telek, K. Trivedi, **Recent Developments in Non-Markovian Stochastic Petri Nets**, *Journal of Systems Circuits and Computers*, Vol. 8, No. 1, pp. 119-158, 1998.
- [Bri94] O. Bridal, **Reliability Estimates for Repairable Fault-Tolerant Systems**, *Proceedings of Nordic Seminar on Dependable Computing Systems*, 1994.
- [Bri95] O. Bridal, **A Methodology for Reliability Analysis of Fault-Tolerant Systems with Repairable Subsystems**, *Proceedings of the 2nd IMA Conference on the Mathematics of Dependable Systems*, 1995.
- [BSI00] **BSI Guide to the Evaluation of Fieldbus Protocols - Selecting the Best Fieldbus for your Application**, BSI Standards, 2000.
- [BT86] A. Bobbio, K. Trivedi, **An Aggregation Technique for the Transient Analysis of Stiff Markov Chains**, *IEEE Transactions on Computers*, Vol. C-35, No. 9, pp. 803-814, 1986.
- [BT95] A. Bobbio, M. Telek, **Markov Regenerative with Non-Overlapping Activity Cycles**, *Proceedings of the International Computer Performance and Dependability Symposium*, IPDS95, pp. 124-133, 1995.

-
- [BT97] A. Bobbio, M. Telek, **Specifications and Solution Techniques for Non-Markovian Stochastic Petri Nets**, *Technical Report*, 1997.
- [BT98] A. Bobbio, M. Telek, **Non-Exponential Stochastic Petri Nets: an Overview of Methods and Techniques**, *Computer Systems and Engineering*, Vol. 3, No. 6, pp. 339-351, 1998.
- [Buc04] P. Buchholz, **Adaptive Decomposition and Approximation for the Analysis of Stochastic Petri Nets**, *Proceedings of Dependable Systems and Networks-Performance and Dependability Symposium*, 2004.
- [Buc94] P. Buchholz, **Exact and Ordinary Lumpability in Finite Markov Chains**, *Journal of Applied Probability*, Vol. 31, pp. 59-75, 1994.
- [CAN02] **CANOpen - Application Layer and Communication Profile**, CiA DS 301, version 4.02, CiA, 2002.
- [CAN91] **CAN Specification - Version 2.0**, Robert Bosch GmbH, 1991.
- [Cass93] C. Cassandras, **Discrete Event Systems: Modeling and Performance Analysis**, Irwin and Aksen Associates, 1993.
- [CB03] A. Coccoli, A. Bondavalli, **Analysis of Safety Architectures**, *Proceedings of the 9th IEEE International Workshop on Object-Oriented Real-Time Dependable Systems, WORDS'03*, 2003.
- [CBC+93] G. Ciardo, A. Blakemore, P. Chimento, J. Muppla, K. Trivedi, **Automated Generation and Analysis of Markov Reward Models Using Stochastic Reward Nets**, *IMA Volumes in Mathematics and its Applications - Linear Algebra, Markov Chains and Queueing Models*, Vol. 48, pp. 145-191, Springer Verlag, 1993.
- [CC86] K. Chae, G. Clark, **System Reliability in the Presence of Common-Cause Failures**, *IEEE Transactions on Reliability*, Vol. R-35, No. 1, pp. 32-35, 1986.
- [CCD+01] G. Clark, T. Courtney, D. Daly, D. Deavours, S. Derisavi, J. Doyle, W. Sanders, P. Webster, **The Möbius Modeling Tool**, *Proceedings of the 9th International Workshop on Petri Nets and Performance Models*, 2001.
- [CDV95] G. Cena, L. Durante, A. Valenzano, **Standard field bus networks for industrial applications**, *Computer Standards & Interfaces*, Vol. 17, pp. 155-167, 1995.
- [CEE03] **Special Report: Machine Safety Networks**, Control Engineering Europe, 2003.
- [CGL94] G. Ciardo, R. German, C. Lindemann, **A Characterization of the Stochastic Process Underlying a Stochastic Petri Net**, *IEEE Transactions on Software Engineering*, Vol. 20, No.7, July, 1994.
- [Cha94] J. Charzinski, **Performance of the Error Detection Mechanisms in CAN**, *Proceedings of the 1st International CAN Conference*, 1994.
- [Cia87] G. Ciardo, **Toward a Definition of Modeling Power for Stochastic Petri Net Models**, *Proceedings of the International Workshop on Petri Nets and Performance Models*, IEEE Computer Society Press, pp. 54-62, 1987.
- [CJP97] C. Choi, B. Johnson, J. Profeta III, **Safety Issues in the Comparative Analysis of Dependable Architectures**, *IEEE Transactions on Reliability*, Vol. 46, No. 3, pp. 316-322, 1997.
-

Bibliografia

- [CL93] G. Ciardo, C. Lindemann, **Analysis of Deterministic and Stochastic Petri Nets**, *Proceedings of the International Workshop on Petri Nets and Performance Models, PNPM'93*, 1993.
- [CM00] S. Cavalieri, S. Monforte, **Petri Net Based Fault-Tolerance Evaluation of IEC 61158 Fieldbus Standard**, *Proceedings of the International Symposium in Industrial Electronics, ISIE'00*, 2000.
- [CMB+93] G. Chiola, M. Ajmone Marsan, G. Balbo, G. Conte, **Generalized Stochastic Petri Nets: A Definition at the Net Level and Its Implications**, *IEEE Transactions on Software Engineering*, Vol. 19, No. 2, February, 1993.
- [CMM99] S. Cavalieri, O. Mirabella, S. Monforte, **Enhancing Reliability in IEC/ISA Fieldbus**, *Computers Standards & Interfaces*, Vol. 21, pp. 217-240, 1999.
- [CMP+90] V. Catania, L. Milazzo, A. Puliafito, L. Vita, **Enhancing Reliability in an Industrial LAN: Design and Performability Evaluation**, *IEEE Transactions on Industrial Electronics*, Vol. 37, No. 6, pp. 433-441, 1990.
- [CMR+01] J. Cunha, R. Maia, M. Rela, J. Silva, **A Study of Failure Modes in Feedback Control Systems**, *Proceedings of the International Conference on Dependable Systems and Networks, DSN'01*, 2001.
- [CMS+90] G. Ciardo, R. Marie, B. Sericola, K. Trivedi, **Performability Analysis Using Semi-Markov Reward Processes**, *IEEE Transactions on Computers*, Vol. 39, No. 10, pp. 1251-1264, 1990.
- [CMS98] J. Carreira, H. Madeira, J. Silva, **Xception: A Technique for the Experimental Evaluation of Dependability in Moderns Computers**, *IEEE Transactions on Software Engineering*, Vol. 24, No. 2, pp. 125-136, 1998.
- [CMT+02a] S. Cavalieri, S. Monforte, E. Tovar, F. Vasques, **Multi-Master PROFIBUS-DP Modelling and Worst Case Analysis-Based Evaluation**, *Proceedings of the 15th IFAC World Congress on Automatic Control*, 2002.
- [CMT+02b] S. Cavalieri, S. Monforte, E. Tovar, F. Vasques, **Evaluating Worst Case Response Time in Mono and Multi-Master PROFIBUS-DP**, *Proceedings of the 4th IEEE International Conference on Factory Communications Systems, WFCS'02*, 2002.
- [Col87] C. Colburn, **The Combinatorics of Network Reliability**, Oxford University Press, 1987.
- [Con01] C. Constantinescu, **Dependability Analysis of a Fault-Tolerant Processor**, *Proceedings of the IEEE Pacific Rim International Symposium on Dependable Computing*, 2001.
- [Con03] C. Constantinescu, **Experimental Evaluation of Error-Detection Mechanisms**, *IEEE Transactions on Reliability*, Vol. 52, No. 1, pp. 53-57, 2003.
- [Con92] C. Constantinescu, **Predicting Performability of a Fault-Tolerant Microcomputer for Process Control**, *IEEE Transactions on Reliability*, Vol. 41, No. 4, pp. 558-564, 1992.
- [Con94] C. Constantinescu, **A Decomposition Method for Reliability Analysis of Real-Time Computing Systems**, *Proceedings of the Annual Reliability and Maintainability Symposium*, 1994.

-
- [CP01] P. Chevochot, I. Puaut, **Experimental Evaluation of the Fail-Silent Behaviour of a Distributed Real-Time Run-Time Support Built from COTS Components**, *Proceedings of the International Conference on Dependable Systems and Networks, DSN'01*, 2001.
- [CP04] J. Carvalho, P. Portugal, **A Fault-Injection Framework for Dependability Evalaution of Profibus-DP Networks**, *Technical Report, Universidade do Porto*, 2004.
- [CPA03] J. Carvalho, P. Portugal, A. Carvalho, **A Framework for Dependability Evaluation of PROFIBUS Networks**, *Proceedings of the International Symposium on Industrial Electronics, ISIE'03*, 2003.
- [Cris91] F. Cristian, **Understanding Fault-Tolerant Distributed Systems**, *Communications of ACM, Vol. 34, No. 2, pp. 56-78*, 1991.
- [CRS+97] J. Campelo, F. Rodríguez, J. Serrano, P. Gil, **Dependability Evaluation of Fault-Tolerant Architectures in Distributed Industrial Control Systems**, *Proceedings of the IEEE International Workshop on Factory Communication Systems, WFCS'97*, 1997.
- [CRS02] J. Cunha, M. Rela, J. Silva, **On the Use of Disaster Prediction for Failure-Tolerance in Feedback Control Systems**, *Proceedings of the International Conference on Dependable Systems and Networks, DSN'02*, 2002.
- [CT92] H. Choi, K. Trivedi, **Approximate Performance Models of Polling Systems Using Stochastic Petri Nets**, *Proceedings of the International Conference on Computer Communications*, 1992.
- [CT93] G. Ciardo, K. Trivedi, **A Decomposition Approach for Stochastic Reward Net Models**, *Performance Evaluation, Vol. 18, No. 1, pp. 37-59*, 1993.
- [CT95] C. Constantinecu, K. Trivedi, **Dependability Modeling of Real-Time Systems Using Stochastic Reward Nets**, *Microelectronics and Reliability, Vol. 35, No. 6, pp. 903-914*, 1995.
- [CT96] G. Ciardo, M. Tilgner, **On the Use of Kronecker Operators for the Solution of Generalized Stochastic Petri Nets**, *ICASE Report 96-35 CR-198336, NASA*, 1996.
- [CTR99] B. Conrard, J. Thiriet, M. Robert, **A Tool for the Evaluation of Fieldbuses Reliability for Intelligence Distributed Automation System Design**, *Advances in Manufacturing, Decision, Control and Information Technology, pp. 157-167, Springer-Verlag*, 1999.
- [CV04] G. Cena, A. Valenzano, **Operating Principles and features of CAN Networks**, *The Industrial Information Technology Handbook*, 2004.
- [CWT98] H. Choi, W. Wang, K. Trivedi, **Analysis of Conditional MTTF of Fault-Tolerant Systems**, *Microelectronics and Reliability, Vol. 38, No. 3, pp. 393-401*, 1998.
- [CYR+99a] J. Campelo, P. Yuste, P. Gil, J. Serrano, **Hierarchical Reliability and Safety Models of Fault Tolerant Distributed Industrial Control Systems**, *Proceedings of 18th Conference on Computer Safety, Reliability and Security, SAFECOMP'99*, 1999.
- [CYR+99b] J. Campelo, P. Yuste, P. Gil, J. Serrano, **Dependability Evaluation of Fault Tolerant Distributed Industrial Control Systems**, *Proceedings of the International Workshop on Parallel and Distributed Real-Time Systems, WPDRTS'99*, 1999.
-

Bibliografia

- [CZ96] G. Ciardo, R. Zijal, **Well-defined stochastic Petri Nets**, *Proceedings 4th International Workshop on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, MASCOTS96*, pp. 278-284, 1996.
- [DA92] R. David, H. Alla, **Petri Nets and Grafcet: Tools for Modeling Discrete Event Systems**, Prentice Hall, 1992.
- [DBB92] J. Dugan, S. Bavuso, M. Boyd, **Dynamic Fault-Tree Models for Fault-Tolerant Computer System**, *IEEE Transactions on Reliability*, Vol. R-41, No. 9, pp. 363-377, 1992.
- [DDH01] J.-D. Decotignie, P. Dallemagne, A. El-Hoiydi, **Architectures for the Interconnection of Wireless and Wireline Fieldbusses**, *IFAC International Conference on Fieldbus Systems and their Applications - FeT01*, 2001.
- [DDP88] J. Daigle, A. Seidmann, J. Pimentel, **Communications for Manufacturing: An Overview**, *IEEE Network*, Vol. 2, No. 3, pp. 6-12, 1988.
- [Dec01] J.-D. Decotignie, **A Perspective on Ethernet-TCP/IP as a Fieldbus**, *IFAC International Conference on Fieldbus Systems and their Applications - FeT01*, 2001.
- [DGK+02] J. Díaz, D. Garcia, K. Kim, C. Lee, L. Bello, J. López, S. Min, O. Mirabella, **Stochastic Analysis of Periodic Real-Time Systems**, *Proceedings of the 23rd IEEE Real-Time Systems Symposium, RTSS'02*, 2002.
- [Dhi79] B. Dhillon, **A Hazard Rate Model**, *IEEE Transactions on Reliability*, Vol. 28, No. 150, 1979.
- [Dhi99] B. Dhillon, **Design Reliability: Fundamentals and Applications**, CRC Press, 1999.
- [DL94] J. Dugan, M. Lyu, **System Reliability Analysis of an N-Version Programming Application**, *IEEE Transactions on Reliability*, Vol. 43, No. 4, pp. 513-519, 1994.
- [DNE00] **EN 50325-2 - Industrial Communications Subsystem Based on ISO 11898 (CAN) for Controller-Device Interfaces - Part 2: DeviceNet**, CENELEC, 2000.
- [DS81] B. Dhillon, C. Singh, **Engineering Reliability - New Techniques and Applications**, John Wiley & Sons, 1981.
- [DS99] D. Deavours, W. Sanders, **An Efficient Well-Specified Check**, *Proceedings of the 8th International Workshop on Petri Nets and Performance Models, PNPM'99*, 1999.
- [DSP88] J. Daigle, A. Seidmann, J. Pimentel, **Communications for Manufacturing: An Overview**, *IEEE Network*, Vol. 2, No. 3, pp. 6-12, 1988.
- [DT89] J. Dugan, K. Trivedi, **Coverage Modeling for Dependability Analysis of Fault-Tolerant Systems**, *IEEE Transactions on Computers*, Vol. 38, No. 6, pp. 775-787, 1989.
- [DTG+84] J. Bechta Dugan, K. Trivedi, R. Geist, V. Nicola, **Extended Stochastic Petri Nets: applications and analysis**, *Proceedings PERFORMANCE'84*, 1984.
- [DTS+86] J. Dugan, K. Trivedi, M. Smotherman, R. Geist, **The Hybrid Automated Reliability Predictor**, *AIAA Journal of Guidance, Control and Dynamics*, Vol. 9, No. 3, pp. 319-331, 1986.

-
- [DV98] C. Demartini, A. Valenzano, **The EN 50170 Standard for a European Fieldbus**, *Computer Standards & Interfaces*, Vol. 19, pp. 257-273, 1998.
- [Dyr89] D. Dyer, **Unification of Reliability / Availability / Repairability Models for Markov Systems**, *IEEE Transaction on Reliability*, Vol. 38, No. 2, pp. 246-254, June, 1989.
- [Eco04] M. Economou, **The Merits and Limitations of Reliability Predictions**, *Proceedings of the Annual Reliability and Maintainability Symposium*, 2004.
- [EEH+01] H. Edler, J. Eriksson, J. Hedberg, H. Sjöström, **PALBUS - Tasks 10.1 to 10.13**, *Technical Reports, Swedish National Testing and Research Institute*, 2001.
- [EN02] **EN 50325-4 :Industrial Communications Subsystem Based on ISO 11898 (CAN) for Controller-Device Interfaces - Part 4: CANopen**, CENELEC, 2002.
- [EN96a] **EN 50170 - General Purpose Field Communication System - Volume 2/3 (PROFIBUS)**, CENELEC, 1996.
- [EN96b] **EN 50170, Volume 2/3 (PROFIBUS) - General Purpose Field Communication System - Part 8-2 - User Specifications**, CENELEC, 1996.
- [EN99] **EN 50170/A2 - General Purpose Field Communication System**, CENELEC, 1999.
- [Eth01] **EtherNet/IP Adaptation of CIP Specification**, Open DeviceNet Vendor Association & ControlNet International, 2001.
- [Fel02] M. Felser, **The Fieldbus Standards: History and Structures**, *MICROSWISS Technology Leadership*, 2002.
- [FFS+02] A. Flammini, P. Ferrari, E. Sisinni, D. Marioli, A. Taroni, **Sensor Interfaces: from Field-Bus to Ethernet and Internet**, *International Journal of Sensors and Actuators*, Vol. 101, pp. 194-202, 2002.
- [FKK99] N. Fota, M. Kaaniche, K. Kanoun, **Incremental Approach for Building Stochastic Petri Nets for Dependability Modeling**, *LAAS Report 99073*, 1999.
- [Fle04] **FlexRay Communication System - Protocol Specification - Version 2.0**, FlexRay Consortium, 2004.
- [FMD+00] T. Führer, B. Müller, W. Dieterle, F. Hartwich, R. Hugel, M. Walther, **Time Triggered Communication on CAN**, *Proceedings 7th International CAN Conference*, 2000.
- [FNS01] J. Froidevaux, O. Nick, M. Suzan, **Use of Fieldbus in Safety Related Systems, an Evaluation of WorldFIP According to Proven-in-use Concept of IEC 61508**, *IFAC International Conference on Fieldbus Systems and their Applications - Fet01*, 2001.
- [FOF+04] J. Ferreira, A. Oliveira, P. Fonseca, L. Almeida, J. Fonseca, **An Experiment to Assess Bit Error Rate in CAN: Early Results**, *Proceedings of the 3rd International Workshop on Real-Time Networks (RTN)*, 2004.
- [Fol99] P. Folkesson, **Assessment and Comparison of Physical Fault Injection Techniques**, *Phd Thesis, Chalmers University of Technology*, 1999.
- [FPT+98] R. Fricks, A. Puliafito, M. Telek, K. Trivedi, **Applications of Non-Markovian Stochastic Petri Nets**, *Performance Evaluation*, Vol. 26, No. 2, pp. 15-27, 1998.
-

Bibliografia

- [FS02] M. Falser, T. Sauter, **The Fieldbus War: History or Short Break Between Battles?**, *4th IEEE Workshop on Factory Communication Systems, WFCS'04*, 2002.
- [FS04] M. Falser, T. Sauter, **Standardization of Industrial Ethernet - the Next Battlefield**, *IEEE International Workshop on Factory Communication Systems, WFCS'2004*, 2004.
- [FT97] R. Fricks, K. Trivedi, **Modeling Failure Dependencies in Reliability Analysis Using Stochastic Petri Nets**, *Proceedings of the European Simulation Multi-Conference, ESM'97*, 1997.
- [FTCS82] **Special Session. Fundamental concepts on fault tolerance**, 12th International Fault-Tolerant Computing Symposium, 1982.
- [FTP+98] R. Fricks, M. Telek, A. Puliafito, K. Trivedi, **Markov Renewal Theory Applied to Performability Evaluation**, *Modeling and Simulation of Advanced Computer Systems: Applications and Systems*, Gordon and Breach Publishers, 1998.
- [GCB98] F. Grandoni, S. Chiaradonna, A. Bondavalli, **A new Heuristic to Discriminate between Transient and Intermittent Faults**, *Proceedings of the 3rd International Symposium on High-Assurance Systems Engineering, HASE'98*, 1998.
- [Gei91] R. Geist, **Extended Behavioral Decomposition for Estimating Ultrahigh Reliability**, *IEEE Transactions on Reliability*, Vol. 40, No. 1, pp. 22-27, 1991.
- [Ger00] R. German, **Performance Analysis of Communication Systems: Modeling with Non-Markovian Stochastic Petri Nets**, John Wiley & Sons, 2000.
- [Ger99] R. German, **Markov Regenerative Stochastic Petri Nets with General Execution Policies: Supplementary Variables Analysis and a Prototype Tool**, *Performance Evaluation*, Vol. 39, pp. 165-188, 1999.
- [GLT95] R. German, D. Logothetis, K. Trivedi, **Transient Analysis of Markov Regenerative Stochastic Petri Nets: A Comparison of Approaches**, *6th International Conference on Petri Nets and Performance Models, PNPM95*, pp. 103-112, 1995.
- [GN01] B. Gaujal, N. Navet, **Fault Confinement Mechanisms on CAN: Analysis and Improvements**, *Proceedings 4th IFAC International Conference on Fieldbus Systems and their Applications, FeT01*, 2001.
- [GN02] B. Gaujal, N. Navet, **Fault Confinement Mechanisms on CAN: Analysis and Improvements**, *Rapport de Recherche, INRIA*, 2002.
- [GNP02a] C. Guerrero, G. Rodríguez-Navas, J. Proenza, **Design and Implementation of a Redundancy Manager for Triple Redundant CAN Controllers**, *Proceedings of the 28th Annual Conference on Industrial Electronics, IECON'02*, 2002.
- [GNP02b] C. Guerrero, G. Rodríguez-Navas, J. Proenza, **Hardware Support for Fault Tolerance in Triple Redundant CAN Controllers**, *Proceedings of the 9th International Conference on Electronics, Circuits and Systems, IECON'02*, 2002.
- [Gob98] W. Goble, **Control Systems Safety Evaluation & Reliability**, ISA Press, 1998.
- [Goe85] A. Goel, **Software Reliability Models: Assumptions, Limitations, and Applicability**, *IEEE Transactions on Software Engineering*, Vol. SE-11, No. 12, pp. 1411-1423, 1985.

-
- [Gro00] M. Groover, **Automation Production Systems, and Computer-Integrated Manufacturing**, Prentice Hall, 2000.
- [GS89] R. Geist, M. Smotherman, **Ultrahigh Reliability Estimates Through Simulation**, *Proceedings of the Annual Reliability and Maintainability Symposium*, 1989.
- [GS97] R. Guerraoui, A. Schiper, **Software-Based Replication for Fault-Tolerance**, *IEEE Computer*, Vol.30, No. 4, pp. 68-74, 1997.
- [GT99] R. German, M. Telek, **Formal Relation of Markov Renewal Theory and Supplementary Variables in the Analysis of Stochastic Petri Nets**, *Proceedings of the 8th International Workshop on Petri Net and Performance Models, PNPM'99*, 1999.
- [Haa02] P. Haas, **Stochastic Petri Nets: Modelling, Stability, Simulation**, Springer Verlag, 2002.
- [Hav99] B. Haverkort, **Performance Evaluation of Polling-Based Communication Systems Using SPNs**, *Applications of Petri Nets to Communication Networks*, pp. 176-209, Springer-Verlag, 1999.
- [Hef97] D. Heffernan, **A Technical Overview of Fieldbus Developments from Origins to Present Day Standards**, Technical Report, Department of Electronic and Computer Engineering, University of Limerick, 1997.
- [Her02] U. Herzog, **Formal Methods for Performance Evaluation**, *LNCS 2090: Lectures on Formal Methods and Performance Analysis*, pp. 1-37, Springer-Verlag, 2002.
- [HHK+03] J. Hérard, J. Hedberg, M. Kivipuro, T. Malm, H. Edler, H. Sjöström, T. Strawinski, **Validation of Communication in Safety-Critical Control Systems**, *Nordtest Technical Report TR-543*, 2003.
- [Hil00] M. Hiller, **Executable Assertions for Detecting Data Errors in Embedded Control Systems**, *Proceedings of the International Conference on Dependable Systems and Networks, DSN'00*, 2000.
- [HKN+98] G. Horton, V. Kulkarni, D. Nicol, K. Trivedi, **Fluid Stochastic Petri Nets: Theory, Applications and Solution Techniques**, *European Journal of Operations Research*, Vol. 105, No. 1, pp. 184-201, 1998.
- [HLB+97] H. Hansson, H. Lawson, O. Bridal, C. Eriksson, Sven Larsson, He Lon, M. Stromberg, **BASEMENT: An Architecture and Methodology for Distributed Automotive Real-Time Systems**, *IEEE Transactions on Computers*, Vol. 46, No. 9, pp. 1016 - 1027, 1997.
- [HMC97] S. Haddad, P. Moreaux, G. Chiola, **Efficient Handling of Phase-type Distributions in Generalized Stochastic Petri Nets**, *Application and Theory of Petri Nets (18th International Conference), Lecture Notes in Computer Science*, Vol. 1248, pp. 175-194, 1997.
- [HMR+01] B. Haverkort, R. Marie, G. Rubino, K. Trivedi, **Performability Modeling Tools and Techniques**, John Wiley & Sons, 2001.
- [HN96] B. Haverkort, I. Niemegeers, **Performability Modelling Tools and Techniques**, *Performance Evaluation*, Vo. 25, pp. 17-40, 1996.
-

Bibliografia

- [HNN+02] H. Hansson, T. Nolte, C. Norström, S. Punnekkat, **Integrating Reliability and Timming Analysis of CAN-Based Systems**, *IEEE Transactions on Industrial Electronics*, Vol. 49. No. 6, December, 2002.
- [HNP00] H. Hansson, C. Norström, S. Punnekkat, **Reliability Modelling of Time-Critical Distributed Systems**, *Formal Techniques in Real-Time and Fault-Tolerant Systems, Lecture Notes in Computer Science*, Vol. 1926, Springer Verlag, 2000.
- [HNP01] H. Hansson, C. Norström, S. Punnekkat, **A Simulation based Approach for Estimating the Reliability of Distributed Real-Time Systems**, *Proceedings of the 8rd IEEE International Conference in Emerging Technologies and Factory Automation, ETFA'01*, 2001.
- [Hod02] B. Hodges, **Wireless Ethernet in Factory Floor and Industrial Applications**, *ISA Industrial In-plant Communications Conference*, 2002.
- [HON04] **Industrial Safety Products**, Honeywell, 2004.
- [Hor02] G. Horton, **A New Paradigm for the Numerical Simulation of Stochastic Petri Nets with General Firing Times**, *Proceedings of the European Simulation Symposium*, 2002.
- [HPS+00] A. Horváth, A. Puliafito, M. Scarpa, M. Telek, **Analysis and Evaluation of non-Markovian Stochastic Petri Nets**, *Proceedings of the 11th International Conference in Computer Performance Evaluation: Modelling Techniques and Tools, TOOLS'00*, 2000.
- [HSZ+00] C. Hirel, R. Sahner, X. Zang, K. Trivedi, **Reliability and Performability Modeling Using SHARPE 2000**, *Proceedings of the 11th International Conference in Computer Performance Evaluation: Modelling Techniques and Tools, TOOLS'00*, 2000.
- [HTT02] **SPNP: Stochastic Petri Nets. Version 6.0**, 2002.
- [Hug01] T. A. Hughes, **Programmable Controllers 3rd Edition**, ISA Press, 2001.
- [HW81] P. Heidelberger, P. Welch, **A Spectral Method for Confidence Interval Generation and Run Length Control in Simulations**, *Communications of the ACM*, Vol. 21, No. 4, pp. 223-245, 1981.
- [Ida02] **Interface for Distributed Automation: Architecture Description and Specification**, IDA, 2002.
- [IEB04] <http://ethernet.industrial-networking.com>,
- [IEC00] **IEC 61499 - Function Blocks for Industrial-Process Measurement and Control Systems**, IEC, 2000.
- [IEC02] **IEC 62280-1 - Railway Applications – Communications, Signalling and Processing Systems – Part 1: Safety Related Communication in Closed Transmission Systems**, IEC, 2002.
- [IEC03a] **IEC 61158 - Digital data communications for measurement and control - Fieldbus for use in Industrial Control Systems**, IEC, 2003.
- [IEC03b] **IEC 61784-1 - Digital data communications for measurement and control - Part 1: Profile sets for continuous and discrete manufacturing relative to fieldbus use in industrial control systems**, IEC, 2003.
- [IEC03c] **IEC 61804 - Function Blocks (FB) for Process Control**, IEC, 2003.

-
- [IEC04a] IEC 61508 - Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems, IEC, 2004.
- [IEC04b] IEC 61511- Functional Safety - Safety Instrumented Systems for the Process Industry Sector, IEC, 2004.
- [IL02] F. Iwanitz, J. Lange, **OPC: Fundamentals, Implementation and Application**, Hüthig Verlag Heidelberg, 2002.
- [Int03] **White paper: INTERBUS Safety**, INTERBUS Club Deutschland e. V., 2003.
- [ISO03] ISO 9506 - Industrial automation systems - Manufacturing Message Specification, ISO, 2003.
- [ISO99] ISO 11898 - Road Vehicles - Controller Area Network, ISO, 1999.
- [ISO99b] ISO 11898 - Road Vehicles - Controller Area Network - All Parts, ISO, 1999.
- [ISA02] Safety Instrumented Functions (SIF) - Safety Integrity Level (SIL) Evaluation Techniques Part 1 to 5, ISA Technical Report TR84.00.02, 2002.
- [ISO03a] ISO 11898-2 - Road Vehicles - Controller Area Network (CAN) - Part 2: High-Speed Medium Access Unit, ISO, 2003.
- [ISO03b] ISO 11992-1 - Road Vehicles - Interchange of Digital Information on Electrical Connections Between Towing and Towed Vehicles - Part 1: Physical and Data-Link Layers, ISO, 2003.
- [ISO04a] ISO/DIS 11898-3 - Road Vehicles - Controller Area Network (CAN) - Part 3: Low-Speed, Fault-Tolerant, Medium Dependent Interface, ISO, 2004.
- [ISO04b] ISO 11898-4 - Road Vehicles - Controller Area Network (CAN) - Part 4: Time-Triggered Communication, ISO, 2004.
- [ISO94] ISO/IEC 7498-1 - Information technology - Open Systems Interconnection - Basic Reference Model: The Basic Model, ISO, 1994.
- [ISO99a] ISO 13849 - Safety of Machinery - Safety-Related Parts of Control Systems, ISO, 1999.
- [JA92] G. Juanole, Y. Atamna, **Modelling Communications in the Fieldbus FIP with Stochastic Timed Petri Nets**, *LAAS Report 92165*, 1992.
- [JAB94] G. Juanole, Y. Atamna, N. Berge, **Modeling and Evaluation of the Filedbus FIP with Stochstics Timed Petri Nets**, *LAAS Report 94240*, 1994.
- [Jal94] P. Jalote, **Fault Tolerance in Distributed Systems**, Prentice Hall, 1994.
- [JBA+98] J. Rufino, P. Veríssimo, G. Arroz, C. Almeida, L. Rodrigues, **Fault-Tolerant Broadcasts in CAN**, *Proceedings of the 28th IEEE International Symposium on Fault-Tolerant Computing*, 1998.
- [JH02] U. Juelly, D. Hablawetz, **New Technologies for Safety Related and Fault Tolerant Control Systems in a Open Communication Environment**, *2nd Fieldbus International Exhibition*, 2002.
- [JHC+02] A. Jhumka, M. Hiller, V. Claesson, N. Suri, **On Systematic Design of Globally Consistent Executable Assertions in Embedded Software**, *Proceedings of the International Conference on Languages, Compilers, and Tools for Embedded Systems*, 2002.
-

Bibliografia

- [JLK92] J.-C. Laprie, K. Kanoun, **X-Ware Reliability and Availability Modeling**, *IEEE Transactions on Software Engineering*, Vol. 18, No. 2, pp. 130-147, 1992.
- [JM88] A. Johnson, M. Malek, **Survey of Software Tools for Evaluating Reliability, Availability, and Serviceability**, *ACM Computing Surveys*, Vol. 20, No. 4, pp. 227-269, 1998.
- [JNL03] F. Jumel, N. Navet, F. Simonot-Lion, **Influence des Performances d'une Architecture Informatique sur la Fiabilité des Systèmes Échantillonnés**, *Proceedings of the Real-Time Symposium*, 2003.
- [Joh99] U. Johannsmeyer, **Fieldbus for Process Automation in Hazardous Locations - Actual Developments**, *IEE Conference on Explosion Safety in Hazardous Areas*, 1999.
- [JSW04] U. Jecht, W. Stripf, P. Wenzel, **PROFIBUS - Open Solutions for the World of Automation**, *The Industrial Information Technology Handbook*, CRC Press, 2004.
- [Kao97] E. Kao, **An Introduction to Stochastic Processes**, Duxbury Press, 1997.
- [KB00] K. Kanoun, M. Borrel, **Fault-Tolerant System Dependability - Explicit Modeling of Hardware and Software Component-Interactions**, *IEEE Transactions on Reliability*, Vol. 49, No. 4, pp. 363-375, 2000.
- [KBJ00] L. Kaufman, S. Bhide, B. Johnson, **Modeling of Common-Mode Failures in Digital Embedded Systems**, *Proceedings of the Annual Reliability and Maintainability Symposium*, 2000.
- [KDK+93] H. Kopetz, A. Damm, C. Koza, M. Mulazzani, W. Schwabl, C. Senft, R. Zailinger, **Distributed Fault-Tolerant Real-Time Systems: The MARS Approach**, *Advances in Real-Time Systems*, pp. 486-501, IEEE Press, 1993.
- [Kel94] C. Kelling, **Control Variates Selection Strategies for Timed Petri Nets**, *Proceedings of the European Simulation Symposium*, 1994.
- [Kel95] C. Kelling, **TimeNET-Sim- A Parallel Simulator for Stochastic Petri Nets**, *Proceedings of the 28th IEEE Annual Simulation Symposium*, 1995.
- [Kel96] C. Kelling, **A Framework for Rare Event Simulation of Stochastic Petri Nets Using "RESTART"**, *Proceedings of the Winter Simulation Conference*, 1996.
- [KH96] C. Kelling, G. Hommel, **Rare Event Simulation with an Adaptive RESTART Method in a Petri Net Modeling Environment**, *Proceedings of 4th WPDRTS*, 1996.
- [KKG+90] H. Kopetz, H. Kantz, G. Grünsteidl, P. Purchner, J. Reisinger, **Tolerating Transient Faults in MARS**, *Proceedings of the 20th International Symposium in Fault-Tolerant Computing, FTCS-20*, 1990.
- [KKW03] G. Klutke, P. Kiessler, M. Wortman, **A Critical Look at the Bathtub Curve**, *IEEE Transactions on Reliability*, Vol. 52, No. 1, pp. 125-129, 2003.
- [KM04] W. Kastner, F. Masilko, **EDDL inside FDT/DTM**, *Proceedings of the IEEE International Conference on Factory Communication Systems, WFCS'04*, 2004.
- [Kop01] H. Kopetz, **The Three Interfaces of a Smart Transducer**, *IFAC International Conference on Fieldbus Systems and their Applications - FeT'01*, 2001.

-
- [Kop97] H. Kopetz, **Real-Time Systems: Design Principles for Distributed Embedded Applications**, Kluwer Academic Publishers, 1997.
- [Kor99] S. Koriem , **Fast and Simple Decomposition Techniques for the Reliability Analysis of Interconnection Networks**, *The Journal of Systems and Software*, Vol. 45, pp. 155-171, 1999.
- [KP91] K. Kanoun, D.Powell, **Dependability Evaluation of Bus and Ring Communicaion Topologies for DELTA-4 Distributed Fault-Tolerant Architecture**, *Proceedings of 10th Symposium on Reliable Distributed Systems*, pp. 130-141, 1991.
- [KP95] S. Koubias, G. Papadopoulos, **Modern fieldbus communication architectures for real-time industrial applications**, *Computers in Industry*, Vol. 26, pp. 243-253, 1995.
- [Kri04] U. Krieger, **Numerical Solution Methods for Markov Chains**, *Technical Report, Otto-Friedrich-University Bamberg*, 2004.
- [KS00] H. Kim, K. Shin, **Evaluation of Fault Tolerance Latency from Real-Time Application's Perspectives**, *IEEE Transactions on Computers*, Vol. 49, No.1, pp. 55-64, 2000.
- [KS94] H. Kim, K. Shin, **On the Maximum Feedback Delay in Linear/Nonlinear Control System with Input Disturbances Caused by Controller-Computer Failures**, *IEEE Transactions on Control Systems Technology*, Vol. 2 No. 2, pp. 110-122, 1994.
- [KT91] H. Kantz, K. Trivedi, **Reliability Modeling of the MARS System: A Case Study in the Use of Different Tools and Techniques**, *Proceedings of the 4th International Workshop on Petri Nets and Performance Models, PNPM'91*, 1991.
- [Kul95] V. Kulkarni, **Modeling and Analysis of Stochastic Systems**, Chapman & Hall, 1995.
- [KWF+88] R. Kieckhafer, C. Walter, A. Finn, P. Thambidurai, **The MAFT Architecture for Distrubuted Fault Tolerance**, *IEEE Transactions on Computers*, Vol. 37, No. 4, pp. 398-405, 1988.
- [KWS00] H. Kim, A. White, K. Shin, **Effects of Electromagnetic Interference on Controller-Computer Upsets and System Stability**, *IEEE Transactions on Control Systems Technology*, Vol. 8, No. 2, pp. 351-357, 2000.
- [KWS98] H. Kim, A. White, K. Shin, **Reliability Modeling of Hard-Real-Time Systems**, *Proceedings of the Fault-Tolerant Computers Systems Conference, FTCS-28*, 1998.
- [LA90] P. Lee, T. Anderson, **Fault Tolerance: Principles and Practice**, 2nd edition, Springer-Verlag, 1990.
- [LA90] P. Lee, T. Anderson, **Fault Tolerance - Principles and Practices**, Springer-Verlag, 1990.
- [LAB+95] J-C. Laprie, J. Arlat, C. Béounes, K. Kanoun, **Definition and Analysis of Hardware-and-Software Fault-Tolerant Architectures**, *Predictably Dependable Computing Systems*, pp. 103-122, 1995.
-

Bibliografia

- [Lap85] J.-C. Laprie, **Dependable computing and fault tolerance: concepts and terminology**, *15th International Fault-Tolerant Computing Symposium*, 1985.
- [Lap92] J.-C. Laprie, **Dependability: Basic Concepts and Terminology**1992, Springer-Verlag, 1992.
- [Lap93] J.C.-Laprie, **Dependability of Computer Systems: from Concepts to Limits**, *Proceedings of SAFECOMP'93*, 1993.
- [Lap95] J.-C. Laprie, **Dependability -Its Attributes, Impairments and Means**, Predictably Dependable Computing Systems, ESPRIT Basic Research Series, Springer, 1995.
- [LB01] G. Lima, A. Burns, **An Effective Schedulability Analysis for Fault-Tolerant Hard Real-Time Systems**, *Proceedings of the 13th Euromicro Conference on Real-Time Systems, ECRTS'01*, 2001.
- [Lev01] P. Leviti, **IEC61158: An Offense to Technicians**, *IFAC International Conference on Fieldbus Systems and their Applications - FeT01*, 2001.
- [Lew01] R. Lewis, **Modelling Control Systems using IEC 61499: Applying function blocks to distributed systems**, IEE Press, 2001.
- [LGT+85] W. Lee, D. Grosh, F. Tillman, C. Lie, **Fault Tree Analysis, Methods and Applications - A Review**, *IEEE Transactions on Reliability*, Vol. R-34, No. 3, pp. 194-203, 1985.
- [LH94] J. Lala, R. Harper, **Architectural Principles for Safety-Critical Real-Time Applications**, *Proceedings of the IEEE*, Vol. 82, No. 1, pp.25-40, 1994.
- [Lia01] F. Lian, **Analysis, Design, Modeling, and Control of Networked Control Systems**, *PhD Thesis, University of Michigan*, 2001.
- [Lin95] C. Lindemann, **A Software Package for Efficient Solution of Deterministic and Stochastic Petri Nets**, *Performance Evaluation*, Vol. 22, pp. 3-21, 1995.
- [Lin98] C. Lindemann, **Performance Modelling with Deterministics and Stochastic Petri Nets**, John Wiley & Sons, 1998.
- [LK00] A. Law, W. Kelton, **Simulation Modeling and Analysis**, 3rd Edition, McGraw-Hill, 2000.
- [LPS02] M. Lobashov, G. Pratl, T. Sauter, **Applicability of Internet Protocols for Fieldbus Access**, *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA'02*, 2002.
- [LS90] A. Laemmel, M. Shooman, **Bounding and Approximating Markov Models**, *Proceedings of the Annual Reliability and Maintainability Symposium*, 1990.
- [LS96] C. Lindemann, G. Shedler, **Numerical Analysis of Deterministic and Stochastic Petri Nets with Concurrent Deterministic Transitions**, *Performance Evaluation*, Vol. 27&28, pp. 565-582, 1996.
- [LSD03] E. Lacroix, R. St-Denis, **Web Technologies in Support of Virtual Manufacturing Systems**, *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA'03*, 2003.

- [LTP95] D. Logothetis, K. Trivedi, A. Puliafito, **Markov Regenerative Models**, *Proceedings of IEEE International Conference on Parallel and Distributed Systems*, 1995.
- [LYT03] M. Lanus, L. Yin, K. Trivedi, **Hierarchical Composition and Aggregation of State-Based Availability and Performability Models**, *IEEE Transactions on Reliability*, Vol. 52, No.1, pp. 44-52, 2003.
- [MAA94] S. Metge, M. Aguera, J. Arlat, S. Bachmann, C. Bourdeau, J.-E. Doucet, K. Kanoun, J.-C. Laprie, J. de Souza, D. Powell, P. Spiesser, **SURF-2. Outil d'Evaluation de la Suréte de Fonctionnement par Chaines de Markov et Reseaux de Petri Stochastiques**, *LAAS Report 94088*, 1994.
- [Mac01] S. Mackay, **Foundaton Fieldbus High Speed Ethernet (HSE) and TCP/IP**, Foundation Fieldbus End User Council Australia Inc., 2001.
- [Mag95] A. Magalhães, **Estabilização dos Controladores de Tempo-Real Através da Complacência Temporal dos Objectos Controlados**, Tese de Doutoramento, Faculdade de Engenharia da Universidade do Porto, 1995.
- [Map04] <http://www.maplesoft.com>, 2004.
- [MAP85] **Manufacturing Automation Protocol Version 2.1**, General Motors Corporation, 1985.
- [Math04] <http://www.wolfram.com>, 2004.
- [Mau00] G. Mauri, **Integrating Safety Analysis Techniques, Supporting Indentification of Common Cause Failures**, *PhD Thesis, University of York*, 2000.
- [MAV+00] S. Monforte, M. Alves, F. Vasques, E. Tovar, **Designing Real-Time Systems Based on Mono-Master Profibus-DP Networks**, *Proceeding of the 16th IFAC Workshop on Distributed Computer Control Systems, DCCS'00*, 2000.
- [MB99] I. Mura, A. Bondavalli, **Hierarchical Modeling & Evaluation of Phased-Mission Systems**, *IEEE Transactions on Reliability*, Vol. 48, No. 4, pp.360-368, 1999.
- [MBB+89] M. Ajmone Marsan, G. Balbo, A. Bobbio, G. Chiola, G. Conte, A. Cumani, **The effect of execution policies on the semantics and analysis of stochastic Petri nets**, *IEEE Transactions on Software Engineering*, Vo. 15, No.7, July, pp. 832-846, 1989.
- [MBC+95] M. Ajamone Marsan, G. Balbo, G. Conte, S. Donatelli, G. Francesschinis, **Modelling with Generalized Stochastic Petri Nets**, Jonh Wiley & Sons, 1995.
- [MBZ+99] I. Mura, A. Bondavalli, X. Zang, K. Trivedi, **Dependability Modeling and Evaluation of Phased Mission Systems: a DSPN Approach**, *Proceedings of the Dependable Computing for Critical Applications, DCCA'99*, 1999.
- [MC87] M. Ajmone Marsan, G. Chiola, **On Petri Nets with Deterministic and Exponentially Distributed Firing Times**, *Advances in Petri Nets 1987, Lecture Notes in Computer Science 266*, pp. 132-145, 1987.
- [MCB84] M. Ajmone Marsan, G. Conte, G. Balbo, **A Class of Generalized Stochastic Petri Nets for Performance Evaluation of Multiprocessor Systems**, *ACM Transactions on Computer Systems*, Vol. 2, No. 2, pp. 93-122, 1984.

Bibliografia

- [MCT94] J. Muppala, G. Ciardo, K. Trivedi, **Stochastic Reward Nets for Reliability Prediction**, *Communications in Reliability, Maintainability and Serviceability*, Vol. 1, No. 2, pp. 9-20, 1994.
- [Mey80] J. Mayer, **On Evaluating the Performability of Degradable Computing Systems**, *IEEE Transactions on Computers*, Vol. C-29, No. 8, pp. 720-731, 1980.
- [MFT00] J. Muppala, R. Fricks, K. Trivedi, **Techniques for System Dependability Evaluation**, *Computational Probability*, pp. 445-480, Kluwer Academic Publishers, 2000.
- [MGM+99] R. Martinez, P. Gil, G. Martin, C. Perez, J. Serrano, **Experimental Valiadation of High-Speed Fault-Tolerant Systems Using Physical Fault Injection**, *Proceedings of the International Working Conference on Dependable Computing for Critical Applications*, DCCA-7, 1999.
- [MH03] S. Molnar, G. Horton, **Proxel-Based Simulation of Stochastic Petri Nets Containng Immediate Transitions**, *Electronic Notes in Theoretical Computer Science*, Vol. 85, No. 4, 2003.
- [MIL80] **MIL-STD-1629A - Failure Mode and Effects Analysis**, National Technical Information Service, 1980.
- [MIL91] **MIL-HDBK-217F - Reliability Prediction of Electronic Equipment**, Departament of Defense, USA, 1991.
- [MK03] J. Morris, P. Koopman, **Critical Message Integrity over a Shared Network**, *Proceedings of the 5th IFAC International Conference on Fieldbus Systems and their Applications*, FeT03, 2003.
- [ML96] J. Muppala, C. Lin, **Dependability Analysis of Large-Scale Distributed Systems Using Stochastic Petri Nets**, *IEEE International Conference on Systems, Man, and Cybernetics*, 1996.
- [MMT94] M. Malhotra, J. Muppala, K. Trivedi, **Stiffness-Tolerant Methods for Transient Analysis of Stiff Markov Chains**, *Microelectronics and Reliability*, Vol. 34, No. 11, pp. 1825-1841, 1994.
- [MMT96] J. Muppala, M. Malhotra, K. Trivedi , **Markov Dependability Models of Complex Systems: Analysis Techniques**, *Reliability and Maintenance of Complex Systems*, pp. 442-486, Springer-Verlag, 1996.
- [Mod02] **MODBUS Messaging on TCP/IP Implementation Guide**, MODBUS.ORG, 2002.
- [Mol82] M. Molloy, **Performance Analysis Using Stochastic Petri Nets**, *IEEE Transactions on Computers*, Vol. 31, No.9, pp. 913-917, 1982.
- [Mol85] M. Molloy, **Discrete Time Stochastic Petri Nets**, *IEEE Transactions in Software Engineering*, Vol. 11, No. 4, pp. 417-423, 1985.
- [MPA+98] H. Moon, H. Park, S. Ahn, W. Kwon, **Performance Degradation of the IEEE 802.4 Token Bus Network in a Noisy Environment**, *Computer Communications*, Vol. 21, pp. 547-557, 1998.
- [MR93] M. Malhotra, A. Reibman, **Selecting and Implementing Phase Approximations for Semi-Markov Models**, *Stochastic Models*, Vol. 9, No. 4, pp. 473-506, 1993.

-
- [MS97] A. Moorsel, W. Sanders, **Transient Solution of Markov Models by Combining Adaptive and Standard Uniformization**, *IEEE Transactions on Reliability*, Vol. 46, No. 3, pp. 430-440, 1997.
- [MSG89] R. Muntz, E. Silva, A. Goyal, **Bounding Availability of Repairable Computer Systems**, *IEEE Transactions on Computers*, Vol. 38, No. 12, pp. 33-59, 1989.
- [MSM00] S. Mitra, N. Saxena, E. McCluskey, **Common-Mode Failures in Redundant VLSI Systems: A Survey**, *IEEE Transactions on Reliability*, Vol. 49, No. 3, pp. 285-295, 2000.
- [MT93] M. Malhotra, K. Trivedi, **A Methodology for Formal Expression of Hierarchy in Model Solution**, *Proceedings of the 5th International Workshop on Petri Nets and Performance Models, PNPM'93*, 1993.
- [MT94] M. Malhotra, K. Trivedi, **Power-Hierarchy of Dependability-Models Types**, *IEEE Transactions on Reliability*, Vol. 43, No. 3, pp. 493-502, 1994.
- [MT95] M. Malhotra, K. Tivedi, **Dependability Modeling Using Petri-Nets**, *IEEE Transactions on Reliability*, Vol. 44, No. 3, pp.428-440, 1995.
- [MT96] V. Mainkar, K. Trivedi, **Sufficient Conditions for Existence of a Fixed Point in Stochastic Reward Net-Based Iterative Models**, *IEEE Transactions on Software Engineering*, Vol. 22, pp. 640-653, 1996.
- [Mul93] S. Mullender, **Distributed Systems - Second Edition**, ACM Press, 1993.
- [Mur89] T. Murata, **Petri Nets: Properties, Analysis and Applications**, *Proceedings of the IEEE*, Vol. 77, No. 4, April, 1989.
- [Ne97] NAMUR, **Fieldbus for Safety-Related Uses**, Interessengemeinschaft Prozessleittechnik der chemischen und pharmazeutischen Industrie, 1997.
- [Neu03] P. Neumann, **Virtual Automation Network - problems to be solved**, *Proceedings of the IEEE Conference on Emerging Technologies and Factory Automation, ETFA'03*, 2003.
- [Nic98] M. Nicholson, **Selecting a Topology for Safety-Critical Real-Time Control Systems**, *PhD Thesis, University of York*, 1998.
- [NSN03] V. Nicola, P. Shahabuddin, M. Nakayama, **Techniques for Fast Simulation of Models of Highly Dependable Systems**, *IEEE Transactions on Reliability*, Vol. 30, No. 3, pp. 246-264, 2003.
- [NSS00] N. Navet, Y. Song, F. Simonot, **Worst-case Deadline Failure Probability in Real-Time Applications Distributed over CAN (Controller Area Network)**, *Journal of Systems Architecture*, Vol. 46, No. 7, pp. 607-618, 2000.
- [Odv03] **White Paper: Safety Networks, Increase Productivity, Reduce Work-Related Accidents and Save Money**, ODVA, 2003.
- [OMG98] **The Common Object Request Broker: Architecture and Specification**, OMG - Object Managment Group, 1998.
- [OPC98] **OPC Overview**, OPC Foundation, 1998.
- [OS94] W. Obal, W. Sanders, **An Environment for Importance Sampling Based on Stochastic Activity Networks**, *Proceedings of the 13th Symposium on reliable Distributed Systems*, 1994.
-

Bibliografia

- [PA01a] P. Portugal, A. Carvalho, **On Dependability Evaluation of Fieldbus Networks: a Permanent Fault Analysis**, *Proceedings of 27th Annual Conference of the IEEE Industrial Electronics Society, IECON'01*, 2001.
- [PA01b] P. Portugal, A. Carvalho, **On Dependability Evaluation of Fieldbus Networks: A Transient Fault Analysis**, *IFAC International Conference on Fieldbus Systems and their Applications, FeT'01*, 2001.
- [PA04a] P. Portugal, A. Carvalho, **A Framework for Dependability Evaluation of Fieldbus Networks**, *Proceedings of the 5th IEEE International Workshop on Factory Communication Systems, WFCS'04*, 2004.
- [PA04b] P. Portugal, A. Carvalho, **An Approach Based on Stochastic Petri Nets for Dependability Evaluation of Profibus-DP Networks**, *Proceedings of 30th Annual Conference of the IEEE Industrial Electronics Society, IECON'04*, 2004.
- [PA04c] P. Portugal, A. Carvalho, **Dependability Evaluation of Profibus-DP Networks by Using a Stochastic Petri Net Framework**, *Proceedings of the IEEE International Conference on Industrial Technology, ICIT'04*, 2004.
- [PA04d] P. Portugal, A. Carvalho, **A Stochastic Petri Net Framework for Dependability Evaluation of Fieldbus Networks – A Controller Area Network (CAN) Example**, *Proceedings of International Conference on Mechatronics and Robotics, MECROB'04*, 2004.
- [PAD+99] D. Powell, J. Arlat, L. Dukic, A. Bondavalli, P. Coppola, A. Fantechi, E. Jenn, C. Rabéjac, A. Wellings, **GUARDS: A Generic Upgradable Architecture for Real-Time Dependable Systems**, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 10, No.6, pp. 580-599, 1999.
- [PAP04] M. Pereira, P. Alves, M. Poli, **CORBA in Manufacturing - Technology Overview**, *The Industrial Information Technology Handbook*, CRC Press, 2004.
- [Pap98] I. Papazoglou, **Mathematical Foundations of Event Trees**, *Reliability Engineering and System Safety*, Vol. 61, pp. 169-183, 1998.
- [Pat98] R. Patzke, **Fieldbus Basics**, *Computer Standards & Interfaces*, Vol. 19, 1998.
- [PB99] Y. Purwantoro, S. Bennett, **Decomposition Technique for Integrated Dependability Evaluation of Hardware-Software Systems Using Stochastic Activity Networks**, *Proceedings of 25th EUROMICRO Conference - 2nd Workshop on Dependable Computing Systems*, 1999.
- [PBG99] M. Peller, J. Berwanger, R. Griebbach, **Bytefight Specification - Draft - Version 0.5**, BMW AG, 1999.
- [PCA+96] D. Powell, M. Cukier, J. Arlat, Y. Crouzet, **Estimation of Time-Dependent Coverage**, *LASS Report 96466*, 1996.
- [PD88] P. Plainevaux, J.-D. Decotignie, **Time Critical Communication Networks: Field Buses**, *IEEE Network*, Vol. 2, No. 3, pp. 55-63, 1988.
- [PD96] L. Pullum, J. Dugan, **Fault Tree Models for the Analysis of Complex Computer-Based Systems**, *Proceedings of the Annual Reliability and Maintainability Symposium*, 1996.
- [PG80] A. Pagès, M. Gondran, **Fiabilité des Systèmes**, Editions Eyrolles, 1980.

-
- [PG89] P. Glynn, **A GSMP Formalism for Discrete Event Systems**, *Proceedings of the IEEE*, Vol. 77, No. 1, pp. 14-23, 1989.
- [PHN00] S. Punnekkat, H. Hansson, C. Norström, **Response Time Analysis under Errors for CAN**, *Proceedings of the IEEE Real-Time Technology and Applications Symposium, RTAS'00*, 2000.
- [Pig00] R. Piggin, **A Fieldbus for Machine Safety**, *IEE Review*, July, 2000.
- [Pig04] R. Piggin, **Safety Networking - New technology enables greater flexibility**, *ISA InTech*, May, 2004.
- [Pim03] J. Pimentel, **Safety-Reliability of Distributed Embedded System Fault Tolerant Units**, *Proceedings of the 29th Annual Conference of the IEEE Industrial Electronics Society, IECON'03*, 2003.
- [Pim90] J. R. Pimentel, **Communication Networks for Manufacturing**, Prentice-Hall, 1990.
- [PMA+03] D. Powell, E. Martins J. Arlat, Y. Crouzet, **Estimators for Fault Tolerance Coverage Evaluation**, *Proceedings of the 23rd IEEE International Symposium on Fault-Tolerant Computing, FTCS-23*, 1993.
- [PMA+95] D. Powell, E. Martins, J. Arlat, Y. Crouzet, **Estimators for Fault Tolerance Coverage Estimation**, *IEEE Transactions on Computers*, Vo. 44, No. 2, pp. 261-274, 1995.
- [PMM+03] A. Perles, X. Molero, A. Marti, J. Serrano, **Analysis of New Proposals for Parallel Replication Based Discrete-Event Simulation**, *Proceedings of the Industrial Simulation Conference, ISC'03*, 2003.
- [Pom00] W. Pomales, **Software Fault-Tolerance: A Tutorial**, *Technical Report TM-2000-210616*, NASA, 2000.
- [Pow94] D. Powell, **Distributed Fault Tolerance - lessons Learnt from Delta-4**, *Hardware and Software Architectures for Fault-Tolerance*, pp. 199-217, Springer-Verlag, 1994.
- [Pow95] D. Powell, **Fault Assumptions and Assumption Coverage - Revised 1995**, *LAAS Report N° 90074*, 1995.
- [Pro02] **PROFIsafe - Profile for Safety Technology**, PNO, 2002.
- [Pro03] **PROFI-net Architecture Description and Specification**, PNO, 2003.
- [Pro04] <http://www.profibus.com>, 2004.
- [PS02] J. Pimentel, M. Salazar, **Dependability of Distributed Control System Fault Tolerant Units**, *Proceedings of the 28th Annual Conference of the IEEE Industrial Electronics Society, IECON'02*, 2002.
- [PSB+98] M. Pizza, L. Stringini, A. Bondavalli, F. Giandomenico, **Optimal Discrimination between Transient and Permanent Faults**, *Proceedings of the 3rd International Symposium on High-Assurance Systems Engineering, HASE'98*, 1998.
- [PST98] A. Puliafito, M. Scarpa, K. Trivedi, **Petri Nets with k Simultaneously Enabled Generally Distributed Timed Transitions**, *Performance Evaluation*, Vol. 32, pp. 1-34, 1998.
-

Bibliografia

- [PT00] K. Popstojanova, K. Trivedi, **Stochastic Modeling Formalisms for Dependability, Performance and Performability**, *Performance Evaluation: Origins and Directions*, pp. 403-422, Springer Verlag, 2000.
- [PTT97] A. Puliafito, M. Telek, K. Trivedi, **The Evolution of Stochastic Petri Nets**, *Proceedings of the World Congress on Systems and Simulation, WCSS'97*, 1997.
- [PVT00] L. Pinho, F. Vasques, E. Tovar, **Integrating Inaccessibility in Response Time Analysis of CAN Networks**, *Proceedings of the 3rd IEEE Workshop on Factory Communication Systems*, 2000.
- [Ram03] C. Ramebäck, **Process Automation Systems - History and Future**, *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA'03*, 2003.
- [RBA99] J. Rufino, P. Veríssimo, G. Arroz, **A Columbus' Egg Idea for the CAN Media Redundancy**, *Proceedings of the 29th International Symposium on Fault-Tolerant Computing, FTCS'99*, 1999.
- [RD01] C. Diedrich, F. Russo, **Function Block Applications in Control Systems Based on IEC 61804, ISA 2001**, 2001.
- [RLB+95] B. Rostamzadeh, H. Lönn, R. Snedsbol, J. Torin, **DECAPO: A Distributed Computer Architecture for Safety-Critical Control Applications**, *Proceedings of the IEEE International Symposium on Intelligent Vehicles*, 1995.
- [RNS93] U. Rembold, B. Nnaji, A. Storr, **Computer Integrated Manufacturing and Engineering**, Addison-Wesley, 1993.
- [Ros02] S. Ross, **Simulation, 3rd Edition**, Academic Press, 2002.
- [RPM+98] J. Rufino, N. Pedrosa, J. Monteiro, P. Veríssimo, G. Arroz, **Hardware Support for CAN Fault-Tolerant Communication**, *Proceedings of the 5th IEEE International Conference on Electronics, Circuits and Systems*, 1998.
- [RRD95] P. Raja, L. Ruiz, J.-D. Decotignie, **On the Necessary Real-Time Conditions for the Producer-Distributor-Consumer Model**, *Proceedings of the IEEE International Workshop on Factory Communication Systems, WFCS'95*, 1995.
- [Ruf95] **An Overview of the Controller Area Network**, Technical Report, 1995.
- [Ruf97a] J. Rufino, **Redundant CAN Architectures for Dependable Communication**, *CSTC Technical Report RT-97-07*, 1997.
- [Ruf97b] J. Rufino, **Dual-Media Redundancy Mechanisms for CAN**, *CSTC Technical Report RT-97-01*, 1997.
- [Rus01] J. Rushby, **Bus Architectures for Safety-Critical Embedded Systems**, *Proceedings of the 1st Workshop on Embedded Software*, 2001.
- [Rus03] J. Rushby, **A Comparison of Bus Architectures for Safety-Critical Embedded Systems**, *NASA Report CR-2003-212161*, 2003.
- [RV95] J. Rufino, P. Verissimo, **A Study on the Inaccessibility Characteristics of the Controller Area Network**, *Proceedings of the 2nd International CAN Conference*, 1995.

-
- [RVA99] J. Rufino, P. Veríssimo, G. Arroz, **Fundamental Issues in the Design of a CAN-based Fault-Tolerant Real-Time Communication Infrastructure for DEAR-COTS**, *CSTC Technical Report RT-99-02, Instituto Superior Técnico*, 1999.
- [RVV94] P. Rosa, F. Vasques, R. Vallete, **The Network Transparency Concept in Fieldbus Based Distributed Systems**, *Proceedings of the International Symposium in Industrial Electronics, ISIE'94*, 1994.
- [SAE00] **Single Wire Can Network for Vehicle Applications**, SAE, 2000.
- [Saf04] <http://www.safetybus.de>, 2004.
- [SBB04] G. Shabgahi, J. Bass, S. Bennet, **A Taxonomy for Software Voting Algorithms Used in Safety-Critical Systems**, *IEEE Transactions on Reliability*, Vol. 53, No. 3, pp. 319-328, 2004.
- [SBS97] A. Stefano, L. Bello, C. Santoro, **Synchronous Producer-Consumer Model Transactions for Real-Time Distributed Process Control**, *Proceedings of the IEEE International Workshop on Factory Communication Systems, WFCS'97*, 1997.
- [Sch04] K. Schwarz, **The Standard Message Specification for Industrial Automation Systems - ISO 9506 (MMS)**, *The Industrial Information Technology Handbook*, CRC Press, 2004.
- [Sch98] H. Schumny, **Fieldbuses in Measurement and Control**, *Computer Standards & Interfaces*, Vol. 19, pp. 295-304, 1998.
- [SDS99] **Smart Distributed System - Application Layer Protocol - Version 2.0**, Honeywell, 1999.
- [SGB00] F. Sheldon, S. Greiner, M. Benzigner, **Specification, Safety and Reliability Using Stochastic Petri Nets**, *Proceedings of the 10th International Workshop on Software Specification and Design, IWSS'00*, 2000.
- [Sho70] M. Shooman, **The Equivalence of Reliability Diagram and Fault-Tree Analysis**, *IEEE Transactions on Reliability*, Vol. R-19, pp. 74-75, 1970.
- [Sho94] M. Shooman, **A Study of Occurrence Rates of Electromagnetic Interference (EMI) to Aircraft with Focus on HIRF (External) High Intensity Radiated Fields**, *NASA CR-194 895*, 1994.
- [SHS+97] C. Scheidler, G. Heiner, R. Sasse, E. Fuchs, H. Kopetz and C. Temple, **The Time-Triggered Architecture (TTA)**, *Annual Conference and Exhibition European Multimedia, Embedded Systems and Electronic Commerce, EMMSEC'97*, 1997.
- [SIG04] **SIGUARD Machine Safety Products**, Siemens Industrial Control Products, 2004.
- [SJO04] T. Skeie, S. Johannessen, O. Holmeide, **Switched Ethernet in Automation Networking**, *The Industrial Information Technology Handbook*, CRC Press, 2004.
- [SK92] K. Shin, H. Kim, **Derivation and Application of Hard Deadlines for Real-Time Control Systems**, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 22, No. 6, pp. 1403-1413, 1992.
- [SL87] M. Shooman, A. Laemmel, **Simplification of Markov Models by State Merging**, *Proceedings of the Annual Reliability and Maintainability Symposium*, 1987.
-

Bibliografia

- [SM01] W. Sanders, J. Meyer, **Stochastic Activity Networks: Formal Definitions and Concepts**, *Lecture Notes in Computer Science*, No. 2090, pp. 315-343, 2001.
- [SM91a] W. Sanders, J. Meyer, **Reduced Base Model Construction Methods for Stochastic Activity Networks**, *IEEE Journal on Selected Areas in Communications*, Vol. 9, No. 1, pp. 25-36, 1991.
- [SM91b] W. Sanders, J. Meyer, **A Unified Approach for Specifying Measures of Performance, Dependability and Performability**, Dependable Computing for Critical Applications, Vol. 4 of Dependable Computing and Fault-Tolerant Systems, Springer-Verlag, 1991.
- [SM97] G. Schickuber, O. McCarthy, **Distributed fieldbus and control network systems**, *Computing & Control Engineering Journal*, February, 1997.
- [SMM89] W. Sanders, J. Meyer, K. Muralidhar, **Performability of a Token Bus Network Under Transient Fault Conditions**, *Proceedings of the 19th Annual International Symposium on Fault-Tolerance Computing*, FTCS-19, 1989.
- [SMN03] I. Snook, J. Marshall, R. Newman, **Physics of Failure As an Integrated Part of Design for Reliability**, *Proceedings of the Annual Reliability and Maintainability Symposium*, 2003.
- [SS94] R. Smith, D. Dietrich, **The Bathtub Curve: An Alternative Explanation**, *Proceedings of the Annual Reliability and Maintainability Symposium*, 1994.
- [Sto96] N. Storey, **Safety-Critical Computer Systems**, Addison-Welsey, 1996.
- [STR88] R. Smith, K. Trivedi, A. Ramesh, **Performability Analysis: Measures, an Algorithm and a Case Study**, *IEEE Transactions on Computers*, Vol. 37, No. 4, pp. 406-417, 1988.
- [SVV+00] V. Schiffer, K. Vandesteeg, D. Vasko, J. Lenner, **Introduction to DeviceNet Safety**, *IEEE Workshop on Factory Communications Systems*, WFCS'00, 2000.
- [TB94] K. Tindell, A. Burns, **Guaranteed Message Latencies for Distributed Safety-Critical Hard Real-Time Networks**, *Technical Report YCS 229*, University of York, 1994.
- [TBP96] M. Telek, A. Bobbio, A. Puliafito, **Steady State Solution of MRSPN with Mixed Preemption Policies**, *Proceedings of the IEEE International Conference on Parallel Distributed Systems*, 1996.
- [TBW95] K. Tindell, A. Burns, A. Wellings, **Calculating Controller Area Network (CAN) Message Response Times**, *Control Engineering Practice*, Vol. 3, No. 8, pp. 1163-1169, 1995.
- [Tem98] C. Temple, **Avoiding the Babbling-Idiot Failure in a Time-Triggered Communication System**, *Proceedings of the 28th International Symposium on Fault-Tolerant Computers Systems*, 1998.
- [TFM+93] K. Trivedi, G. Ciardo, M. Malhotra, R. Sahner, **Dependability and Performability Analysis**, *Performance Evaluation of Computer and Communication Systems*, *Lecture Notes in Computer Science*, pp. 587-612, Springer-Verlag, 1993.
- [TG83] K. Trivedi, R. Geist, **Decomposition in Reliability Analysis of Fault-Tolerant Systems**, *IEEE Transactions on Reliability*, Vol. R-32, No. 5, pp. 463-468, 1983.

-
- [Tho98] J. P. Tomesse, **A Review of the Fieldbuses**, *Annual Reviews in Control*, Vol. 22, pp. 35-45, 1998.
- [Tho99] J. P. Thomesse, **Fieldbuses and interoperability**, *Control Engineering Practice*, Vol. 7, pp. 81-94, 1999.
- [Til95] M. Tilgner, **An Approach to Formalize Structural Decomposition and Aggregation for Stochastic Reward Net Models**, *Proceedings of 6th International Workshop on Petri Nets and Performance Models, PNPM'95*, 1995.
- [TK03] K. Trivedi, V. Kulkarni, **FSPNs: Fluid Stochastic Petri Nets**, *Proceedings of the 14th International Conference on the Application and Theory of Petri Nets*, 1993.
- [TK99] T. Kletz, **Hazop and Hazan: Identifying and Assessing Process Industry Hazards**, Taylor & Francis Group, 1999.
- [TM93] K. Trivedi, M. Malhotra, **Reliability and Performability Techniques and Tools: A Survey**, *Proceedings of the 7th ITG/GI Conference on Measurement, Modelling and Evaluation of Computer and Communication Systems*, 1993.
- [TMG+94] L. Tomek, V. Mainkar, R. Geist, K. Trivedi, **Reliability Modeling of Life-Critical Real-Time Systems**, *Proceedings of the IEEE*, Vol. 82, No. 1, 1994.
- [Tov99] E. Tovar, **Supporting Real-Time Communications with Standard Factory-Floor Networks**, Tese de Doutorado, Faculdade de Engenharia da Universidade do Porto, 1999.
- [Tra99] E. Tran, **Multi-Bit Error Vulnerabilities in Controller Area Network Protocol**, *Technical Report*, Carnegie Mellon University, 1999.
- [TRF03] K. Trivedi, S. Ramani, R. Fricks, **Recent Advances in Modeling Response-Time Distributions in Real-Time Systems**, *Proceedings of the IEEE*, Vol. 91, No.7, pp. 1023-1036, 2003.
- [Triv02] K. Trivedi, **Probability and Statistics with Reliability, Queuing, and Computer Science Applications**, John Wiley & Sons, 2002.
- [TTP03] **Time-Triggered Protocol TTP/C High-Level Specification Document**, TTTech, 2003.
- [TV98a] E. Tovar, F. Vasques, **Guaranteeing Real-Time Message Deadlines in PROFIBUS Networks**, *Proceedings of the 10th Euromicro Workshop on Real-Time Systems*, 1998.
- [TV98b] E. Tovar, F. Vasques, **Setting Target Rotation Time in Profibus Based Real-Time Distributed Applications**, *Proceedings of the 15th IFAC Workshop on Distributed Computer Control Systems*, 1998.
- [TV99a] E. Tovar, F. Vasques, **Real-Time Communications Using Profibus Networks**, *IEEE Transactions on Industrial Electronics*, Vol. 46, No. 6, pp. 1241-1251, 1999.
- [TV99b] E. Tovar, F. Vasques, **Cycle Time Properties of the PROFIBUS Timed Token Protocol**, *Computer Communications*, Vol. 22, pp. 1206-1216, 1999.
- [TWB95] K. Tindell, A. Wellings, A. Burns, **Analysis of Hard-Real Time Communications**, *Real-Time Systems*, Vol. 9, pp. 147-171, 1995.
-

Bibliografia

- [Tys99] J. Tyszer, **Object-Oriented Computer Simulation of Discrete-Event Systems**, Kluwer Academic Publishers, 1999.
- [Vau98] J. Vaurio, **An Implicit Method for Incorporating Common-Cause Failures in System Analysis**, *IEEE Transactions on Reliability*, Vol. 47, No. 2, pp. 173-180, 1998.
- [Ver03] A. Verwer, **AS-Interface Safety at Work**, *Manchester Metropolitan University*, 2003.
- [Vit01] S. Vitturi, **On the use of Ethernet at low level of factory communication systems**, *Computer Standards & Interfaces*, Vol. 23, pp. 267-277, 2001.
- [Vit04] S. Vitturi, **On the Effects if the Acyclic Traffic on Profibus-DP Networks**, *Computers Standards & Interfaces*, Vol. 26, pp. 131-144, 2004.
- [VJ94] F. Vasques, G. Juanole, **Pre-Run-Time Schedulability Analysis in Fieldbus Networks**, *Proceedings of the International Conference on Industrial Electronics, Control and Instrumentation, IECON'94*, 1994.
- [VL89] P. Veríssimo, R. Lemos, **Confiança no Funcionamento: Proposta para uma Terminologia em Português**, INESC Relatório Técnico, 1989.
- [VRM97] P. Veríssimo, J. Rufino, L. Ming, **How Hard is Hard Real-Time Communication Fieldbuses**, *Proceedings of the 27th International Symposium on Fault-Tolerant Computing, FTCS'97*, 1997.
- [VV04] V. Volovoi, **Frameworks for System-Dependability Modeling**, *Submitted to IEEE Transactions on Reliability*, 2004.
- [VW04] C. Vetter, T. Werner, **Integration Between Production and Business Systems**, *The Industrial Information Technology Handbook*, CRC Press, 2004.
- [Wang98] J. Wang, **Timed Petri Nets: Theory and Application**, Kluwer Academic Publishers, 1998.
- [WCS+03] C. Wilwert, A. Charlois, Y. Song, A. Gilbert, F. Lion, **Impact of Fault-Tolerance Mechanisms on X-by-Wire System Dependability**, *Proceedings of SAE 2003 World Congress*, 2003.
- [Wei91] B. Wei, **A Unified Approach to Failure Mode, Effects and Criticality Analysis (FMECA)**, *Proceedings of Annual Reliability and Maintainability Symposium*, 1991.
- [WES04] **Intrinsic Safety Principles**, Westlook, 2004.
- [Whi02] J. Whitaker, **Electronic Systems Maintenance Handbook - 2nd Edition**, CRC Press, 2002.
- [WI00] N. Worsell, A. Ioannides, **Machinery Risk Assessment Validation Literature Review**, Health and Safety Laboratory, 2000.
- [Wil02] A. Willig, **Ring Stability of the PROFIBUS Token-Passing Protocol Over Error-Prone Links**, *IEEE Transactions on Industrial Electronics*, Vol. 48, No.5, pp. 1025-1033, 2002.
- [Wil04] A. Willig, **Wireless LAN Technology for the Factory Floor**, *The Industrial Information Technology Handbook*, CRC Press, 2004.
- [Wil96] A. Williamson, **Discrete Event Simulation in the Möbius Modeling Framework**, *Master Thesis, University of Illinois at Urbana-Champaign*, 1996.

-
- [Wil99a] A. Willig, **Analysis and Tuning of the PROFIBUS Token Passing Protocol for User over Error Prone Links**, *TKN Technical Report TKN-99-001*, 1999.
- [Wil99b] A. Willig, **Markov Modeling of PROFIBUS Ring Membership over Error Prone Links**, *TKN Technical Report TKN-99-004*, 1999.
- [WM95] H. Wang, N. Moayeri, **Finite-State Markov Channel - A Useful Model for Radio Communication Channels**, *IEEE Transactions on Vehicular Technology*, Vol. 44, No. 1, pp. 163-171, 1995.
- [Won95] K. Wong, **A New Framework for Part Failure-Rate Prediction Models**, *IEEE Transactions on Reliability*, Vol. 44, No. 1, pp. 139-146, 1995.
- [WSL+03] C. Wilwert, Y. Song, F. Lion, T. Clément, **Evaluating Quality of Service and Behavioral Reliability of Steer-by-Wire Systems**, *Proceedings of the 9th International Conference on Emerging Technologies and Factory Automation, ETFA'03*, 2003.
- [XBW98] **X-By-Wire: Safety Related Fault Tolerant Systems in Vehicles - Final Report**, X-By-Wire Consortium, 1998.
- [Xin03] L. Xing, **Reliability Analysis of Fault-Tolerant Systems with Common-Cause Failures**, *Proceedings of the International Conference on Dependable Systems and Networks, DSN'03*, 2003.
- [Yam00] S. Yamada, **Software Reliability Models and Their Applications: A Survey**, *International Seminar on Software Reliability of Man-Machine Systems - Theories, Methods and Information Systems Applications*, 2000.
- [ZC96] R. Zijal, G. Ciardo, **Discrete Deterministic and Stochastic Petri Nets**, *ICASE Technical Report 96-72*, Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, 1996.
- [ZCH97] R. Zijal, G. Ciardo, G. Hommel, **Discrete Deterministic and Stochastic Petri Nets**, *ITG/GI-Fachtagung: Messung, Modellierung und Bewertung von Rechen-und Kommunikationssystemen, MBB97*, pp. 103-105, 1997.
- [ZFG+00] A. Zimmermann, J. Freiheit, R. German, G. Hommel, **Petri Net Modelling and Performability Evaluation with TimeNET 3.0**, *Proceedings of the 11th International Conference on Modelling Techniques and Tools for Computer Performance Evaluation, TOOLS'00*, 2000.
- [ZFG+99] A. Zimmermann, J. Freiheit, R. German, G. Hommel, **TimeNET 3.0 Tool Description**, *Proceedings of the International Conference on Petri Nets and Performance Models, PNPM'99*, 1999.
- [ZFH01] A. Zimmermann, J. Freiheit, G. Hommel, **Discrete Time Stochastic Petri Nets for Modeling and Evaluation of Real-Time Systems**, *Proceedings of the International Workshop on Parallel and Distributed Real-Time Systems*, 2001.
- [Zim01] A. Zimmermann, **TimeNET 3.0 - User Manual**, Performance Evaluation Group, TU Berlin, 2001.
- [Zio02] E. Zio, **Common Cause Failures - An Analysis Methodology and Examples**, *Technical Report*, 2002.
-

Bibliografia

- [ZPD94] C. Ziegler, D. Powell, P. Deroches, **Dependability of On-Board Automotive Computer Systems**, *Proceedings of the IEEE Intelligent Vehicles Symposium*, 1994.