

**JOSÉ LUIZ PEREIRA DOS SANTOS**

**CÁLCULO DO TEMPO DE RESPOSTA  
DA REDE DE CAMPO PADRÃO WORLD FIP  
DO SISTEMA DE SUPERVISÃO E CONTROLE  
DE UMA UNIDADE HIDROGERADORA  
DE ENERGIA ELÉTRICA**

**FLORIANÓPOLIS**

**2003**

**UNIVERSIDADE FEDERAL DE SANTA CATARINA  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

**CÁLCULO DO TEMPO DE RESPOSTA DA REDE DE CAMPO  
PADRÃO WORLDFIP DO SISTEMA DE SUPERVISÃO E CONTROLE  
DE UMA UNIDADE HIDROGERADORA DE ENERGIA ELÉTRICA**

Dissertação submetida à  
Universidade Federal de Santa Catarina  
como parte dos requisitos para a  
obtenção do grau de Mestre em Engenharia Elétrica.

**José Luiz Pereira dos Santos**

Florianópolis, junho de 2003

**CÁLCULO DO TEMPO DE RESPOSTA DA REDE DE CAMPO  
PADRÃO WORLDFIP DO SISTEMA DE SUPERVISÃO E CONTROLE  
DE UMA UNIDADE HIDROGERADORA DE ENERGIA ELÉTRICA**

José Luiz Pereira dos Santos

'Esta Dissertação foi julgada adequada para a obtenção do título de Mestre em Engenharia Elétrica, Área de Concentração em *Controle, Automação e Informática Industrial*, e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Santa Catarina.'

---

Prof. Rômulo Silva de Oliveira, Dr.  
Orientador

---

Prof. Edson Roberto De Pieri, Dr.  
Coordenador do Programa de Pós-Graduação em Engenharia Elétrica

Banca Examinadora:

---

Prof. Rômulo Silva de Oliveira, Dr.  
Presidente

---

Prof. Marcelo Ricardo Stemmer, Dr.

---

Prof. Carlos Barros Montez, Dr.

---

Prof. Luiz Fernando Friedrich, Dr.

*A Alexandre (**in memoriam**), Morena, Lourdes,  
Miriam, Luciana, Joana e Amanda.*

## AGRADECIMENTOS

A minha dívida pela realização deste trabalho é tão grande, e envolve tantas pessoas queridas... Por isso devo começar pedindo desculpas àqueles que, por inexcusável falha de memória deste autor, não foram explicitamente citados. A eles a minha mais sincera gratidão.

A filosofia nos ensina que Deus manifesta sua presença também através das obras dos homens. Portanto acredito que através das ações humanas tomamos consciência de Sua bondade. Como a vida é a nossa maior dívida os meus primeiros pensamentos de gratidão são dirigidos aos meus pais, a quem devo, além da minha existência, o enorme sacrifício por eles feito pela minha educação.

À Itaipu Binacional, através de seus diretores, gerentes e colaboradores em geral, que me proporcionaram o suporte demandado por uma empreitada deste nível. Em particular agradeço ao Eng. Ricardo Pamplona pelo seu empenho pessoal e ao Eng. Alexandre Machado que desde a primeira hora deu seu crédito e apoio ao projeto. Registro aqui também os nomes dos Engs. Juan Sotuyo e Jorge Habib, professores da Unioeste e grandes incentivadores da qualificação pessoal e aperfeiçoamento tecnológico, que sempre dispensaram inestimáveis ações e palavras de apoio.

Ao meu orientador, Prof. Dr. Rômulo Silva de Oliveira, que soube conduzir com grande maestria a difícil tarefa me “enquadrar” no *modus operandi* de realização da pesquisa, organização e apresentação do tema do trabalho. E tudo isso com profundo profissionalismo, respeito, compreensão e amizade. Através do Prof. Rômulo eu gostaria de estender os meus agradecimentos a todos os demais professores do Departamento de Automação e Sistemas da UFSC.

Aos estagiários da Itaipu e alunos do Curso de Ciência da Computação da Unioeste Edney Matias da Silva e Filipe Correa Lima da Silva, que prestaram inestimável contribuição na elaboração do programa ANAFIP. Na realidade eles foram responsáveis pela implementação dos algoritmos e pelos testes da metodologia desenvolvida. A eles portanto pertence o crédito de uma parte importante do trabalho. E também ao Jefferson Ricardo Lopes, não tanto por disponibilizar o seu *notebook* em um momento crucial para o prosseguimento do trabalho de editoração do texto, mas pela boa vontade com que o fez.

Finalmente eu não poderia deixar de externar os meus agradecimentos à minha esposa, Miriam, pelo incentivo, carinho, dedicação, apoio em todos os momentos e paciência pelo sacrifício dos sábados, domingos e madrugadas utilizados nas intermináveis sessões de pesquisa, digitação e revisão do texto. E também pela impaciência de ver o trabalho concluído.

Resumo da Dissertação apresentada à UFSC como parte dos requisitos necessários para obtenção do grau de Mestre em Engenharia Elétrica.

## **CÁLCULO DO TEMPO DE RESPOSTA DA REDE DE CAMPO PADRÃO WORLDFIP DO SISTEMA DE SUPERVISÃO E CONTROLE DE UMA UNIDADE HIDROGERADORA DE ENERGIA ELÉTRICA**

**José Luiz Pereira dos Santos**

Junho / 2003

Orientador: Rômulo Silva de Oliveira

Área de Concentração: Controle, Automação e Informática Industrial

Palavras-chave: Controle de Processo Distribuído, Automação, Redes Industriais, Barramento de Campo, Sistemas de Tempo Real, Escalonamento.

Número de Páginas: 124

Este trabalho apresenta uma metodologia de cálculo do tempo de resposta de pior caso do tráfego de variáveis aperiódicas em uma rede de campo padrão WorldFIP aplicada no nível de processo de um sistema de supervisão e controle de uma unidade hidrogeradora de energia elétrica de grande porte. Um estudo de caso usando os dados levantados durante a fase preliminar do projeto do sistema de supervisão e controle local das duas novas unidades geradoras que estão sendo instaladas na Usina Hidrelétrica de Itaipu é apresentado como exemplo de aplicação da metodologia. São consideradas duas alternativas de montagem da tabela do árbitro do barramento da rede de campo: uma adota a abordagem de escalonamento taxa monotônica (*rate monotonic* - RM), baseada em prioridades estáticas; a outra emprega um algoritmo baseado em prioridades dinâmicas (*earliest deadline first* - EDF). Os resultados apurados permitem efetuar a comparação das vantagens e desvantagens da adoção de uma ou de outra alternativa de escalonamento do tráfego pela rede de campo de forma a garantir o atendimento das restrições associadas ao tráfego de mensagens de tempo real pelo suporte de comunicação.

Abstract of the Dissertation presented to UFSC as a partial fulfilment of the requirements for the degree of Master in Electrical Engineering.

**CALCULATION OF THE RESPONSE TIME OF A WORLDFIP  
FIELD BUS NETWORK IN THE SUPERVISION AND  
CONTROL SYSTEM OF A HYDROELECTRIC GENERATING UNIT**

**José Luiz Pereira dos Santos**

June / 2003

Advisor: Rômulo Silva de Oliveira

Area of Concentration: Control, Automation and Industrial Computing

Keywords: Distributed Process Control, Automation, Industrial Networks, Fieldbus, Real-Time Systems, Scheduling.

Number of Pages: 124

This dissertation presents a methodology for calculating the worst case response time of the aperiodic variables traffic on a WordFIP fieldbus network applied to the processing level in the local control and supervision system of a large hydroelectrical generator. A case study using the data gathered during the preliminary project phase of the control system design for the Itaipu Hydroelectric Plant's two new generator units is presented as an example of the application of the methodology. Two alternatives for assembling the network bus arbiter table are considered: one adopts the *rate monotonic* scheduling (RM) approach, based on static priorities; the other uses an algorithm based on the *earliest deadline first* (EDF) dynamic priority scheduling approach. The results verified permit comparing the advantages and disadvantages of adopting one or the other traffic scheduling alternative on the fieldbus network, in order to guarantee the compliance with the requirements and constraints associated with the real-time messages traffic over the process-level communication support.

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Sistemas Computacionais de Controle Distribuído . . . . .	1
1.2	Escopo do Trabalho . . . . .	3
1.3	Organização da Dissertação . . . . .	4
<b>2</b>	<b>Sistemas de Tempo Real</b>	<b>6</b>
2.1	Arquitetura de Sistema . . . . .	6
2.2	Requisitos Temporais . . . . .	8
2.2.1	Prazos Máximos de Atendimento das Tarefas . . . . .	8
2.2.2	Outras Características Relevantes dos Sistemas de Tempo Real . . . . .	9
2.2.3	Estratégias de Captura do Estado do Ambiente . . . . .	9
2.3	Modelagem das Tarefas . . . . .	11
2.3.1	Tarefas Periódicas . . . . .	12
2.3.2	Tarefas Esporádicas . . . . .	13
2.3.3	Tarefas Aperiódicas . . . . .	13
2.3.4	Um Modelo de Sistema . . . . .	14
2.4	Escalonamento de Tarefas de Tempo Real . . . . .	14
2.4.1	Abordagens de Escalonamento . . . . .	15
2.4.2	Inserção de Tempo Ocioso . . . . .	16
2.4.3	Critérios de Atribuição de Prioridades . . . . .	17
2.5	Análise das Abordagens de Escalonamento . . . . .	17
2.5.1	Testes de Escalonabilidade . . . . .	17
2.5.2	Executivo Cíclico . . . . .	19
2.5.3	Testes de Escalonabilidade para o Algoritmo RM . . . . .	21
2.5.4	Teste de Escalonabilidade para o Algoritmo Prazo Monotônico - (DM) . . . . .	23

2.5.5	Testes de Escalonabilidade para o Algoritmo EDF . . . . .	25
2.5.6	Aplicabilidade dos Algoritmos de Escalonamento . . . . .	26
2.6	Comunicação Tempo Real . . . . .	27
2.6.1	Modelo Hierárquico do Sistema de Comunicação . . . . .	27
2.6.2	Modelo de Tráfego de Tempo Real Adotado . . . . .	28
2.6.3	Atrasos na Transmissão de Mensagens . . . . .	29
2.6.4	Escalonamento de Mensagens de Tempo Real . . . . .	30
<b>3</b>	<b>A Rede de Campo Padrão WorldFIP</b>	<b>32</b>
3.1	Redes de Comunicação do Tipo Barramento de Campo . . . . .	32
3.1.1	Protocolos de Controle de Acesso ao Meio . . . . .	32
3.1.2	Arquitetura de Redes de Campo . . . . .	34
3.1.3	Unidade de Dados de Protocolo . . . . .	34
3.1.4	Redes de Campo Padronizadas . . . . .	35
3.2	A Rede WorldFIP . . . . .	36
3.2.1	Camada Física . . . . .	37
3.2.2	Camada de Enlace de Dados . . . . .	38
3.2.3	Camada de Aplicação . . . . .	39
3.2.4	Interfaces entre as Camadas . . . . .	41
3.2.5	Estrutura dos Quadros ID_DAT (Pergunta) . . . . .	44
3.2.6	Estrutura dos Quadros Quadros RP_DAT (Resposta) . . . . .	46
3.2.7	Durações das Transferências de <i>Buffers</i> . . . . .	46
3.2.8	Controle do Tráfego Periódico . . . . .	47
3.2.9	Montagem da TAB Usando a Abordagem Taxa Monotônica . . . . .	48
3.2.10	Montagem da TAB Usando a Abordagem <i>Earliest Deadline First</i> (EDF) . . . . .	49
3.2.11	Controle do Tráfego Aperiódico . . . . .	51
3.2.12	Eficiência da Rede . . . . .	55
<b>4</b>	<b>Uma Metodologia de Cálculo do Tempo de Resposta da Rede WorldFIP</b>	<b>56</b>
4.1	Principais Fatores de Influência para o Tempo de Resposta . . . . .	56
4.2	Modelos das Variáveis . . . . .	57
4.2.1	Variáveis Periódicas . . . . .	57
4.2.2	Variáveis Aperiódicas . . . . .	57

4.3	Teste de Escalonabilidade para o Algoritmo Executivo Cíclico . . . . .	58
4.4	Matriz de Escalonamento do Tráfego Periódico . . . . .	59
4.4.1	Matriz de Escalonamento Gerada pelo Algoritmo RM . . . . .	59
4.4.2	Matriz de Escalonamento Gerada pelo Algoritmo EDF . . . . .	59
4.5	Cálculo da Duração das Transações . . . . .	60
4.6	Determinação do Tempo de Resposta de Pior Caso do Tráfego Aperiódico . . . . .	61
4.6.1	Cálculo da Interferência Causada pelo Tráfego Periódico ( $I_p^k$ ) . . . . .	62
4.6.2	Cálculo da Demanda na Rede Devida ao Tráfego Aperiódico ( $W_a^k$ ) . . . . .	63
4.6.3	Algoritmo de Cálculo do Tempo de Resposta . . . . .	64
<b>5</b>	<b>Controle do Processo de Geração de Energia Elétrica</b>	<b>66</b>
5.1	O Processo Industrial de Geração de Energia . . . . .	66
5.1.1	Notas Históricas . . . . .	66
5.1.2	Sistemas Computacionais de Controle Aplicados a Usinas Geradoras e Subestações de Energia Elétrica no Brasil [4] . . . . .	68
5.2	Processo Físico (Objeto Controlado) . . . . .	68
5.2.1	Turbina Hidráulica . . . . .	69
5.2.2	Gerador . . . . .	70
5.2.3	Equipamentos Agregados e Sistemas Auxiliares . . . . .	71
5.2.4	Requisitos Ambientais do Processo Físico . . . . .	73
5.3	Sistema de Controle do Processo . . . . .	74
5.3.1	Funções Lógicas de Controle . . . . .	74
5.3.2	Níveis de Supervisão e Comando . . . . .	74
5.3.3	Suporte de Comunicação de Dados . . . . .	75
5.3.4	Tempos de Resposta do Sistema . . . . .	77
<b>6</b>	<b>Estudo de Caso - Cálculo do Tempo de Resposta da Rede de Campo do SDSC</b>	<b>78</b>
6.1	Esquema Lógico de Interconexão dos Controladores . . . . .	78
6.2	Sinais Físicos do Processo . . . . .	79
6.2.1	Sinais Discretos . . . . .	79
6.2.2	Sinais Analógicos . . . . .	80
6.3	Critérios de Formação das Variáveis e Quantificação dos Parâmetros da Rede . . . . .	82
6.3.1	Atributos das Variáveis Periódicas . . . . .	83

6.3.2	Atributos das Variáveis Aperiódicas . . . . .	83
6.3.3	Sinais Periódicos e Aperiódicos × Estações Produtoras . . . . .	84
6.4	Aplicação do Método de Cálculo . . . . .	84
6.4.1	O Programa ANAFIP . . . . .	84
6.4.2	Resultados Numéricos das Simulações . . . . .	85
6.4.3	Apresentação Gráfica dos Resultados . . . . .	86
<b>7</b>	<b>Conclusões</b>	<b>91</b>
<b>A</b>	<b>Tempos de Resposta da Rede WorldFIP</b>	<b>94</b>
A.1	Rede com Taxa de Transmissão de 1 Mb/s . . . . .	94
A.1.1	Resultados com VAR_AP = 12 bytes . . . . .	94
A.1.2	Resultados com VAR_AP = 24 bytes . . . . .	95
A.1.3	Resultados com VAR_AP = 36 bytes . . . . .	95
A.1.4	Resultados com VAR_AP = 48 bytes . . . . .	96
A.1.5	Resultados com VAR_AP = 64 bytes . . . . .	96
A.1.6	Resultados com VAR_AP = 96 bytes . . . . .	97
A.1.7	Resultados com VAR_AP = 126 bytes . . . . .	97
A.2	Rede com Taxa de Transmissão de 2,5 Mb/s . . . . .	98
A.2.1	Resultados com VAR_AP = 12 bytes . . . . .	98
A.2.2	Resultados com VAR_AP = 24 bytes . . . . .	98
A.2.3	Resultados com VAR_AP = 36 bytes . . . . .	99
A.2.4	Resultados com VAR_AP = 48 bytes . . . . .	99
A.2.5	Resultados com VAR_AP = 64 bytes . . . . .	100
A.2.6	Resultados com VAR_AP = 96 bytes . . . . .	100
A.2.7	Resultados com VAR_AP = 126 bytes . . . . .	101
<b>B</b>	<b>Códigos-Fonte dos Algoritmos Básicos</b>	<b>102</b>
B.1	Função mdc_mmc.c . . . . .	102
B.2	Função mrm.c . . . . .	106
B.3	Função medf.c . . . . .	110
B.4	Função jit.c . . . . .	115

<b>C ANAFIP - Interface com o Usuário</b>	<b>118</b>
C.1 Tela Principal . . . . .	118
C.2 Entrada de Parâmetros . . . . .	119
C.3 Edição de Grupos . . . . .	119
C.4 Edição de Estações . . . . .	120
C.5 Execução de Caso . . . . .	120

# Lista de Figuras

2.1	Interfaces do Sistema de Controle . . . . .	7
2.2	Valores das Tarefas em Função do Atendimento dos Prazos . . . . .	8
2.3	Tempo de Resposta de uma Tarefa . . . . .	12
2.4	Taxonomia de Abordagens de Escalonamento de Tarefas de Tempo Real . . . . .	15
2.5	Testes de Escalonabilidade . . . . .	18
2.6	Inserção de Tempo Ocioso . . . . .	21
2.7	Escalonamento de Tarefa Esporádica . . . . .	25
2.8	Modelo Hierárquico do Processo de Produção Industrial de Energia Elétrica . . . . .	28
3.1	Protocolos de Acesso ao Meio (MAC) Adotados em Redes de Campo . . . . .	34
3.2	Arquitetura de Protocolo de Redes de Campo . . . . .	34
3.3	Arquiteturas de Sistema - Centralizado / Distribuído . . . . .	36
3.4	Camadas de Comunicação do Protocolo WorldFIP . . . . .	36
3.5	Codificação dos Símbolos na Camada Física . . . . .	37
3.6	Codificação de um Quadro WorldFIP . . . . .	37
3.7	Transfêrencia de Buffer . . . . .	38
3.8	Formato do Quadro na Camada de Enlace . . . . .	40
3.9	Um Exemplo de PDU do Tipo Compacto . . . . .	41
3.10	Serviços Disponibilizados pelas Camadas de Aplicação e de Enlace . . . . .	41
3.11	Operações de Leitura e de Escrita Locais . . . . .	42
3.12	Transferência de Buffers . . . . .	43
3.13	Leitura de Variável Remota - Transferência Aperiódica . . . . .	45
3.14	Estrutura do Quadro ID_DAT . . . . .	45
3.15	Estrutura do Quadro RP_DAT . . . . .	46
3.16	Estrutura do Quadro RP_RQ . . . . .	46

3.17	Duração da Transferência de <i>Buffers</i> . . . . .	47
3.18	Grade de Tempo Referente ao Tráfego de Variáveis Periódicas . . . . .	48
3.19	Tráfego de Variáveis Periódicas - Grade de Tempo da Abordagem RM . . . . .	49
3.20	Tráfego de Variáveis Periódicas - Abordagem RM - Sistema não Escalonável . . . . .	50
3.21	Tráfego de Variáveis Periódicas - Grade de Tempo da Abordagem EDF . . . . .	51
3.22	Janelas de Tempo do Ciclo Elementar . . . . .	51
3.23	Filas de Requisições Aperiódicas no Árbitro do Barramento . . . . .	52
3.24	Tempo de Atendimento de Requisições Aperiódicas . . . . .	54
5.1	Controle Convencional - Arquitetura Centralizada . . . . .	67
5.2	Estados e Transições Operacionais da Unidade Geradora . . . . .	69
5.3	Arquitetura de <i>Hardware</i> do SDSC . . . . .	76
6.1	Arranjo Lógico dos Controladores de Entrada e Saída . . . . .	79
6.2	Tempos de Resposta - RM $\times$ EDF - <b>Estação A000 (1,0 Mb/s - <math>T_1 = 10\text{ms}</math>)</b> . . . . .	86
6.3	Tempos de Resposta - RM $\times$ EDF - <b>Estação A007 (1,0 Mb/s - <math>T_1 = 10\text{ms}</math>)</b> . . . . .	87
6.4	Tempos de Resposta - RM $\times$ EDF - <b>Estação A000 (1,0 Mb/s - <math>T_1 = 20\text{ms}</math>)</b> . . . . .	87
6.5	Tempos de Resposta - RM $\times$ EDF - <b>Estação A007 (1,0 Mb/s - <math>T_1 = 20\text{ms}</math>)</b> . . . . .	88
6.6	Tempos de Resposta - RM $\times$ EDF - <b>Estação A000 (2,5 Mb/s - <math>T_1 = 5\text{ms}</math>)</b> . . . . .	88
6.7	Tempos de Resposta - RM $\times$ EDF - <b>Estação A007 (2,5 Mb/s - <math>T_1 = 5\text{ms}</math>)</b> . . . . .	89
6.8	Tempos de Resposta - RM $\times$ EDF - <b>Estação A000 (2,5 Mb/s - <math>T_1 = 10\text{ms}</math>)</b> . . . . .	89
6.9	Tempos de Resposta - RM $\times$ EDF - <b>Estação A007 (2,5 Mb/s - <math>T_1 = 10\text{ms}</math>)</b> . . . . .	89
6.10	Tempos de Resposta - RM $\times$ EDF - <b>Estação A000 (2,5 Mb/s - <math>T_1 = 20\text{ms}</math>)</b> . . . . .	90
6.11	Tempos de Resposta - RM $\times$ EDF - <b>Estação A007 (2,5 Mb/s - <math>T_1 = 20\text{ms}</math>)</b> . . . . .	90
C.1	ANAFIP - Tela principal . . . . .	118
C.2	ANAFIP - Entrada de Parâmetros . . . . .	119
C.3	ANAFIP - Edição de Grupos . . . . .	119
C.4	ANAFIP - Edição de Estações . . . . .	120
C.5	ANAFIP - Execução de um Caso . . . . .	120

# Lista de Tabelas

2.1	Conjunto de Tarefas Periódicas . . . . .	20
3.1	Parâmetros das Variáveis Periódicas de um Sistema Distribuído . . . . .	49
5.1	Atrasos Máximos Admissíveis para os Sinais Gerados pelo Processo . . . . .	77
6.1	Quantificação dos Sinais Discretos . . . . .	81
6.2	Quantificação dos Sinais Analógicos . . . . .	82
6.3	Valores dos Períodos - Grupos de Sinais Periódicos . . . . .	83
6.4	Grupos de Sinais Periódicos e Aperiódicos $\times$ Estações Produtoras . . . . .	84
A.1	Tempos de Resposta - RM $\times$ EDF ( $tx\_rate = 1$ Mb/s - VAR <sub>AP</sub> = 12 bytes) . . . . .	94
A.2	Tempos de Resposta - RM $\times$ EDF ( $tx\_rate = 1$ Mb/s - VAR <sub>AP</sub> = 24 bytes) . . . . .	95
A.3	Tempos de Resposta - RM $\times$ EDF ( $tx\_rate = 1$ Mb/s - VAR <sub>AP</sub> = 36 bytes) . . . . .	95
A.4	Tempos de Resposta - RM $\times$ EDF ( $tx\_rate = 1$ Mb/s - VAR <sub>AP</sub> = 48 bytes) . . . . .	96
A.5	Tempos de Resposta - RM $\times$ EDF ( $tx\_rate = 1$ Mb/s - VAR <sub>AP</sub> = 64 bytes) . . . . .	96
A.6	Tempos de Resposta - RM $\times$ EDF ( $tx\_rate = 1$ Mb/s - VAR <sub>AP</sub> = 96 bytes) . . . . .	97
A.7	Tempos de Resposta - RM $\times$ EDF ( $tx\_rate = 1$ Mb/s - VAR <sub>AP</sub> = 126 bytes) . . . . .	97
A.8	Tempos de Resposta - RM $\times$ EDF ( $tx\_rate = 2,5$ Mb/s - VAR <sub>AP</sub> = 12 bytes) . . . . .	98
A.9	Tempos de Resposta - RM $\times$ EDF ( $tx\_rate = 2,5$ Mb/s - VAR <sub>AP</sub> = 24 bytes) . . . . .	98
A.10	Tempos de Resposta - RM $\times$ EDF ( $tx\_rate = 2,5$ Mb/s - VAR <sub>AP</sub> = 36 bytes) . . . . .	99
A.11	Tempos de Resposta - RM $\times$ EDF ( $tx\_rate = 2,5$ Mb/s - VAR <sub>AP</sub> = 48 bytes) . . . . .	99
A.12	Tempos de Resposta - RM $\times$ EDF ( $tx\_rate = 2,5$ Mb/s - VAR <sub>AP</sub> = 64 bytes) . . . . .	100
A.13	Tempos de Resposta - RM $\times$ EDF ( $tx\_rate = 2,5$ Mb/s - VAR <sub>AP</sub> = 96 bytes) . . . . .	100
A.14	Tempos de Resposta - RM $\times$ EDF ( $tx\_rate = 2,5$ Mb/s - VAR <sub>AP</sub> = 126 bytes) . . . . .	101

# Capítulo 1

## Introdução

Um processo industrial típico é formado por subsistemas de equipamentos e dispositivos localizados em uma área que pode ser relativamente extensa. Apesar da separação física estes subsistemas devem interagir e cooperar sob supervisão e controle de um sistema que tenha capacidade de processar, transferir e armazenar as informações relevantes para o correto funcionamento do processo nas condições especificadas.

### 1.1 Sistemas Computacionais de Controle Distribuído

O conceito de **sistema distribuído** adotado nesta Dissertação é definido por Pimentel [32] como uma coleção de “diversos dispositivos inteligentes autônomos que cooperam entre si para atingir um objetivo comum. Os dispositivos inteligentes são capazes de suportar processos que coordenam suas atividades e realizam o intercâmbio de informações por meio de uma rede de comunicação”. A aplicação deste conceito evidencia a essencialidade do suporte de comunicação em relação aos **sistemas computacionais de controle distribuído (SCCD)** (*distributed computer-controlled systems - DCCS*).

Um SCCD geralmente implementa laços de controle que são executados em diferentes nós computacionais. Estes sistemas tipicamente baseiam-se em conjuntos de tarefas periódicas e esporádicas com prazos de execução pré-definidos e possivelmente com relações de precedência. O desempenho global do sistema depende do tempo de processamento das variáveis de controle localmente em cada nó e dos atrasos na transmissão das informações através dos meios de comunicação. Requisitos de estabilidade e segurança podem ser direta ou indiretamente mapeados como restrições temporais no

nível de cada estação ou no nível do suporte de comunicação. Estes atributos elegem os processos industriais como fortes candidatos à aplicação de sistemas computacionais de controle distribuído.

O suporte de comunicação de um SCCD é implementado como um conjunto de redes locais formando uma estrutura hierárquica com um determinado número de níveis funcionais caracterizados por atributos específicos e por restrições temporais que refletem a natureza de suas atividades relevantes.

As redes de comunicação empregadas nos níveis inferiores dos processos industriais interligam dispositivos tais como sensores, atuadores e controladores. Estas redes são baseadas em protocolos especializados denominados **barramentos de campo** (*fieldbus*) ou **redes de campo**. As redes de campo devem suportar o tráfego de dados entre estes elementos dentro de janelas de tempo limitadas cooperando com os demais componentes do sistema de forma a garantir a correção lógica dos processamentos, a estabilidade operacional do processo e, em certas aplicações, a integridade do ambiente.

A especificação das restrições temporais de forma explícita, a fim de atender às demandas do ambiente, é um dos requisitos que permitem incluir as redes de campo no conjunto dos **sistemas de tempo real** considerando-se a taxonomia encontrada na literatura [12]. Nestes sistemas as falhas no atendimento das restrições temporais são, em determinados casos, consideradas críticas no que diz respeito às suas conseqüências [39].

Um problema fundamental encontrado na análise e no projeto de sistemas distribuídos de tempo real é a alocação e o escalonamento das tarefas nos recursos computacionais disponíveis e o gerenciamento do tráfego de dados nos suportes de comunicação. Existe uma dificuldade intrínseca em compatibilizar dois objetivos fundamentais [8]: garantir que os resultados sejam disponibilizados no momento especificado e dotar o sistema de flexibilidade para adaptar-se a um ambiente dinâmico e, assim, aumentar a sua utilidade.

A teoria de escalonamento tempo real pode ser empregada como uma ferramenta de suporte para definição do modelo matemático do sistema de forma que suas propriedades temporais sejam determinadas explicitamente através de equações e algoritmos apropriados. Esta ferramenta considera as políticas usadas para alocar os diversos recursos disputados, dispensando ênfase especial aos processadores e aos meios de comunicação. Esta teoria é relativamente recente, tendo sido usada com maior freqüência no mundo acadêmico e em aplicações estratégicas tais como sistemas de defesa e controle de tráfego aéreo. Recentemente os benefícios decorrentes de sua utilização tem-se estendido para

o campo das aplicações industriais e comerciais uma vez que, adequadamente aplicada, ela poderá trazer enormes benefícios para as aplicações de sistemas de tempo real de maneira mais generalizada.

Apesar das diferenças existentes entre o escalonamento de tarefas em um processador e o escalonamento de mensagens em uma rede de comunicação, há analogias entre estas duas atividades que justificam a aplicação, no escalonamento de mensagens, dos princípios desenvolvidos para o escalonamento de tarefas em um processador. Cardeira e Marmeri [10] destacam algumas dessas analogias:

1. Escalonar tarefas em um processador significa serializar a sua execução. Analogamente o escalonamento de mensagens em uma rede pode ser interpretado como a serialização da transmissão das mensagens através do meio físico.
2. Duas tarefas não podem ser executadas em um mesmo processador ao mesmo tempo, assim como duas mensagens não poderão ser transmitidas através de um suporte físico de uma rede, operando em banda base, no mesmo instante. Neste caso ocorrerá o evento denominado **colisão de dados**, que resultará na perda do conteúdo das mensagens se não forem adotados mecanismos de resolução deste tipo de contenção.
3. Uma tarefa tem um tempo de computação que depende, basicamente, do número de instruções. Similarmente, o tempo de transmissão da mensagem depende do número de bits transmitidos.

## 1.2 Escopo do Trabalho

O emprego do protocolo WorldFIP [43] em plantas de geração de energia elétrica no Brasil motivou a pesquisa de um modelo matemático que possa ser utilizado como ferramenta de verificação do atendimento dos requisitos temporais associados ao tráfego de mensagens de tempo real em sistemas que adotem este padrão de rede de campo no nível de controle do processo.

O objetivo deste trabalho é apresentar uma metodologia de cálculo do tempo de resposta da rede WorldFIP adotada como suporte de comunicação do nível de processo (nível 1) de um sistema de supervisão e controle de uma planta de geração de energia elétrica.

A aplicação da metodologia foi consubstanciada em um estudo de caso utilizando os dados levantados durante a fase preliminar do projeto do sistema de controle das duas novas unidades geradoras que estão sendo instaladas na Usina Hidrelétrica de Itaipu. As simulações foram feitas através da utilização do programa ANAFIP, que foi desenvolvido para servir como ferramenta de

suporte à análise do comportamento temporal da rede WorldFIP em função dos valores atribuídos aos parâmetros de controle identificados no modelo adotado para o sistema de comunicação.

São apresentadas duas alternativas de montagem da tabela utilizada pelo árbitro de barramento da rede de campo para controlar o tráfego de variáveis periódicas: uma adota a abordagem de escalonamento taxa monotônica (*rate monotonic* - RM), baseada em prioridades fixas; a outra emprega um algoritmo de escalonamento baseado na atribuição dinâmica de prioridades (*earliest deadline first* - EDF). As informações contidas nos resultados apurados podem ser utilizadas para subsidiar a avaliação da influência de cada parâmetro de controle no comportamento da rede de campo. Servem também como base para a comparação das vantagens e desvantagens da adoção de cada um dos algoritmos de escalonamento como ferramenta de montagem da tabela do árbitro do barramento.

### 1.3 Organização da Dissertação

Os assuntos abordados na Dissertação estão distribuídos em sete capítulos, iniciando pela Introdução (Capítulo 1).

O Capítulo 2 apresenta a conceituação básica dos sistemas de tempo real com ênfase nas definições, terminologia e modelos que podem ser adotados como referência para a análise e projeto do suporte de comunicação empregado nos níveis inferiores da estrutura hierárquica dos sistemas computacionais de controle distribuído aplicados ao processo de geração de energia elétrica. Com base nos resultados da pesquisa dos trabalhos relevantes publicados sobre o assunto foram selecionados os tópicos utilizados como base para o modelamento da rede de campo de forma que sejam atendidos os requisitos temporais inerentes à aplicação pretendida. Desta forma foi possível a adoção de uma metodologia de verificação do atendimento das restrições temporais associadas ao intercâmbio de dados entre os componentes do sistema.

O Capítulo 3 contém uma breve descrição das características e funcionalidades da rede industrial padrão WorldFIP. Tendo como fontes de informação a documentação técnica disponibilizada pelo desenvolvedor do protocolo e artigos publicados sobre o assunto são apresentados detalhes do mecanismo de controle de acesso ao meio adotado pela camada de enlace da rede e algumas equações que modelam o tratamento das variáveis periódicas e aperiódicas pelas camadas do protocolo.

A partir da conceituação dos sistemas de tempo real e dos atributos da rede WorldFIP tratados nos capítulos anteriores, o Capítulo 4 apresenta uma metodologia de cálculo que serve como ferra-

menta para verificação da escalonabilidade do conjunto de variáveis periódicas e para determinação do limite superior do tempo de resposta de pior caso do tráfego de variáveis aperiódicas em uma rede WorldFIP. São identificados os principais fatores de influência no tempo de resposta deste protocolo de comunicação, apresentadas as equações aplicáveis e desenvolvido um algoritmo que quantifica as duas parcelas identificadas como **interferência causada pelo tráfego periódico** e **demanda na rede devida ao tráfego aperiódico**, cuja adição tem como resultado o atraso no atendimento das requisições de transmissão dos dados através da rede de campo.

O contexto de aplicação do método de cálculo desenvolvido no Capítulo 4 é o sistema digital de supervisão e controle associado a cada uma das unidades geradoras de 700000 kW que estão sendo instaladas na Usina Hidrelétrica de Itaipu. Dentro desta perspectiva o Capítulo 5 descreve, de maneira sucinta, os principais equipamentos e sistemas auxiliares que compõem uma planta hidrogeradora de energia elétrica e também apresenta alguns requisitos especificados para o sistema de supervisão e controle do processo a ela associado [23]. A seguir é apresentada a arquitetura de *hardware* do SDSC onde reside a rede cujos atributos foram considerados para a aplicação da metodologia de cálculo do tempo de resposta tratada nesta Dissertação.

No Capítulo 6 encontra-se a descrição do estudo de caso onde é aplicado o método de cálculo desenvolvido no Capítulo 4 empregando os dados levantados durante a fase preliminar do projeto do SDSC. Os sinais adquiridos através da interface de instrumentação são apresentados em tabelas que contêm os dados classificados em função dos atributos adotados como base para o critério de formação das estruturas de dados que trafegam pela rede. O algoritmo de cálculo foi implementado em um programa denominado ANAFIP, contendo módulos de entrada, classificação, ordenação e estruturação dos dados para então aplicar os procedimentos de cálculo que verificam a escalonabilidade das variáveis periódicas e produzem os tempos de resposta das variáveis aperiódicas produzidas pelas estações subscritoras da rede WorldFIP.

O Capítulo 7 apresenta as conclusões do trabalho.

## Capítulo 2

# Sistemas de Tempo Real

Os **sistemas computacionais de tempo real** distinguem-se por sua capacidade de produzir resultados cuja correção depende não apenas dos valores lógicos mas também dos instantes físicos nos quais estes resultados são produzidos [25]. O termo **sistema de tempo real** usualmente refere-se ao sistema completo, incluindo computadores, sistemas operacionais, programas de aplicação, dispositivos de entrada e saída (E/S), suportes de comunicação e interfaces com o ambiente.

A literatura especializada classifica os sistemas de tempo real a partir de diversas perspectivas. O interesse deste trabalho concentra-se na classificação que considera as possíveis conseqüências das falhas no atendimento, pelo sistema de controle, dos prazos estabelecidos para a produção dos resultados esperados. Os sistemas de tempo real **críticos** (*hard real-time systems*), a exemplo dos que geralmente são empregados para controle de processos industriais, têm tarefas com prazos de atendimento que, se não obedecidos, os resultados poderão causar perdas materiais ou econômicas significativas. Em contrapartida os sistemas de tempo real **brandos** (*soft real-time systems*) são aqueles cujo ocasional não atendimento do prazo de qualquer uma de suas tarefas, apesar de indesejável, não é catastrófico.

### 2.1 Arquitetura de Sistema

No contexto desta dissertação, o termo **sistema distribuído** representa um conjunto de **estações** ou **nós computacionais** interligados por uma rede de comunicação de dados. Cada nó consiste de um sistema computacional autônomo, com todos os recursos de suporte de *hardware* e *software*. Os nós têm comportamento dinâmico independente e processam as tarefas (ver seção 2.3) que cooperam para realizar os objetivos funcionais especificados para o sistema.

Um sistema de controle em tempo real é formado por diversos componentes lógicos. Kopetz [25] classifica estes componentes em três subsistemas funcionais: o processo físico (**objeto controlado**), um sistema computacional que exerce ações sobre o objeto controlado (**sistema de controle**) e o **operador humano** (Fig. 2.1). O objeto controlado e o operador humano podem ainda ser agrupados em um conjunto denominado **ambiente**. O ambiente impõe as **restrições temporais** e constitui-se na principal fonte dos requisitos que deverão ser atendidos pelo sistema de controle.

O ambiente destaca-se, adicionalmente, por dois componentes que definem as áreas de interação entre os subsistemas funcionais: 1) o processo físico comunica-se com o sistema de controle através da **interface de instrumentação**, representada pelos sensores e atuadores; 2) os diálogos entre o operador humano e o sistema de controle para intervenção no comportamento do objeto controlado com finalidades operacionais e os serviços disponibilizados pelo sistema de controle para os usuários finais são realizados através da **interface homem-máquina**. Os dispositivos de entrada mais comuns desta interface são o teclado e o dispositivo apontador (*mouse*); os principais componentes de saída são o monitor de vídeo e a impressora.

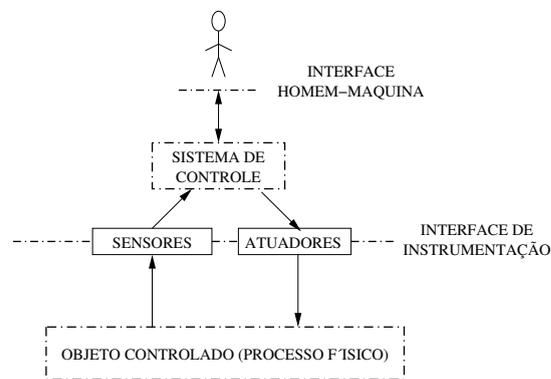


Figura 2.1: Interfaces do Sistema de Controle

Os sistemas de tempo real geralmente processam informações estruturadas sob a forma de fluxos contínuos de dados (*streams*) gerados por eventos múltiplos e independentes, e produzem os resultados que são aplicados ao ambiente através de suas interfaces. Estes eventos têm taxas de chegada que, muitas vezes, são imprevisíveis. Ainda assim a resposta do sistema de controle deve atender aos limites de tempo especificados na definição de requisitos de *software*. Como consequência a carga computacional demandada pelo tratamento da massa de dados produzidos pelo ambiente pode variar com o tempo de forma significativa, o que requer a adoção de uma disciplina de ordenação da utilização dos recursos disponíveis, que geralmente são limitados, de modo a garantir a correta resposta do sistema de controle para todas as condições operacionais impostas pelo ambiente.

## 2.2 Requisitos Temporais

### 2.2.1 Prazos Máximos de Atendimento das Tarefas

Os termos **tarefa** (*task*) e **processo** (*process*) são empregados, na literatura relativa a sistemas de tempo real, com significados que dão margem a diferentes interpretações. A conceituação e o modelo matemático adotados para o termo **tarefa**, neste trabalho, estão definidos na seção 2.3.

Uma característica comum aos sistemas de tempo real é a inclusão das relações temporais em suas especificações de requisitos sob a forma de prazos de execução. O atendimento destes requisitos resulta na principal restrição associada a certos sistemas que é a exigência de processamento das tarefas dentro de janelas de tempo limitadas, cuja duração específica é função da aplicação. Define-se como **prazo** (*deadline*) o último instante aceitável para que a tarefa produza e disponibilize o resultado do processamento de sua instância corrente. Esta conceituação implica na atribuição de **valor** às possíveis conseqüências do não atendimento dos prazos.

Se o resultado ainda tiver alguma utilidade quando sua produção ocorrer depois de ser ultrapassado o prazo, este será classificado como **brando**. Se, uma vez ultrapassado o prazo, o resultado não tiver valor então o prazo será **firme**. Se o não atendimento do prazo resultar em perdas econômicas, materiais ou mesmo humanas, ou seja, se os resultados forem **catastróficos**, o prazo será classificado como **crítico**. Com base nesta classificação um sistema de tempo real crítico será aquele que contenha pelo menos uma tarefa com prazo de atendimento crítico [25].

Na Fig. 2.2 as curvas tempo  $\times$  valor representam o conceito de classificação das tarefas computacionais em função do atendimento dos prazos. Nestas curvas, o intervalo de tempo despendido entre o início da execução de uma tarefa e o vencimento do prazo especificado para sua conclusão é mapeado contra o valor que o resultado representa para o sistema [6].

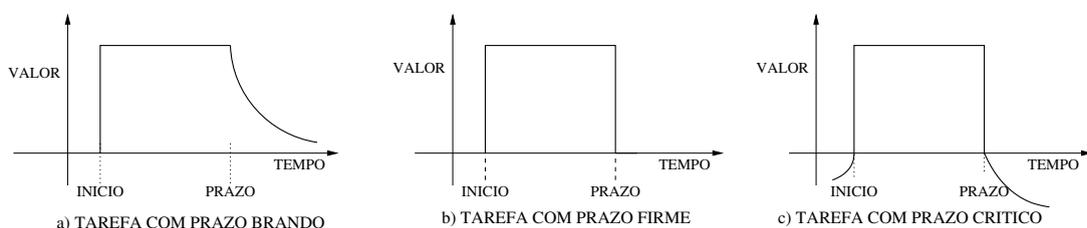


Figura 2.2: Valores das Tarefas em Função do Atendimento dos Prazos

Na maioria dos sistemas de tempo real de grande porte nem todos os eventos são tratados por tarefas com prazos críticos. Alguns não têm prazos de conclusão explicitamente definidos e outros

têm prazos brandos (*soft deadlines*). Se no sistema não existir nenhuma tarefa com prazo crítico então o sistema será brando [25].

### 2.2.2 Outras Características Relevantes dos Sistemas de Tempo Real

Os sistemas de tempo real apresentam ainda características adicionais que os distinguem de outros sistemas de controle baseados em *software* computacional [15]:

**Sistemas embutidos:** Em muitos casos o sistema de tempo real é parte de um sistema mais abrangente, composto de *hardware* e *software* integrados que podem ainda incluir, entre seus componentes, um subsistema mecânico e mesmo uma interface homem-máquina. Os demais componentes e o operador humano, se houver, constituem-se no ambiente deste sistema, que é então denominado **sistema embutido** (*embedded system*) - o sistema funcional completo é conhecido como **produto inteligente** [25].

**Sistemas de Automação de Plantas Industriais:** A automação de plantas industriais foi o ponto histórico de partida da aplicação de computadores para controle de processos. A evolução da tecnologia de instrumentação industrial e a disponibilidade de controladores programáveis possibilitou a instalação de suportes adequados para o comando e a supervisão de processos em salas de comando centralizadas, com o aumento da quantidade e principalmente da qualidade das informações relativas ao processo controlado, em que pese a eventual redução do número de operadores no nível de chão-de-fábrica.

**Sistemas reativos:** Os sistemas de tempo real também classificam-se, de forma geral, como **sistemas reativos**, uma vez que as ações de controle devem responder pronta e continuamente a estímulos do ambiente [12].

### 2.2.3 Estratégias de Captura do Estado do Ambiente

A interação funcional entre os nós computacionais de um sistema de tempo real e o ambiente depende do conhecimento, pelo sistema de controle, do estado corrente das variáveis do processo controlado e das demandas por serviços requisitados pelo operador. Para isso, as estruturas de dados referentes aos objetos de imagem das entidades do ambiente deverão ser mantidas atualizadas sob a forma de um banco de dados cuja validade tem duração finita [25].

Liu [28] define a **idade de um objeto de imagem** como o tempo transcorrido a partir da última atualização da variável, ou seja, desde o instante em que o seu valor corresponda ao valor instantâneo

da entidade do mundo real que ela representa. Esta característica conduz ao conceito de **consistência temporal** de dados, que define como **temporalmente consistente** um conjunto de objetos de imagem cuja idade máxima seja inferior a um determinado limite.

Os diversos nós de um sistema distribuído utilizam cópias dos objetos de imagem do processo. A atualização dos valores das variáveis correspondentes a estes objetos deverá ser atômica a fim de garantir a consistência temporal dos dados, de forma global. Este requisito é a base da definição do conceito de **consistência espacial** de dados.

Com relação à consistência temporal dos dados duas abordagens distintas são empregadas para ativação das tarefas computacionais: **disparo por evento** (*event triggered approach*) e **disparo por tempo** (*time triggered approach*).

Define-se como **evento** um estímulo que pode ser gerado pelo ambiente (evento externo) ou pelo próprio sistema de controle (evento interno) que represente ou tenha como consequência uma mudança de estado significativa no ambiente.

Na abordagem de disparo por evento o sistema de controle é notificado de qualquer alteração significativa do estado do ambiente. Recebendo tal notificação, o sistema dispara as ações apropriadas. Nas situações de ocorrência de eventos simultâneos deverá ser estabelecido um critério de ordenação da execução das ações de controle já que o nível de contenção para utilização dos recursos do sistema poderá ser tão elevado que o risco de algumas tarefas deixarem de atender seus prazos finais passa a ser significativo.

Na abordagem de disparo por tempo as ações são ativadas em instantes pré-definidos, geralmente de forma periódica, para efetuar a varredura das variáveis de estado do ambiente a uma taxa definida em função da resposta dinâmica do ambiente. Mesmo quando não há mudanças significativas de estado, as ações do sistema de controle são continuamente disparadas de forma cíclica. Porém, como os instantes de chegada são pré-definidos, é possível administrar o nível de contenção entre as ações através do controle adequado das fases relativas das tarefas nestes instantes.

A abordagem adotada pelo sistema de controle tem influência direta sobre algumas de suas características, inclusive as propriedades temporais. A abordagem de disparo por evento resulta numa utilização de recursos mais eficiente. Entretanto o desempenho temporal do sistema, no pior caso, depende da quantidade de eventos com chances de ocorrência simultânea. Por outro lado, a abordagem de disparo por tempo tem um comportamento temporal mais estável devido ao conhecimento

apriorístico dos instantes de chegada das tarefas. Assim, enquanto a abordagem de disparo por eventos normalmente é empregada para monitoração de eventos esporádicos tais como alarmes ou faltas, a abordagem de disparo por tempo é naturalmente apropriada para o tratamento de ações de controle executadas a partir da amostragem periódica das variáveis de estado do ambiente.

Os atributos do modelo de tarefas adotado pelo sistema determinam a abordagem a ser empregada no nível do suporte de comunicação. Suportes de comunicação empregados em sistemas com variáveis de estado que, por sua natureza, devem ser atualizadas regularmente a intervalos predeterminados implementam um controle autônomo, ou seja, todas as atividades de tráfego são controladas no nível do próprio suporte de comunicação.

No caso de adoção do critério de disparo por evento as ações dependem de controle externo pois, neste nível, as ações são disparadas em resposta a sinais gerados pela aplicação, que encontra-se fora da esfera de controle do suporte de comunicação.

### 2.3 Modelagem das Tarefas

As tarefas constituem-se em unidades lógicas de computação programadas para execução em um processador [7]. Uma tarefa é um programa seqüencial ativado a partir da ocorrência de determinados eventos. Tendo em vista o escopo desta Dissertação, os assuntos aqui abordados enfatizam os paradigmas de ativação das tarefas **periódicas** (subseção 2.3.1) e **esporádicas** (subseção 2.3.2).

Formalmente o modelo de tarefa adotado ( $\tau$ ) é representado como uma tupla com três parâmetros.

$$\tau = (C, T, D) \quad (2.1)$$

Cada um dos parâmetros da equação (2.1) tem o significado descrito a seguir:

- $C$  - **custo computacional**, que representa tempo máximo de execução (ou tempo de pior caso) da tarefa em um processador dedicado;
- $T$  - intervalo mínimo entre os instantes de chegada de duas instâncias consecutivas da tarefa  $\tau$ ;
- $D$  - **prazo relativo**, definido como o intervalo de tempo máximo admissível para a conclusão da execução da instância corrente da tarefa, a partir do instante de sua chegada.

A **instância** de uma tarefa é o objeto básico de escalonamento. Seus atributos predominantes são caracterizados pelos parâmetros do modelo representado pela equação (2.1) e pelo instante de **chegada**. Uma tarefa é **ativada** no instante de chegada da requisição de utilização do recurso e então permanece em uma fila aguardando para ser **liberada** para execução. Entre os instantes de **liberação** e de início da **execução** a tarefa estará **pronta** (*ready*). O tempo transcorrido entre os instantes de chegada e de conclusão é definido como **tempo de resposta** ( $R$ ) da tarefa (Fig. 2.3). Uma tarefa será denominada **atômica** quando a sua execução não puder ser interrompida por outra de prioridade mais elevada ou, em outras palavras, quando for **não preemptiva**.

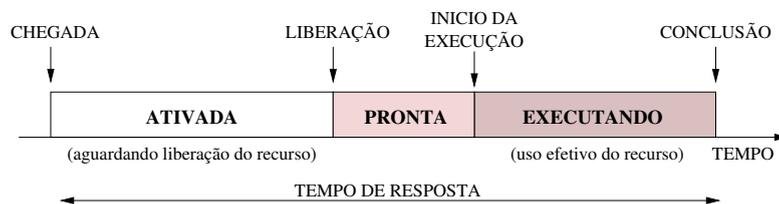


Figura 2.3: Tempo de Resposta de uma Tarefa

Neste trabalho assume-se que os eventos são gerados com uma frequência máxima; portanto o intervalo de tempo entre os instantes de chegada de duas instâncias consecutivas de uma tarefa qualquer tem um limite inferior. Além disso, a chegada da instância de uma tarefa resulta em uma única execução a ser iniciada no instante especificado pelo **algoritmo de escalonamento** (ver subseção 2.4).

A principal restrição imposta às tarefas críticas de um sistema de tempo real pode ser traduzida pela relação entre o custo, o tempo de resposta e o prazo, isto é:  $C \leq R \leq D$ .

### 2.3.1 Tarefas Periódicas

Classifica-se uma tarefa como **periódica** se o parâmetro  $T$  especificar um intervalo de tempo constante entre os instantes de chegada de duas de suas instâncias consecutivas. Nestas circunstâncias  $T$  é denominado como o **período** da tarefa. O comportamento de uma tarefa periódica  $\tau_p$  caracteriza-se pelas condições a seguir, considerando-se que  $A_k$  seja o instante de chegada de sua  $k^{\text{ésima}}$  instância:

1. A  $(k + 1)^{\text{ésima}}$  chegada de  $\tau_p$  ocorrerá no instante  $A_{k+1} = A_k + T$ .
2. A  $k^{\text{ésima}}$  execução de  $\tau_p$  deverá começar **não antes** de  $A_k$  e ser concluída **não depois** da expiração do prazo relativo, dado por  $A_k + D$ . Isto requer que  $C$  unidades de tempo de processamento sejam alocadas para execução de  $\tau_p$  no intervalo  $[A_k, A_k + D]$ .

Uma tarefa será classificada como **independente** se a sua execução não depender de sincronização [37]. Quando a liberação de uma tarefa periódica ocorrer simultaneamente com a liberação de todas as demais tarefas com prioridades de execução mais elevadas que a sua então o tempo de resposta desta tarefa será máximo (ou de pior caso). O momento correspondente a este evento é denominado **instante crítico** da tarefa [27].

### 2.3.2 Tarefas Esporádicas

Quando o parâmetro  $T$  especificar um intervalo **mínimo** entre os instantes de chegada de duas instâncias consecutivas da tarefa  $\tau_s$  esta será denominada **esporádica**. A restrição temporal relativa ao comportamento de uma tarefa esporádica é menos restritiva que o de uma tarefa periódica. Se  $A_k$  for o instante da  $k^{\text{ésima}}$  chegada de  $\tau_s$  então:

1. A chegada da  $(k + 1)^{\text{ésima}}$  instância de  $\tau_s$  ocorrerá **não antes de**  $A_{k+1} = A_k + T$ ; portanto  $A_{k+1} \geq A_k + T$ .
2. A  $k^{\text{ésima}}$  execução de  $\tau_s$  deverá começar **não antes de**  $A_k$  e ser concluída não depois do prazo relativo especificado ( $A_k + D$ ). Isto requer que  $C$  unidades de tempo de processamento sejam alocadas para execução de  $\tau_s$  no intervalo  $[A_k, A_k + D]$ .

Na prática, tarefas periódicas geralmente são encontradas em aplicações tais como controle de processos, que requerem amostragem e processamento contínuo de dados produzidos pelos sensores e que representam a imagem corrente do valor de certas variáveis de estado do ambiente.

As tarefas esporádicas são associadas a processamentos ativados em resposta a eventos, tais como requisições de usuários ou interrupções não periódicas de dispositivos. Estes eventos ocorrem de forma repetitiva, porém o intervalo de tempo entre ativações não é constante, podendo até mesmo ser arbitrariamente longos.

### 2.3.3 Tarefas Aperiódicas

As tarefas que têm instantes de chegada aleatórios, possivelmente em rápida sucessão e sem que seja conhecido o intervalo mínimo de tempo entre instantes de chegada consecutivos são classificadas como **aperiódicas**. Não é possível atribuir prazos rígidos a essa classe de tarefas, portanto o sistema de controle deverá atendê-las com uma abordagem de **melhor esforço** (ver seção 2.4).

### 2.3.4 Um Modelo de Sistema

O modelo de sistema que será adotado para discussão do escalonamento de tarefas é citado por Ausdsley e Burns [7]. A expressão (2.2) representa um conjunto com cardinalidade  $n$ , onde cada elemento ( $\tau_i$ ) é uma tarefa descrita pela expressão (2.1) e caracterizada pelos parâmetros: custo ( $C_i$ ), período ( $T_i$ ) e prazo ( $D_i$ ).

$$F = \{\tau_1, \tau_2, \dots, \tau_n\} \quad (2.2)$$

Inicialmente o modelo considera as seguintes restrições:

- Não há armazenamento temporário de dados (*data buffering*) e portanto  $C_i \leq D_i \leq T_i$  (isto é, cada tarefa  $\tau_i$  tem custo com duração no máximo igual ao respectivo prazo, que por sua vez deverá ser limitado por seu período);
- Tarefas periódicas e esporádicas;
- Os chaveamentos de contexto instantâneos, ou seja, com custo computacional nulo;
- Relações de precedência entre as tarefas ditas pelo algoritmo de escalonamento;
- Compartilhamento de recursos protegidos entre as tarefas;
- Custos computacionais limitados por valores máximos (ou de pior caso);
- Sistemas com diversos nós computacionais, com alocação estática ou dinâmica de tarefas.

## 2.4 Escalonamento de Tarefas de Tempo Real

Define-se como **escalonamento** a alocação do tempo de utilização de recursos computacionais, tais como processadores e meios de comunicação, de forma que seja garantido o atendimento dos requisitos especificados para as tarefas de tempo real. Neste sentido o escalonamento é uma atividade realizada em diversas etapas, dentre as quais destacam-se:

- Definição de um critério para determinação da ordem de processamento das tarefas, denominado **algoritmo de escalonamento**.
- Análise do comportamento do sistema no cenário de pior caso. Geralmente esta verificação é efetuada através dos **testes de escalonabilidade** (seção 2.5.1).
- Montagem da escala de execução das tarefas.

### 2.4.1 Abordagens de Escalonamento

A literatura registra diversas abordagens de escalonamento em função das características do modelo de tarefas adotado para o sistema. O esquema mostrado na Fig. 2.4 ilustra uma taxonomia das abordagens mais empregadas, dentre as quais destacam-se os grupos **garantia em tempo de projeto** (*pre-run-time*), **garantia em tempo de execução** (*run-time*) e **melhor esforço** (*best effort*) [12], [29], [34].

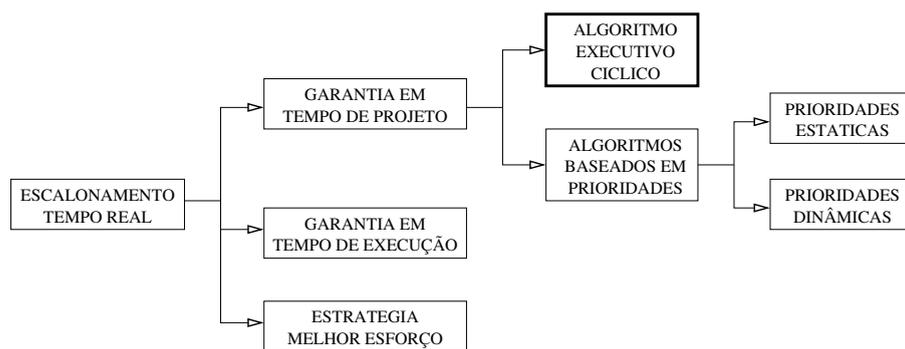


Figura 2.4: Taxonomia de Abordagens de Escalonamento de Tarefas de Tempo Real

Incluem-se no grupo garantia em tempo de projeto os métodos de construção da escala de execução baseados no conhecimento prévio do ambiente, traduzido pelas seguintes premissas: 1) os parâmetros das tarefas são estáticos; 2) os recursos disponíveis são suficientes para atendimento das restrições temporais das tarefas, em qualquer condição de operação, ou estado, do ambiente.

As abordagens de interesse desta dissertação pertencem ao grupo garantia em tempo de projeto. Estas abordagens, representadas pelo **executivo cíclico** (ver subseção 2.5.2) e pelos algoritmos **baseados em prioridades**, são tradicionalmente adotadas para o escalonamento de tarefas em sistemas de controle de processos, aí incluindo-se os suportes de comunicação. O algoritmo executivo cíclico é uma estrutura de controle baseada em uma tabela estática cujas entradas determinam a ordem de execução das tarefas periódicas do sistema de forma determinística [5].

Os algoritmos de escalonamento empregados na abordagem que leva em conta a perenidade das prioridades das tarefas, que podem permanecer fixas durante toda a vida do sistema (prioridades estáticas) ou ser alteradas em tempo de execução (prioridades dinâmicas) não exigem, a rigor, que seja construída uma tabela explícita. Em vez disso a abordagem baseada em prioridades concede o direito de ocupar o recurso, em tempo de execução, à tarefa com prioridade mais elevada.

Os sistemas que empregam a abordagem baseada em prioridades dinâmicas são mais flexíveis

no que diz respeito ao tratamento de eventos esporádicos, cujos instantes de ocorrência não podem ser determinados *a priori*. O seu custo de execução (*overhead*) é mais elevado que o da abordagem baseada em prioridades estáticas apesar de sua adoção resultar, teoricamente, em melhor utilização dos recursos do sistema. Deve ser ressaltado, entretanto, que os sistemas de tempo real críticos geralmente não admitem a ocorrência de eventos imprevistos, de forma que a escalonabilidade deve ser garantida em tempo de projeto.

Sistemas cujos instantes de chegada das tarefas não sejam previamente conhecidos, ou para os quais haja impossibilidade de determinação das condições de carga de pior caso, podem ser escalonados através da adoção das estratégias associadas ao grupo garantia dinâmica. Neste grupo cada tarefa que seja ativada é submetida, em tempo de execução, a um **teste de aceitação** [12] que verifica a escalonabilidade do conjunto formado pela tarefa candidata à utilização do recurso compartilhado e por todas as demais tarefas que encontrem-se na fila de pronto, aguardando para serem liberadas.

O grupo menor esforço tenta encontrar uma escala em tempo de execução porém não oferece garantia de atendimento das tarefas. Por esta razão sua aplicabilidade restringe-se a sistemas de tempo real brandos.

Os mecanismos de escalonamento podem ainda ser classificadas como **preemptivos** [27] ou **não preemptivos** [24], dependendo de sua capacidade de suspender arbitrariamente a execução de uma tarefa e retomá-la posteriormente sem afetar o seu resultado, exceto a alteração do seu tempo de resposta.

No contexto do escalonamento preemptivo a tarefa com prioridade mais elevada poderá interromper a execução de uma tarefa com menor prioridade. Por outro lado em esquemas não-preemptivos a tarefa que estiver sendo processada continuará sua execução até a conclusão, mesmo que tenha prioridade mais baixa que uma tarefa de maior prioridade que tenha sido ativada. Esta condição é denominada **inversão de prioridades** devido à não preempção, que resulta no **bloqueio** da execução de uma tarefa de prioridade mais elevada por uma tarefa com prioridade de valor inferior.

#### 2.4.2 Inserção de Tempo Ocioso

A adoção de algoritmos não preemptivos baseados em prioridades estáticas pode resultar na redução da escalonabilidade das tarefas do sistema [44]. Uma técnica geralmente empregada para garantir o escalonamento de tarefas periódicas em sistemas baseados em algoritmos não preemptivos

denomina-se **inserção de tempo ocioso** (*inserted idle time*) [17]. O mecanismo baseia-se em retardar a liberação das tarefas com durações mais longas e aguardar a liberação de tarefas com prazos menores e prioridades mais elevadas de forma que elas sejam executadas em primeiro lugar. De outra forma estas tarefas perderiam seus prazos.

### 2.4.3 Critérios de Atribuição de Prioridades

A solução para o problema de escalonamento de tarefas periódicas e independentes em um processador foi apresentado, de forma pioneira, no artigo de Liu e Layland [27]. Neste artigo o algoritmo denominado **taxa monotônica** (*Rate Monotonic - RM*) é descrito como um esquema de atribuição de prioridades às tarefas na razão inversa de seus períodos - quanto menor for o período, maior será o valor da prioridade atribuída. Isto é,  $\forall i, j : T_i < T_j \Rightarrow P_i > P_j$ .

Se, além de tarefas periódicas, também for considerado o escalonamento das tarefas esporádicas do sistema, não será adequado considerar que os prazos relativos sejam iguais aos respectivos períodos. Neste caso poderá ser mais conveniente a adoção de um esquema de atribuição de prioridades baseado nos prazos relativos das tarefas - terão prioridades mais elevadas as tarefas com menores prazos relativos, ou seja,  $\forall i, j : D_i < D_j \Rightarrow P_i > P_j$ . Este esquema de atribuição de prioridades denomina-se **prazo monotônico** (*deadline monotonic - DM*).

As prioridades atribuídas com base nos algoritmos RM e DM permanecem fixas durante toda a vida do sistema. Por esta razão os mecanismos de escalonamento que empregam estes princípios são denominados algoritmos baseados em **prioridades estáticas**.

O artigo de Liu e Layland [27] também descreve um esquema que elege a próxima tarefa a ter direito de usar o recurso em função da proximidade de vencimento do seu prazo de execução. O algoritmo *Earliest Deadline First (EDF)* emprega um mecanismo de atribuição de **prioridades dinâmicas** no qual tem prioridade mais elevada a tarefa cujo prazo estiver mais próximo de se expirar.

## 2.5 Análise das Abordagens de Escalonamento

### 2.5.1 Testes de Escalonabilidade

O emprego sistemático de determinadas ferramentas matemáticas para comprovar a existência de uma escala capaz de atender aos prazos de execução especificados para as tarefas sob qualquer

condição de operação denomina-se **análise de escalonabilidade**. Portanto a análise de escalonabilidade tem como objetivo encontrar uma escala **realizável** para o sistema em função das premissas do algoritmo de escalonamento adotado. O conjunto de equações matemáticas empregado como suporte para esta atividade de validação de um algoritmo de escalonamento denomina-se **teste de escalonabilidade**.

A realização dos testes requer a eleição de métricas que reflitam o nível de ocupação do recurso requisitado ou o tempo de resposta do sistema. Considerando-se um conjunto de tarefas modelado segundo a expressão (2.2) (ver subseção 2.3.4), a fração do tempo de ocupação do recurso pelas tarefas denomina-se **fator de utilização**, que é definido pela seguinte expressão:

$$U = \sum_{i=1}^n \frac{C_i}{T_i} \quad (2.3)$$

Em função da abrangência dos resultados, os testes de escalonabilidade classificam-se como **suficientes**, **necessários** ou **exatos** [25].

Os testes suficientes geralmente apresentam certa simplicidade de aplicação, porém associada ao risco que rejeitar conjuntos de tarefas escalonáveis. Todos os conjuntos de tarefas aceitos pelos testes suficientes são escalonáveis.

Os testes necessários são menos restritivos que os testes suficientes. Se um determinado conjunto de tarefas não passar neste tipo de teste certamente ele não será escalonável. Entretanto, se um conjunto de tarefas for aceito não significa que ele será escalonável.

Os testes exatos são capazes de identificar conjuntos de tarefas escalonáveis e também aqueles que não são escalonáveis (Fig. 2.5).

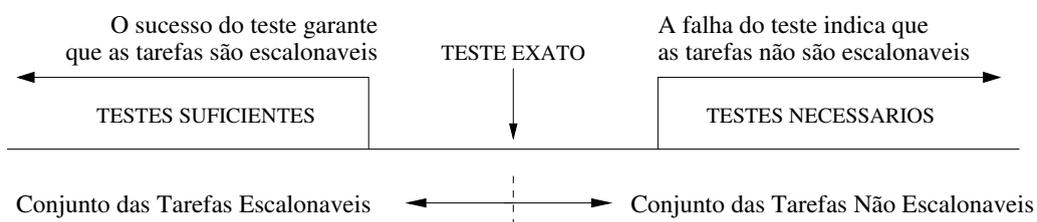


Figura 2.5: Testes de Escalonabilidade

A escalonabilidade poderá também ser verificada através do cálculo do tempo de resposta das tarefas do sistema, efetuado de forma individualizada. Este método apresenta vantagens em relação aos

testes baseados na utilização por serem sempre exatos e também por serem aplicáveis de uma forma mais genérica [9]. Uma tarefa  $\tau_i$  com custo computacional  $C_i$ , terá o tempo de resposta determinado por:

$$R_i = C_i + I_i \quad (2.4)$$

Na equação (2.4) o termo  $I_i$  representa a **interferência** máxima que  $\tau_i$  poderá sofrer de tarefas que tenham prioridades mais elevadas no intervalo  $A_k + R_i$ , onde  $A_k$  é o instante de chegada da tarefa (ver subsecção 2.5.4).

A inclusão do tempo máximo de duração do bloqueio de  $\tau_i$  por tarefas com prioridades inferiores, condição denominada **inversão de prioridades**, tem como consequência o aumento do tempo de resposta. Portanto a parcela  $B_i$  mostrada na equação (2.5) representa um certo grau de redução da escalonabilidade do sistema [9].

$$R_i = C_i + I_i + B_i \quad (2.5)$$

### 2.5.2 Executivo Cíclico

O algoritmo de escalonamento executivo cíclico baseia-se em uma tabela ou grade de tempo (*time-grid*) construída em tempo de projeto, que contém a seqüência de execução das tarefas do sistema. Esta ordem é determinística, uma vez que os instantes de início de execução das tarefas são conhecidos e os respectivos instantes de conclusão são previsíveis [5]. Em tempo de execução o escalonador desempenha a função de **despachante**, determinando a ordem de ocupação do recurso pelas tarefas periódicas [12].

O algoritmo executivo cíclico é uma estrutura ou programa de controle que intercala as execuções de tarefas periódicas em um processador com base em uma escala formada por um **ciclo maior** e por um ou mais **ciclos menores** [5]. O ciclo maior, denominado **macro-ciclo**, tem duração constante e igual ao **mínimo múltiplo comum** (MMC) dos períodos das  $n$  tarefas do sistema.

$$m = MMC(T_i) \quad (i = 1, \dots, n) \quad (2.6)$$

Uma heurística geralmente empregada para construção da escala de execução do algoritmo executivo cíclico considera a existência de uma relação harmônica entre as frequências das tarefas, de forma que seus períodos sejam múltiplos do tempo de duração do ciclo menor. Com base nesta hipótese a duração do ciclo menor, também conhecido como **micro-ciclo** [1] [41] ou **ciclo elementar** [43], será

igual ao **máximo divisor comum** (MDC) dos períodos das  $n$  tarefas do conjunto. Nesta Dissertação será adotada a designação ciclo elementar (CE) como padrão.

$$e = MDC(T_i) \quad (i = 1, \dots, n) \quad (2.7)$$

A relação entre a largura do macro-ciclo e a duração do ciclo elementar corresponde ao número de ciclos elementares contidos em um macro-ciclo.

$$N = \frac{m}{e} \quad (2.8)$$

Para montagem da tabela de execução de um conjunto de tarefas modeladas com base nos atributos definidos na seção 2.3, cuja largura do ciclo elementar seja  $e$ , deverá ser observado que o custo deverá estar relacionado com o prazo de tal forma que  $C_i \leq e \leq D_i$ .

A construção da grade de execução das tarefas segundo a abordagem executivo cíclico geralmente emprega a técnica de inserção de tempo ocioso para garantir a escalonabilidade do sistema (subseção 2.4.2). Quando uma tarefa que estiver pronta não puder ser liberada para execução por não haver tempo disponível em um determinado ciclo elementar este é dito estar **sobrecarregado**. O processamento da tarefa será então atrasado para os ciclos elementares subsequentes até que seja encontrado um CE com tempo suficiente para acomodá-la. Sempre que isto ocorrer, o recurso será mantido ocioso ou, no caso de meios de comunicação, serão transmitidos padrões de dados com formatos especiais denominados **identificadores de preenchimento** (*padding identifiers*) nos instantes finais dos ECs sobrecarregados.

O escalonamento das tarefas periódicas listadas na Tab. 2.1 empregando a abordagem executivo cíclico com base em uma escala construída a partir do algoritmo de atribuição de prioridades RM ( $D_i = T_i$ ) produzirá a grade de tempo mostrada na Fig. 2.6.

Tarefa	Parâmetros		
	$C_i$	$T_i$	$D_i$
$\tau_1$	0,20	1	1
$\tau_2$	0,25	2	2
$\tau_3$	0,25	3	3
$\tau_4$	0,35	4	4
$\tau_5$	0,40	4	4

Tabela 2.1: Conjunto de Tarefas Periódicas

A Fig. 2.6 mostra que, apesar de todas as tarefas serem ativadas em  $t_0 = 0$ , para que o conjunto seja escalonável é necessário que a liberação das mensagens  $\tau_4$  e  $\tau_5$  ocorra somente no segundo CE, resultando na ociosidade do recurso durante algum tempo nos instantes finais dos ciclos elementares.

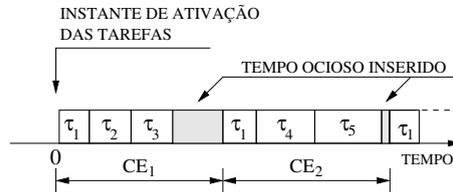


Figura 2.6: Inserção de Tempo Ocioso

A verificação da escalonabilidade do conjunto de tarefas é feita durante a montagem da grade de tempo e por esta razão não são necessários testes adicionais para o algoritmo executivo cíclico. Kopetz [25] considera a montagem da escala um teste suficiente de escalonabilidade. Portanto o esquema de escalonamento executivo cíclico é “testado durante a construção”.

### 2.5.3 Testes de Escalonabilidade para o Algoritmo RM

O artigo pioneiro de Liu e Layland [27] apresenta um teste para o algoritmo RM baseado no nível de utilização do recurso. O sucesso deste teste, que é definido pela equação (2.9), garante que o processamento de todas as tarefas do conjunto será concluído antes dos respectivos prazos.

$$U = n(2^{\frac{1}{n}} - 1) \quad (2.9)$$

A equação (2.9) representa o limite superior da utilização do processador, que decresce desde 0,83, quando  $n = 2$ , até  $\log_e 2 = 0,693$  quando  $n$  tende para o infinito. Se um conjunto de tarefas modelado pela equação (2.2) tiver um fator de utilização inferior a este valor a sua escalonabilidade estará garantida pelo algoritmo RM em sua abordagem preemptiva.

Liu e Layland [27], demonstraram que o algoritmo RM é **ótimo** no sentido de que se um conjunto de tarefas periódicas for escalonável através de qualquer outro esquema de escalonamento preemptivo baseado em prioridades fixas também o será usando-se o algoritmo RM.

O trabalho de Sha e Goodenough [37] introduz uma extensão ao teste de Liu e Layland, através da inclusão dos intervalos de tempo nos quais tarefas com prioridades mais elevadas são bloqueadas

por tarefas menos prioritárias:

$$U \leq i(2^{\frac{1}{i}} - 1) - \frac{B_i}{T_i} \quad (2.10)$$

A inclusão do tempo de bloqueio ( $B_i$ ) de forma explícita na equação (2.10) tem como efeito a redução da escalonabilidade do conjunto de tarefas pois este termo pode ser interpretado como um acréscimo na utilização do recurso pela tarefa  $\tau_i$ .

As restrições representadas pelas equações (2.9) e (2.10) traduzem condições suficientes de escalonabilidade. Entretanto estes testes não são necessários, pois podem existir conjuntos de tarefas escalonáveis pelo algoritmo RM apesar de terem fatores de utilização superiores a estes limites teóricos. Em outras palavras, estes conjuntos de tarefas seriam rejeitados pelos testes acima descritos.

Um teste exato aplicável ao algoritmo RM, e que portanto define uma condição necessária e suficiente de escalonabilidade, encontra-se no trabalho de Lehoczky *et al.* [26]. O artigo considera um conjunto de tarefas modelado pela expressão (2.2) cujos elementos são ordenados de maneira não decrescente de seus períodos. A **demanda cumulativa** sobre o processador devida a todas as tarefas de prioridade igual ou maior que  $\tau_i$  no intervalo  $[0, t]$  é dada por:

$$W_i(t) = \sum_{j=1}^i C_j \left\lceil \frac{t}{T_j} \right\rceil \quad (2.11)$$

A variável  $L_i(t)$ , que representa o nível relativo de utilização do recurso durante o intervalo  $t$ , é definida como:

$$L_i(t) = \frac{W_i(t)}{t} \quad (2.12)$$

Com base nestas equações poderão ser derivados os critérios de escalonabilidade de uma tarefa ou de um conjunto de tarefas periódicas através do algoritmo RM:

- A tarefa  $\tau_i$  será escalonável para qualquer instante de chegada (*phasing*) se e somente se:

$$L_i = \min_{(0 < t < T_i)} L_i(t) \leq 1 \quad (2.13)$$

- O conjunto de tarefas  $F$  será escalonável para qualquer instante de chegada  $A_k$  se e somente se:

$$L = \max_{(1 \leq i \leq n)} L_i \leq 1 \quad (2.14)$$

A escalonabilidade assegurada pela aplicação das equações (2.13) e (2.14) requer a minimização da variável contínua  $t$  no intervalo  $[0, T_i]$ . O atendimento do prazo da tarefa  $\tau_i$  pode ser testado através da verificação do nível de utilização relativo nos instantes onde ocorrem os mínimos locais, que são definidos pelos múltiplos de  $T_j \leq T_i$  para  $1 \leq j \leq i$  e expressos por:

$$S_i = \{kT_j \mid j = 1, \dots, i; k = 1, \dots, \lfloor \frac{T_i}{T_j} \rfloor\} \quad (2.15)$$

$S_i$  é um conjunto cujos elementos são os **pontos de escalonamento** da tarefa  $\tau_i$ , seu prazo e os instantes de chegada das tarefas com prioridades superiores à de  $\tau_i$ , considerando-se que as chegadas ocorram no instante crítico ( $A_k = 0$ ). Neste caso a expressão (2.13) poderá ser reescrita como:

$$L_i = \min_{(t \in S_i)} L_i(t) \leq 1 \quad (2.16)$$

A condição de escalonabilidade do conjunto  $F$  será expressa pela equação (2.14) sendo que as demandas relativas  $L_i$  são calculadas, neste caso, a partir da equação (2.16).

As equações (2.11) a (2.16) são aplicáveis a tarefas pertencentes a um conjunto  $F$  com qualquer de chegada em relação ao instante crítico (fase). O seu efeito, portanto, é o de aumentar o limite de utilização definido pela equação (2.9).

#### 2.5.4 Teste de Escalonabilidade para o Algoritmo Prazo Monotônico - (DM)

O teste de escalonabilidade desenvolvido por Audsley [3] baseia-se em um mecanismo de verificação se cada uma das  $n$  tarefas do conjunto  $F$  (subseção 2.3.4) que tenham sido liberadas no instante crítico completam a execução antes do prazo relativo especificado ( $D_i$ ), portanto:

$$C_i + I_i \leq D_i \quad \forall i : 1 \leq i \leq n \quad (2.17)$$

Na equação (2.17) a **interferência** ( $I_i$ ) representa o tempo máximo que a tarefa  $\tau_i$  deve aguardar para iniciar sua execução devido à ocupação do recurso por tarefas com prioridades mais elevadas, que tenham sido liberadas antes do término do prazo de  $\tau_i$ .

$$I_i = \sum_{j=1}^{i-1} \lceil \frac{D_i}{T_j} \rceil C_j \quad (2.18)$$

A equação (2.17) poderá ser normalizada por  $D_i$ , resultando:

$$\frac{C_i}{D_i} + \frac{I_i}{D_i} \leq 1 \quad (2.19)$$

O teste de escalonabilidade representado pela equação (2.19) traduz uma condição suficiente mas não necessária, pois a interferência quantificada pela equação (2.18) inclui o custo de execução das tarefas de prioridades mais elevadas que poderiam começar antes de  $D_i$  e completar a execução depois de  $D_i$ . Neste caso  $I_i$  poderia ser maior que a verdadeira interferência sofrida por  $\tau_i$ .

Um teste necessário e suficiente para o escalonamento DM pode ser realizado através do algoritmo apresentado a seguir, que considera que todas as tarefas são liberadas no instante crítico [3]. O primeiro instante em que  $\tau_i$  poderia completar a sua execução pode ser calculado pela equação (2.20):

$$t_0 = \sum_{j=1}^i C_j \quad (2.20)$$

E a restrição de escalonabilidade no ponto  $t_0$  é calculada como:

$$\frac{I_i^{t_0}}{t_0} + \frac{C_i}{t_0} \leq 1 \quad (2.21)$$

$I_i^{t_0}$  é a interferência sofrida por  $\tau_i$  no intervalo  $[0, t_0)$ , dada por:

$$I_i^{t_0} = \sum_{j=1}^{i-1} \left[ \frac{t_0}{T_j} \right] C_j \quad (2.22)$$

A condição de escalonabilidade expressa pela equação (2.21) considera que cada uma das tarefas com prioridade superior a  $\tau_i$  será executada apenas uma vez no intervalo  $[0, t_0)$ . Portanto o teste falhará se ocorrer a liberação de uma tarefa de prioridade mais elevada neste intervalo. Por essa razão o teste deverá continuar com a resolução da série de equações definidas pelas expressões (2.23) a (2.25) de forma a varrer todo o intervalo  $[0, D_i]$ .

$$\frac{I_i^{t_k}}{t_k} + \frac{C_i}{t_k} \leq 1 \quad (2.23)$$

$$I_y^x = \sum_{z=1}^{y-1} \left[ \frac{x}{T_z} \right] C_z \quad (2.24)$$

$$t_k = I_i^{t_{k-1}} + C_i \quad (2.25)$$

O instante inicial  $t_0$  é calculado pela equação (2.20). Se o resultado de pelo menos uma dessas equações produzir resultado verdadeiro no intervalo  $[0, D_i]$  então  $\tau_i$  será escalonável neste intervalo. O procedimento deverá ser encerrado quando  $t_k > D_i$ . A partir deste ponto  $\tau_i$  não será escalonável.

O teste definido pelas equações (2.23) a (2.25) poderá também ser aplicado para a verificação da escalonabilidade de tarefas esporádicas.

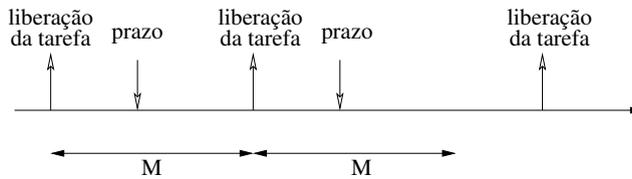


Figura 2.7: Escalonamento de Tarefa Esporádica

A Fig. 2.7 serve para ilustrar a escala de execução da tarefa esporádica  $\tau_s$ . O menor intervalo entre instâncias da tarefa  $\tau_s$  é representado por  $M$ . No intervalo  $[t, t + M]$   $\tau_s$  apresenta comportamento similar ao de uma tarefa periódica com período  $M$ , correspondendo à máxima frequência de ativações e, como conseqüência, impondo a carga máxima ao processador. Para que seja garantido o atendimento do prazo da tarefa esporádica, o seu custo ( $C_s$ ) deverá ser atendido no intervalo  $[t, t + D]$ .

Em resumo, para fins de análise de escalonabilidade, a tarefa esporádica  $\tau_s$  poderá ser descrita como uma tarefa periódica com período  $M$ . Se o prazo de uma tarefa aperiódica puder ser garantido quando sua liberação ocorrer no instante crítico então ela será escalonável e, neste caso, todas as tarefas, periódicas e aperiódicas, atenderão os seus respectivos prazos.

### 2.5.5 Testes de Escalonabilidade para o Algoritmo EDF

Liu e Layland [27] demonstra que um conjunto de tarefas periódicas e independentes liberadas no instante crítico ( $A_k = 0$ ) é escalonável em um processador quando for permitida a preempção, desde que seja atendida a seguinte condição:

$$U = \sum_{i=1}^n \frac{C_i}{T_i} \leq 1 \quad (2.26)$$

Jeffay *et al.* [24] apresenta um teste de escalonabilidade válido para sistemas não preemptivos em geral, que considera também conjuntos de tarefas periódicas e esporádicas.

$$\forall i, 1 < i \leq n; \forall L, T_1 < L < T_i : L \geq C_i + \sum_{j=1}^{i-1} \left\lfloor \frac{L-1}{P_j} \right\rfloor C_j \quad (2.27)$$

### 2.5.6 Aplicabilidade dos Algoritmos de Escalonamento

Apesar das limitações apontadas na literatura o algoritmo executivo cíclico é uma alternativa de escalonamento encontrada, com certa frequência, em aplicações industriais [12], [13], [37]. Uma das principais restrições associadas à implementação desta abordagem é a construção da escala, cuja dificuldade apresenta crescimento exponencial em relação ao número de tarefas do sistema. Esta restrição pode ser contornada, ou pelo menos minimizada, através da adoção de técnicas de construção da escala baseadas em algoritmos tais como o RM ou o EDF. Desta maneira o algoritmo executivo cíclico pode tornar-se uma alternativa atraente para sistemas que apresentem certa simplicidade de implementação ou cujas tarefas tenham características uniformes [44].

Com relação ao algoritmo RM, existem ferramentas matemáticas que podem ser aplicadas para determinação do limite de utilização do processador, ou seja, definição da carga abaixo da qual é garantida a escalonabilidade das tarefas do sistema. O teste desenvolvido por Liu e Layland (subseção 2.5.3), que tem como uma de suas principais virtudes a simplicidade de aplicação, traduz um certo pessimismo pois limita a 69,3% o nível de utilização do processador à medida que aumenta o número de tarefas.

O trabalho de Lehoczcy *et al.* [26] introduz um teste exato cujo emprego resulta em uma maior utilização do processador e destaca algumas das vantagens do algoritmo RM: 1) assegura o atendimento dos requisitos temporais das tarefas em situações de sobrecarga transitória; 2) proporciona tempos de resposta rápidos para tarefas aperiódicas, através do emprego de mecanismos tais como o servidor esporádico ou o algoritmo de troca de prioridade estendida; 3) pode ser modificado para atender os requisitos de sincronização de tarefas usando o protocolo de teto de prioridade; 4) pode ser empregado para escalonar tarefas onde for permitida computação imprecisa; 5) apresenta facilidade de implementação em processadores, em controladores de entrada/saída (E/S) e em suportes de comunicação utilizados em sistemas de tempo real. Esta última propriedade justifica a adoção do algoritmo RM como disciplina de escalonamento do tráfego nas redes de comunicação dos sistemas computacionais de controle de processos.

Pode ainda ser observado que, se os prazos das tarefas forem iguais aos respectivos períodos ( $D_i = T_i$ ) então o algoritmo taxa monotônica será um caso particular do algoritmo prazo monotônico. Portanto o algoritmo DM é uma generalização do algoritmo RM [38].

Uma versão não-preemptiva do algoritmo EDF aplicável para o escalonamento de tarefas

periódicas e também de tarefas esporádicas em um processador é apresentada no trabalho de Jeffrey *et al.* [24]. No artigo é demonstrada a universalidade do algoritmo EDF para o escalonamento de tarefas esporádicas em sistemas nos quais não é admitida a preempção de tarefas.

## 2.6 Comunicação Tempo Real

O suporte de comunicação é um componente crítico dos sistemas computacionais de controle distribuído, uma vez que sua perda poderá resultar na indisponibilidade dos serviços considerados essenciais para o intercâmbio de dados entre as tarefas residentes em diferentes nós computacionais [25].

As tarefas de um sistema de tempo real geralmente efetuam o intercâmbio de informações através de transações de **mensagens**. Mensagens são estruturas de dados que contém, além dos objetos de imagem do processo, informações relevantes para o controle do tráfego entre os transmissores e os receptores (usuários ou consumidores) destes dados.

Certos protocolos de comunicação industrial, tais como a rede WorldFIP (Capítulo 3), adotam os termos **quadro** (*frame*) para designar as transações de variáveis entre as entidades do processo controlado e **mensagem** (*message*) para identificar os serviços de gerenciamento dos recursos do próprio suporte de comunicação. Nesta Dissertação o sentido pretendido para um ou outro termo utilizado poderá ser apreendido a partir do contexto em que o mesmo estiver inserido.

As transações de mensagens entre as tarefas de tempo real críticas devem ser realizadas através de um suporte de comunicação que apresente, como um de seus principais atributos, **previsibilidade**, ou seja, as transações de dados devem ocorrer em janelas de tempo limitadas.

### 2.6.1 Modelo Hierárquico do Sistema de Comunicação

Os conceitos da arquitetura CIM (*computer integrated manufacturing*) [30] [32] podem ser adotados para definição do modelo de estrutura hierárquica do sistema de supervisão e controle do processo de geração de energia elétrica abordado no Capítulo 5.

Na estrutura mostrada na Fig. 2.8 cada atividade relevante do processo é mapeada a um nível hierárquico classificado segundo a natureza de seus requisitos funcionais predominantes. O nível 4 suporta os atributos de natureza mais elevada, associados à gerência administrativa da planta. Nas

camadas mais próximas do processo encontra-se a interface de instrumentação (ver seção 2.1), que inclui entre seus principais recursos a rede de campo (nível 0) e os dispositivos de entrada e saída de dados (nível 0).

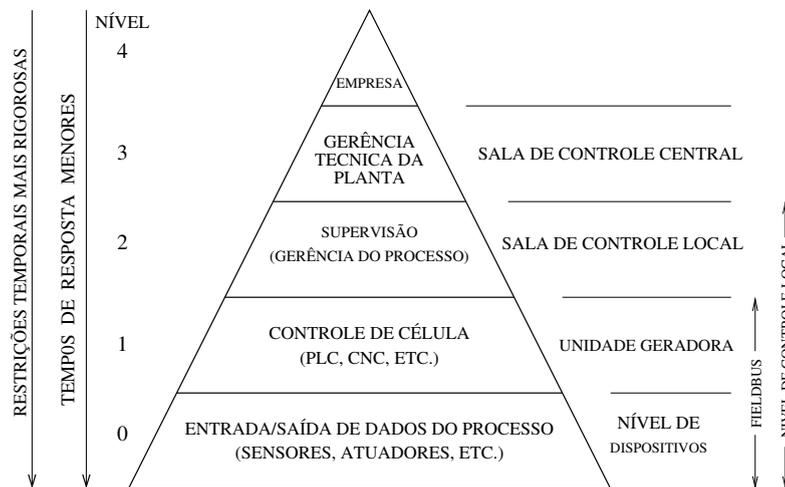


Figura 2.8: Modelo Hierárquico do Processo de Produção Industrial de Energia Elétrica

À medida que as atividades aproximam-se dos níveis inferiores da hierarquia aumentam o rigor das restrições temporais das tarefas computacionais e a demanda por tempos de resposta mais curtos. Portanto os intercâmbios de informações nos níveis 0 e 1 devem ser implementados através de suportes de comunicação com resposta temporal determinista, de modo a garantir o atendimento dos requisitos impostos pelo ambiente, preservando as consistências espacial e temporal dos dados do sistema.

### 2.6.2 Modelo de Tráfego de Tempo Real Adotado

Na literatura especializada, o termo **tráfego de tempo real** geralmente refere-se ao **tráfego síncrono**, ou seja, ao fluxo de mensagens transmitidas através do meio de comunicação em uma base temporalmente contínua. Este tráfego inclui mensagens periódicas e esporádicas, que requerem algum grau de garantia de entrega com pontualidade. Devem também ser consideradas as mensagens aperiódicas (ou assíncronas), que têm restrições temporais brandas e esperam que o sistema as envie numa base de melhor esforço [28].

No modelo considerado neste trabalho, o direito de acesso para transmissão de cada mensagem é definido por uma prioridade cuja atribuição pode ser estática ou dinâmica, dependendo da abordagem de escalonamento adotada (subseção 2.4.3). O modelo é não preemptivo, e portanto admite que o

início da transmissão de uma mensagem seja bloqueada por outra de prioridade inferior durante um período limitado de tempo.

A equação (2.28) define o modelo para um conjunto de mensagens onde cada elemento  $m_i$  é caracterizado por quatro parâmetros: o identificador da estação transmissora ( $k$ ), a largura ( $C_i$ ), o intervalo de tempo entre duas instâncias consecutivas ( $T_i$ ) e o prazo relativo fim-a-fim ( $D_i$ ), que quantifica o limite superior do tempo para cada instância ser disponibilizada no nó de destino, a partir do instante de sua entrega pela tarefa de origem ao suporte de comunicação.

$$m_i = (k, C_i, T_i, D_i) \quad (2.28)$$

Com relação à forma de requisição dos serviços de comunicação, as mensagens podem ser classificadas como:

**Mensagens Periódicas.** Esta classe de mensagens caracteriza-se como uma seqüência de requisições de transmissão ativadas de forma contínua, com intervalo de tempo constante entre duas instâncias consecutivas. Neste caso o parâmetro  $T_i$  é denominado **período**. A literatura de comunicação tempo real classifica o tráfego de mensagens periódicas como **modelo de taxa máxima** (*peak rate model*) [2].

**Mensagens Esporádicas.** Os dados gerados em resposta a eventos nos quais os intervalos de ocorrência de suas instâncias apresentam um padrão de chegadas aleatório são transmitidos por mensagens esporádicas, que poderão também ser matematicamente descritas pela expressão (2.28). Neste caso o parâmetro  $T_i$  representa o intervalo de tempo **mínimo** entre duas instâncias consecutivas de  $m_i$ .

**Mensagens Aperiódicas.** A transmissão de mensagens geradas por tarefas aperiódicas não tem, por definição, prazos relativos definidos. Entretanto, o atraso médio das mensagens aperiódicas devem ser mantidos o menor possível [28].

### 2.6.3 Atrasos na Transmissão de Mensagens

Para garantir o atendimento dos requisitos temporais das tarefas o tempo transcorrido entre a requisição de transmissão da informação pelo nó transmissor e a sua recepção pelo nó receptor deve ser limitado. Este tempo é denominado **atraso fim-a-fim** na transmissão da mensagem. Tindell [40] define quatro parcelas que contribuem para este atraso:

- **atraso de geração:** tempo de pior caso tomado entre o instante de chegada dos dados enviados por uma tarefa transmissora no suporte de comunicação e a composição e enfileiramento da mensagem. Isto inclui o custo de processamento da aplicação para gerar a estrutura da mensagem e o tempo consumido para sua colocação na posição correta na fila de pronto de transmissão.
- **atraso de enfileiramento:** tempo necessário para que a mensagem ganhe direito de acesso ao meio de comunicação depois de ter sido colocada na fila de transmissão. O atraso de enfileiramento é o tempo que a mensagem aguarda para ser removida da fila pela tarefa responsável pelo escalonamento no suporte de comunicação. Em um suporte de comunicação compartilhado, a mensagem disputa o direito de acesso ao meio de transmissão com as mensagens dos outros nós.
- **atraso de transmissão:** tempo de tráfego da mensagem pelo meio físico, após ter sido retirada da fila de mensagens prontas para transmissão.
- **atraso de entrega:** tempo de processamento da mensagem no nó de destino antes de sua entrega à tarefa usuária da informação. Esta atividade inclui funções como decodificação dos identificadores de mensagem, cópia das mensagens nos *buffers* de destino e a notificação ao despachante da chegada da mensagem. Esta função é importante, uma vez que a tarefa receptora da mensagem poderia estar bloqueada, esperando a sua chegada. Na prática o atraso de entrega poderá representar uma parte significativa do atraso fim-a-fim.

Além do atraso na transmissão, a variação máxima do atraso de transmissão entre as instâncias do fluxo de mensagens  $m_i$ , denominada **jitter de transmissão** ( $J_i$ ), constitui-se também em um fator importante a ser considerado na análise do escalonamento do tráfego de tempo real.

#### 2.6.4 Escalonamento de Mensagens de Tempo Real

O escalonamento do tráfego de mensagens através de uma rede de comunicação tempo real tem por finalidade garantir que as restrições temporais associadas às tarefas de um sistema distribuído sejam atendidas, mesmo que as variáveis do ambiente sejam produzidas (geradas) ou utilizadas (consumidas) por tarefas residentes em diferentes nós computacionais.

A teoria de escalonamento de tempo real foi desenvolvida e inicialmente aplicada para a ordenação da execução de tarefas em um único processador. No caso da comunicação tempo real

o meio de transmissão torna-se o recurso central a ser compartilhado pelas mensagens que, neste caso, desempenham um papel equivalente ao das tarefas que disputam o direito de utilização do processador. Com base nesta analogia a maior parte dos métodos desenvolvidos para o escalonamento de tarefas em um processador são aplicáveis para o escalonamento de mensagens através de um canal de comunicação [2].

Deve ser observado, entretanto, que os protocolos de comunicação empregados em redes de acesso compartilhado têm natureza fundamentalmente distribuída, enquanto a execução de tarefas em um processador, na maioria dos casos, baseia-se em algoritmos de escalonamento centralizado [29].

**Atomicidade e frequência de ativações** são duas características que exercem influência fundamental no comportamento temporal das mensagens transacionadas nos níveis inferiores da estrutura hierárquica dos sistemas de controle de processos. As mensagens que trafegam nestes níveis têm larguras relativamente pequenas, com valores típicos da ordem de 1 até 128 bytes. Então, justificado pela busca da eficiência no uso do suporte de comunicação, geralmente o escalonamento de tráfego é do tipo não preemptivo [1]. Além disso, as mensagens de redes de tempo real são implementadas como **quadros** (*frames*), que são transmitidas como unidades atômicas (indivisíveis) pela camada de enlace (ver subseção 3.1.2). Assim, uma vez iniciada a transmissão, ela não poderá mais ser interrompida.

## Capítulo 3

# A Rede de Campo Padrão WorldFIP

### 3.1 Redes de Comunicação do Tipo Barramento de Campo

Redes de comunicação denominadas **barramentos de campo** (*fieldbus*), ou simplesmente **redes de campo**, são tipos especiais de redes de área local (LAN) dedicadas a aplicações orientadas à interconexão de sensores, atuadores e controladores em máquinas e em sistemas de controle de processos no nível de chão de fábrica. Portanto elas constituem-se em um dos componentes essenciais de aplicações de controle de processos em tempo real.

O termo *fieldbus* foi adotado pela IEC [18] para definir uma classe de barramento digital, bi-direcional, serial e multi-ponto, para suceder a interface analógica 4-20mA até então utilizada como padrão de comunicação em sistemas de controle de processos industriais. É interessante registrar que a primeira norma internacional tendo como objeto a comunicação no nível de campo, a IEC 381-1, publicada em 1965, padronizou o sinal analógico 4-20 mA utilizando o par trançado de fios de cobre como meio físico de transmissão [1].

#### 3.1.1 Protocolos de Controle de Acesso ao Meio

Tipicamente as redes de comunicação no nível de campo são do tipo de difusão, como a maioria das redes locais, onde diversos nós compartilham um canal de comunicação. O problema clássico das redes compartilhadas é a administração das colisões, que ocorrem quando dois ou mais nós tentam transmitir suas mensagens simultaneamente. Estas situações são resolvidas pelos protocolos de controle de acesso ao meio (*medium access control - MAC*) [29].

Os métodos de controle de acesso ao meio desempenham um papel importante no escalonamento

do tráfego através do suporte de comunicação de sistemas de tempo real, uma vez que eles estabelecem a ordem pela qual os nós computacionais realizam os intercâmbios de informações relativas ao processo. Via de regra, eles determinam o tempo máximo de vida do direito de utilização do meio de transmissão por cada um dos nós da rede e por isso representam um dos fatores que têm influência direta no tempo de resposta do sistema.

Os métodos de controle de acesso ao meio empregados em redes de campo geralmente classificam-se em dois grupos: protocolos de acesso controlado e protocolos de acesso não controlado [1], [33].

No primeiro grupo, o direito de acesso ao meio é determinado através de um método de controle explícito ou implícito. Neste grupo a abordagem empregada reflete a estratégia de captura do estado do ambiente adotada pelo sistema de controle (ver subseção 2.2.3). Na abordagem de disparo por tempo é utilizado o controle **autônomo**, ou seja, todas as atividades de intercâmbio de dados são controladas pelo próprio suporte de comunicação. Os sistemas orientados a disparo por evento dependem de controle **externo**, pois qualquer atividade de comunicação é realizada em resposta a sinais gerados no nível de aplicação, e portanto fora da esfera de controle do sistema de comunicação.

No segundo grupo classificam-se os protocolos que admitem a possibilidade de ocorrência de colisões e por isso adotam mecanismos especiais para resolução de conflitos e ordenação do acesso ao meio.

O trabalho de Almeida [1] apresenta uma taxonomia dos principais protocolos de acesso ao meio empregados em redes de campo (Fig. 3.1). Dentre eles destaca-se o modelo **produtor-distribuidor-consumidor - PDC**, que é o MAC adotado pelas redes padrão WorldFIP [43] e IEC Fieldbus [18].

O modelo PDC resulta da combinação do esquema **mestre-escravo** com o modelo **produtor-consumidor**. Nesta abordagem o acesso ao barramento é controlado de forma centralizada por uma tarefa residente em um nó que atua como mestre e desempenha o papel de **distribuidor de dados**. Esta tarefa, denominada **árbitro do barramento (AB)**, em vez de conceder o direito de acesso diretamente ao nó produtor do dado como ocorre no modelo mestre-escravo convencional, faz referência a uma entidade de dado e concede o direito de transmissão ao seu produtor, qualquer que seja o nó. Para isso as entidades de dado devem ter identificadores únicos e globais. Assim todos os nós que dependem do dado produzido, denominados **consumidores**, efetuam a leitura do valor transmitido através da

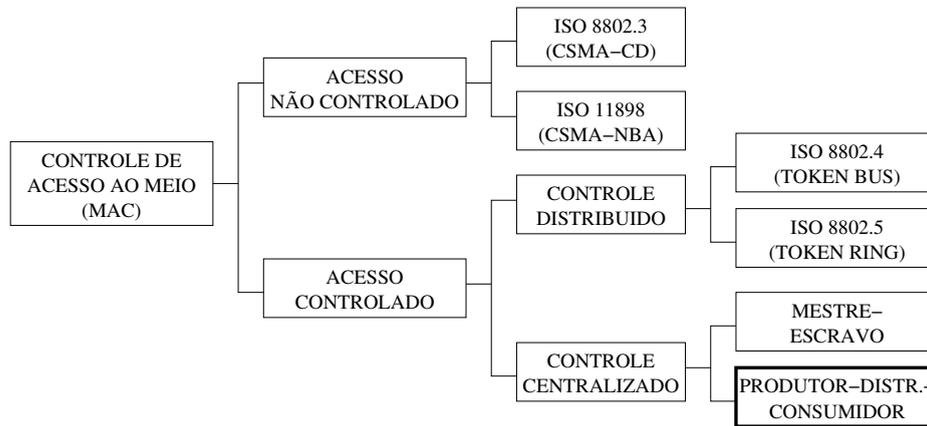


Figura 3.1: Protocolos de Acesso ao Meio (MAC) Adotados em Redes de Campo

rede de forma simultânea. Assim é garantida, de uma maneira relativamente simples, a consistência espacial do dado para todo o sistema.

### 3.1.2 Arquitetura de Redes de Campo

O modelo de referência OSI [22] [32] define uma arquitetura de sete camadas para a implementação dos protocolos e serviços providos pelos sistemas de comunicação organizados com base na interconexão de computadores heterogêneos. Em função das características específicas das redes de campo torna-se mais adequada a adoção de uma arquitetura mais simples, formada apenas por três camadas: física, enlace de dados e aplicação [33].



Figura 3.2: Arquitetura de Protocolo de Redes de Campo

Cardeira e Marmeri [10] apresentam uma análise que justifica o emprego da arquitetura simplificada em função da relevância de cada uma das camadas no atendimento dos principais requisitos exigidos para a garantia do escalonamento do tráfego em redes de campo.

### 3.1.3 Unidade de Dados de Protocolo

A camada de aplicação acrescenta informações de controle aos dados enviados pelas tarefas usuárias dos serviços de comunicação. A estrutura de dados resultante, denominada **Unidade de**

**Dados de Protocolo** (*protocol data unit - PDU*), contendo os dados e as informações sobre seus atributos, é encaminhada à camada de enlace e em seguida à camada física para então ter acesso ao meio de transmissão.

### 3.1.4 Redes de Campo Padronizadas

As vantagens técnicas e econômicas obtidas com o emprego de redes de campo em sistemas de manufatura e em sistemas de controle de processos tiveram como consequência o aparecimento de um número elevado de soluções proprietárias, baseadas em produtos que, por serem incompatíveis, representavam um óbice à integração de sistemas e componentes desenvolvidos por diferentes fornecedores. A demanda por exigência de interoperabilidade entre sistemas e componentes produzidos por diferentes desenvolvedores, manifestada por usuários e integradores, resultou em um esforço de padronização, tanto em escala regional como a nível internacional [41].

No âmbito da IEC surgiram propostas de arquiteturas tais como as redes MIL 1553B [16], HART [36] e BITBUS [19]. A norma IEC 61158, contendo a especificação de requisitos de um barramento para uso industrial, pode ser citada como um exemplo de padronização de redes de campo.

A norma EN 50170, editada pelo CENELEC (European Committee for Electrotechnical Standardization), contém as especificações das redes P-NET, PROFIBUS e WorldFIP. Almeida [1] destaca que, apesar destas alternativas não serem compatíveis entre si, a norma EN 50170 teve o mérito de reduzir o número de ofertas de produtos, no âmbito da Europa, de diversas dezenas a apenas três.

A rede CAN (*Controller Area Network*), originalmente desenvolvida para uso em veículos automotores, foi padronizada pela ISO [20] e baseia-se no método de acesso CSMA/NBA (*carrier sense multiple access / non-destructive bitwise arbitration*). Esta rede adota um mecanismo não destrutivo de resolução de colisões. Devido às suas características excepcionais, a rede CAN tem sido considerada também para aplicações em redes industriais de controle de processo distribuído [1]. Para estas aplicações, existem versões derivadas da rede CAN original tais como CANopen, padronizada pela norma EN 50325-4, e DeviceNet, empregada para interligação de sensores e atuadores.

### 3.2 A Rede WorldFIP

WorldFIP [43] é um padrão de rede de campo desenvolvido para prover a ligação entre o nível 0 (sensores e atuadores) e o nível 1 (CLPs, controladores, etc.) em sistemas de controle e automação de processos. O volume 3 da norma européia EN 50170 estabelece as características do protocolo WorldFIP, cuja natureza dos serviços disponibilizados o qualificam para aplicação a sistemas distribuídos que contêm tarefas de tempo real críticas, destacando-se o suporte para sistemas redundantes, a garantia de tempo de resposta (determinismo), taxas de transmissão elevadas (considerando-se as redes industriais) e alta confiabilidade [42].

O padrão WorldFIP suporta os métodos de controle de acesso ao meio geralmente adotados para redes industriais, tais como: controle centralizado, descentralizado ou mestre-escravo, possibilitando distribuir inteligência, sinais de controle e dados. As tarefas computacionais do sistema poderão residir em uma única unidade de processamento ou ser completamente distribuídas (Fig. 3.3). O mecanismo de difusão de dados (*broadcasting*) do protocolo WorldFIP atende aos requisitos associados aos conceitos de consistência temporal e espacial de dados apresentados na subseção 2.2.3.

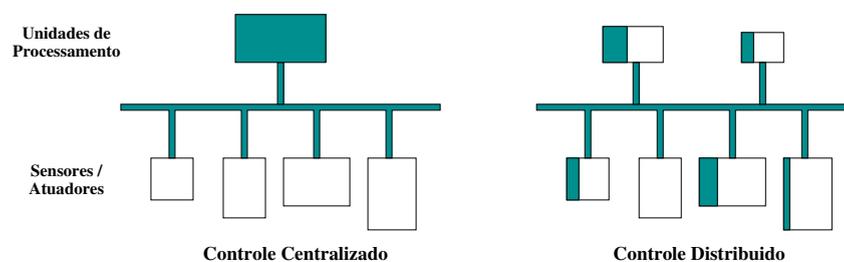


Figura 3.3: Arquiteturas de Sistema - Centralizado / Distribuído

O protocolo WorldFIP é estruturado em três camadas de comunicação do modelo de referência OSI (Fig. 3.4). Cada uma destas camadas está especificada na norma EN 50170.

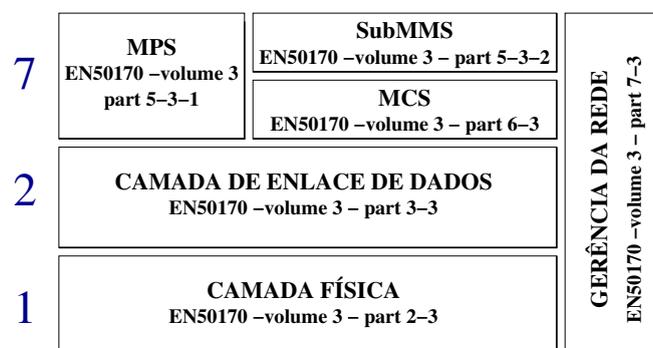


Figura 3.4: Camadas de Comunicação do Protocolo WorldFIP

### 3.2.1 Camada Física

A camada física da rede WorldFIP pode ser implementada em par trançado blindado de cobre ou em fibra ótica, em conformidade com os requisitos especificados no volume 2, parte 2-3, da norma EN 50170.

O padrão WorldFIP define três taxas de transmissão que podem ser adotadas em redes implementadas com cabos de par trançado de cobre: S1 (31,25 kb/s), S2 (1,0 Mb/s) e S3 (2,5 Mb/s). A taxa S2 (1 Mb/s) corresponde à banda padrão (*default*) enquanto as demais são utilizadas em aplicações especiais. A utilização de fibra ótica como meio de transmissão permite o emprego da rede WorldFIP em sistemas com taxa de transmissão de até 5 Mb/s.

A rede WorldFIP emprega o método Manchester para codificação dos sinais transmitidos pelo meio físico. O intervalo de tempo correspondente à transmissão de um bit, definido como  $t_{mac}$ , é dividido em duas partes de igual duração, o que possibilita a sincronização temporal das estações entre si usando os próprios sinais que trafegam pela rede. A Fig. 3.5 ilustra a codificação física de cada um dos símbolos definidos pelo protocolo.

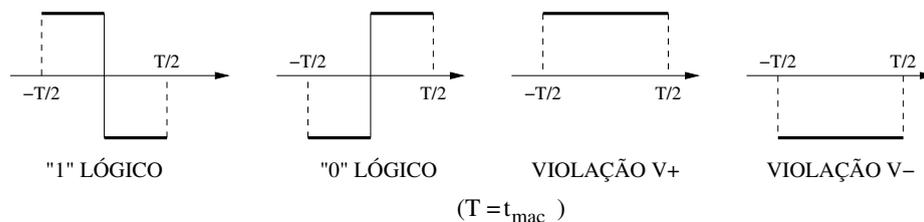


Figura 3.5: Codificação dos Símbolos na Camada Física

Os dados que trafegam através de uma rede WorldFIP são organizados em estruturas denominadas **quadros** (ver subseções 3.2.5 e 3.2.6). Os quadros são formados por três partes: uma seqüência de início de quadro - **FSS** (*frame start sequence*), campos de dados e de verificação - **CAD** (*data and check fields*) e uma seqüência de término de quadro - **FED** (*frame end sequence*) (Fig. 3.6).

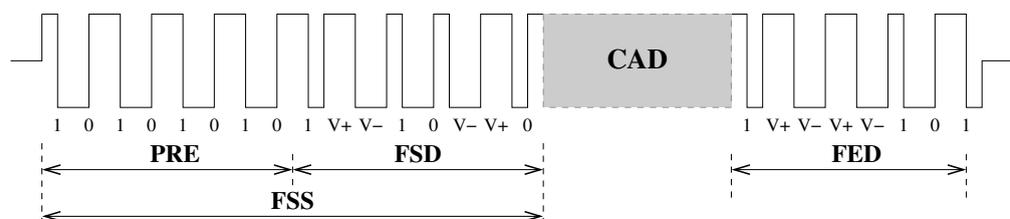


Figura 3.6: Codificação de um Quadro WorldFIP

A seqüência de início de quadro (FSS) contém os campos **preâmbulo** (PRE) e **delimitador de início de quadro** (FSD). O preâmbulo é composto por uma série de oito bits “1” e “0” que as estações receptoras empregam para sincronizarem-se com o sinal de relógio (*clock*) do transmissor. O FSD é formado por uma série de bits e símbolos de violação (“0”, “1”, “V+” e “V-”) que indicam, para a camada de enlace, o início da transmissão da “informação útil” (formada somente de símbolos lógicos “0” e “1”) contida no campo de controle e dados (CAD).

O delimitador de final de quadro (FED) é formado por uma série de bits e símbolos de violação que demarcam o fim do CAD. No total, a camada física adiciona 24 símbolos a qualquer quadro a ser transmitido pela rede.

### 3.2.2 Camada de Enlace de Dados

O protocolo WorldFIP provê dois tipos de serviços de comunicação, disponibilizados pela camada de enlace de dados: **transações de variáveis identificadas** e **transferências de mensagens**. As transações de variáveis identificadas são empregadas para suporte do intercâmbio de dados entre as tarefas de tempo real críticas. As transferências de mensagens suportam um subconjunto dos serviços do padrão MMS (*Manufacturing Messaging Specification*) e são utilizadas para prover suporte às atividades de configuração e de manutenção da rede.

Em cada estação de uma rede WorldFIP a camada de enlace contém um conjunto de *buffers* onde são armazenados os valores correntes das variáveis. Os *buffers* são alocados durante a fase de configuração do sistema e, em função do tipo de serviço a que se destinam, classificam-se como *buffers* de produção ou *buffers* de consumo de dados.

Os *buffers* são disponibilizados para as tarefas locais através dos serviços da camada de aplicação. Os conteúdos dos *buffers* de consumo podem também ser atualizados através da rede por intermédio do serviço denominado **transferência de buffer** (Fig. 3.7).

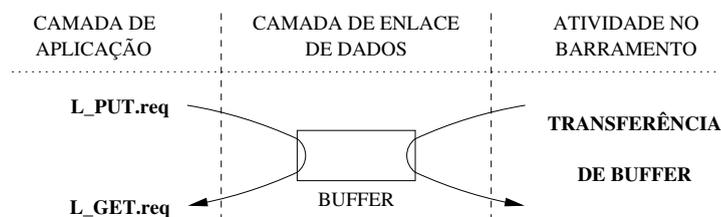


Figura 3.7: Transferência de Buffer

Os *buffers* têm largura máxima de 128 bytes e destinam-se exclusivamente ao armazenamento de variáveis identificadas; ou seja, eles não participam das atividades de transações de mensagens. A camada de enlace contém também outros recursos de comunicação que são disponibilizados para a camada de aplicação (ver subseção 3.2.3).

O modelo de endereçamento WorldFIP é formado por dois espaços distintos:

**Endereçamento de variáveis:** Cada variável da aplicação distribuída é associada a um identificador global, codificado em 16 bits, que a caracteriza. Para cada identificador há um único produtor, porém poderão existir diversos consumidores da variável produzida.

**Endereçamento de mensagens:** As trocas de mensagens podem ser realizadas de maneira ponto-a-ponto ou multi-ponto dentro de um segmento da rede. Cada mensagem transmitida contém o endereço da entidade transmissora e o endereço da entidade de destino. Esses endereços são codificados em palavras de 24 bits que contém a identificação do segmento da rede e o endereço atribuído à estação no segmento.

O controle de acesso ao meio (MAC) do padrão WorldFIP é baseado no modelo produtor-distribuidor-consumidor (PDC) (ver subseção 3.1.1). Neste modelo o árbitro do barramento inicia todas as transações através da difusão de um quadro contendo um determinado identificador. A estação que produz a variável associada ao identificador (**produtor**) responde à requisição transmitindo um quadro com o valor solicitado. Então as estações cujas tarefas dependem da variável produzida (**consumidores**) copiam, em *buffers* locais, o dado contido no quadro transmitido pelo produtor.

As informações transacionadas entre a camada de enlace e a camada física são empacotadas entre os campos **controle** (1 byte) e **FCS** (*frame control sequence*) (2 bytes) (Fig. 3.8). O campo de controle indica o tipo de quadro envolvido na transação, enquanto o FCS está associado ao mecanismo de detecção de erros, sendo o resultado de uma operação polinomial realizada sobre os dados contidos no campo da informação. A camada de enlace acrescenta **três bytes** à informação a ser transmitida.

### 3.2.3 Camada de Aplicação

Os serviços de nível mais elevado de uma rede WorldFIP são disponibilizados para as tarefas usuárias através da camada de aplicação. Estes serviços dividem-se em três grupos distintos:

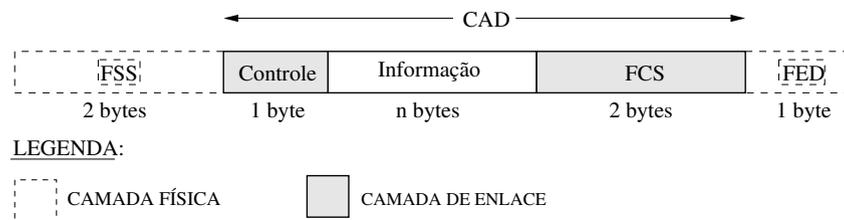


Figura 3.8: Formato do Quadro na Camada de Enlace

- Serviços de aplicação do árbitro do barramento (ABAS)
- Serviços periódicos / aperiódicos de manufatura (MPS)
- Subconjunto dos serviços de mensagens (subMMS)

O controle do tráfego na rede WorldFIP é realizado pelo árbitro do barramento, de forma centralizada. Todas as requisições de leitura ou de escrita de variáveis identificadas disparadas pelas aplicações, tanto locais como remotas, são atendidas através de chamadas de primitivas de serviço cujo mecanismo de operação encontra-se detalhado na subseção 3.2.4.

Dentre as funcionalidades disponibilizadas pela camada de aplicação, merecem destaque os serviços de suporte ao tráfego de variáveis periódicas e aperiódicas do processo (MPS). A subcamada de aplicação MPS utiliza os serviços da camada de enlace para prover os seguintes serviços às tarefas usuárias:

- Serviços de leitura e escrita locais;
- Serviços de leitura e escrita remotas;
- Notificações de transmissão e de recepção de variáveis;
- Informações de atualização da informação consumida;
- Informações das consistências espacial e temporal dos dados.

Os dados enviados pelas tarefas no nível de usuários e as informações de controle acrescentadas pela camada de aplicação de uma rede WorldFIP formam a PDU que, em função do tipo de transação, é limitada a **128 bytes** para o caso de variáveis identificadas, ou **256 bytes** para as transações de mensagens.

As variáveis identificadas são codificadas segundo a sintaxe de transferência ASN.1 [21]. As variáveis podem ser simples (booleano, inteiro, cadeia de bits, cadeia de bytes, número em ponto

flutuante de 4 ou de 8 bytes de acordo com a norma ANSI/IEEE 754, etc.), ou estruturadas (estruturas de tipos simples, tabelas, tabelas de tabelas, listas, etc.).

Usualmente as variáveis são transportadas em PDU do tipo compacto. A Fig. 3.9 ilustra o exemplo de uma PDU referente a um inteiro de 16 bits.

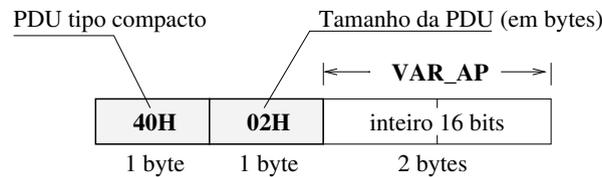


Figura 3.9: Um Exemplo de PDU do Tipo Compacto

A camada de aplicação acrescenta **dois bytes** ao dado a ser transacionado com a camada de enlace: o primeiro indica o tipo de PDU e o segundo informa o tamanho da estrutura da informação correspondente à variável identificada ( $VAR\_AP \leq 126$  bytes).

### 3.2.4 Interfaces entre as Camadas

Os grupos de serviços disponibilizados pelas três camadas do protocolo WorldFIP estão representados no esquema mostrado na Fig. 3.10.

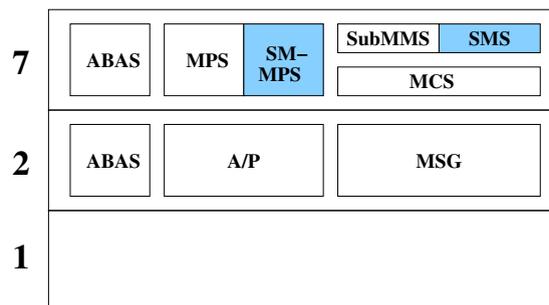


Figura 3.10: Serviços Disponibilizados pelas Camadas de Aplicação e de Enlace

As requisições de transferências de *buffers* periódicas e aperiódicas (MPS e SM-MPS) são suportadas, na camada de enlace, pelo conjunto de serviços A/P. Os serviços de gerenciamento da rede baseados em mensagens (SubMMS e SMS) são atendidos pela camada de enlace pelos serviços MSG.

A Fig. 3.11 ilustra o mecanismo de operação das interfaces entre as camadas de aplicação e de enlace de dados para atender a requisições de leitura e de escrita locais de variáveis identificadas feitas pelas aplicações dos usuários. Para leitura da variável *A*, que tem valor corrente igual a 22, a aplicação emprega a primitiva de serviço `A_READLOC.req(var_A)`. A camada de aplicação então

usa a primitiva  $L\_GET.req(ID\_A)$  para requisitar a leitura do valor corrente armazenado no *buffer* de consumo correspondente. A camada de enlace, através da primitiva  $L\_GET.cnf(22,+)$ , envia o resultado da leitura para a camada de aplicação que então o disponibiliza para o usuário através da primitiva  $A\_READLOC.cnf(22,+)$ .

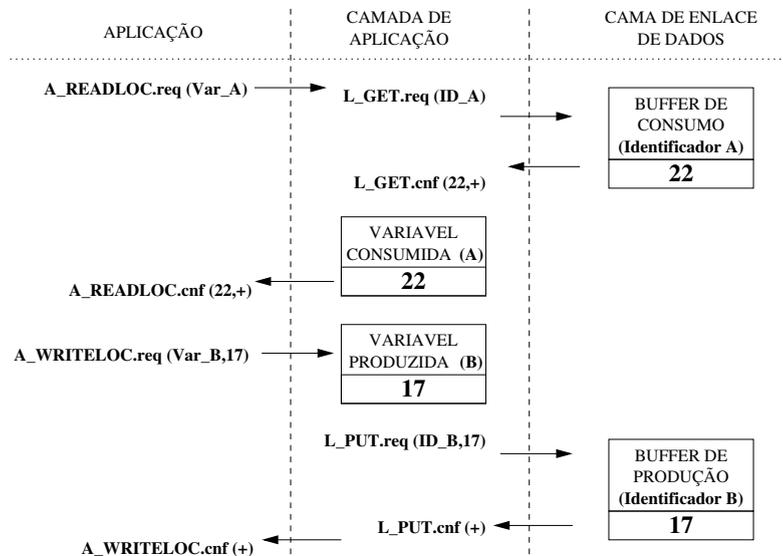


Figura 3.11: Operações de Leitura e de Escrita Locais

A escrita local de variáveis identificadas é efetuada de forma similar à operação de leitura local. Para atualizar uma variável  $B$  cujo valor corrente seja 17, por exemplo, a aplicação invoca o serviço a partir da primitiva  $A\_WRITELOC.req(var\_B,17)$  para que a camada de aplicação requisi-te o serviço da camada de enlace através da primitiva  $L\_PUT.req(ID\_B,17)$ . A camada de enlace efetua a escrita no *buffer* referente ao identificador  $ID\_B$  e confirma a execução da operação através da primitiva  $L\_PUT.cnf(+)$ . Então, a camada de aplicação informa o resultado da operação à aplicação utilizando a primitiva  $A\_WRITELOC.cnf(+)$ . Este mecanismo também está ilustrado na Fig. 3.11.

As operações de leitura e de escrita locais de variáveis são realizadas por transações que utilizam apenas os serviços das camadas de aplicação e de enlace de dados. Portanto elas não geram tráfego pelo barramento.

As estações podem também ter acesso aos valores correntes das variáveis identificadas usando o mecanismo de transferência de *buffers* (Fig. 3.12).

Inicialmente o árbitro do barramento faz a difusão de um quadro  $ID\_DAT$  tendo como parâmetro o identificador da variável requisitada ( $ID\_K$ ). A camada de enlace da estação produtora da variável

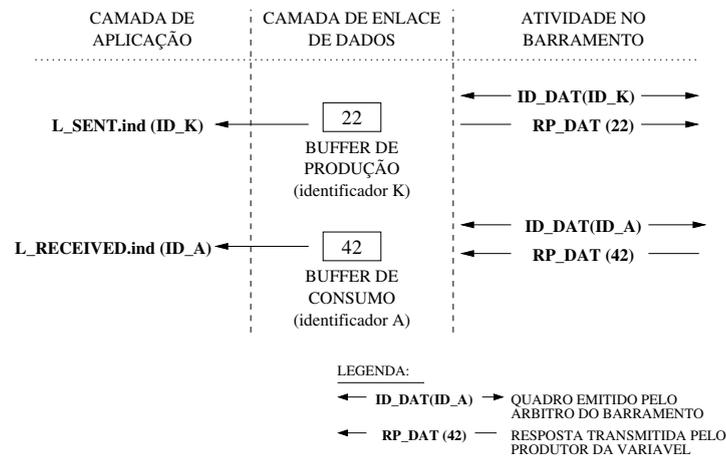


Figura 3.12: Transferência de Buffers

responde com um quadro  $RP\_DAT(valor)$  contendo o valor solicitado e então envia, para a camada de aplicação, uma indicação de transmissão do dado através da primitiva de sincronização  $L\_SENT.ind(ID\_K)$ .

Caso a estação tenha sido configurada como consumidora da variável correspondente ao identificador requisitado, por exemplo  $ID\_A$ , sua camada de enlace aceita o valor do próximo quadro  $RP\_DAT(valor)$  e então envia uma indicação de recepção da variável  $A$  para a camada de aplicação através da primitiva de sincronização  $L\_RECEIVED.ind(ID\_A)$ .

As transações de variáveis identificadas podem ser periódicas ou aperiódicas. As requisições de transações periódicas têm **prioridade mais alta** que as requisições de transações aperiódicas e são realizadas sob controle autônomo do sistema de comunicação.

As transações aperiódicas ocorrem em resposta a demandas dinâmicas utilizando os serviços da camada de aplicação e dispõem de dois níveis de prioridade: **urgente** e **normal**. Para cada nível de prioridade existe uma fila do tipo FCFS (*first come first served*).

O mecanismo de leitura remota das variáveis  $A$  e  $B$ , a partir de requisições de transmissão efetuadas por uma aplicação residente em uma das estações da rede, é ilustrado pela Fig. 3.13.

1. A aplicação efetua requisições de leitura remota das variáveis  $A$  e  $B$  através das primitivas  $A\_READFAR.req(var\_A)$  e  $A\_READFAR.req(var\_B)$ . A camada de aplicação então invoca o serviço de atualização livre usando as primitivas  $L\_FREE\_UPDATE.req(ID\_A, P)$  e  $L\_FREE\_UPDATE.req(ID\_B, P)$ , onde  $P$  é a prioridade atribuída às variáveis;

2. Os identificadores  $ID\_A$  e  $ID\_B$  são adicionados à fila local de requisições de transferências aperiódicas respeitando a prioridade atribuída às transações;
3. Em resposta à primeira requisição de transferência de variável periódica produzida pela estação, referente ao identificador  $ID\_K$  por exemplo, sua camada de enlace informa ao árbitro do barramento que tem requisições de transferências aperiódicas pendentes, através do bit  $RQ$  do campo de controle do quadro  $RP\_DAT(ID\_K)$ . O árbitro do barramento registra este identificador na fila de requisições aperiódicas pendentes residente em sua camada de enlace;
4. Então, em uma janela aperiódica (ver subseção 3.2.11), o árbitro do barramento transmite o quadro  $ID\_RQ(ID\_K)$  para solicitar à estação que envie a sua lista de requisições de transmissão de variáveis aperiódicas pendentes;
5. A estação responde com um quadro  $RP\_RQ(lista)$  e dispara a confirmação da requisição de atualização, da camada de enlace para a camada de aplicação, através da primitiva  $L\_FREE\_UPDATE.cnf(+)$ ;
6. Mais tarde, e também em uma janela aperiódica, o árbitro do barramento solicita a transmissão de cada uma das variáveis contidas na lista ( $A$ ,  $B$ ) usando os quadros  $ID\_DAT(ID\_A)$  e  $ID\_DAT(ID\_B)$ . Após o recebimento dos valores contidos nos quadros  $RP\_DAT(valor)$  emitidos pela estação produtora das variáveis requisitadas, a camada de enlace da estação envia as respectivas indicações de recepção para a camada de aplicação, que pode então usar o serviço de leitura local (ver Fig. 3.11) para acessar os valores recebidos;
7. Cada confirmação, se positiva, contém o valor da variável. Um temporizador da camada de aplicação é inicializado para detectar tempos excessivos na operação.

O mecanismo de escrita remota, invocado a partir da primitiva de serviço  $A\_WRITEFAR.req(var)$ , funciona de maneira similar ao esquema de leitura remota. A camada de aplicação primeiro atualiza o *buffer* local referente à variável produzida e então faz uma requisição de transferência. Quando a camada de aplicação receber a indicação da transmissão, que é encaminhada pela camada de enlace, o serviço será confirmado para a aplicação.

### 3.2.5 Estrutura dos Quadros $ID\_DAT$ (Pergunta)

Os quadros do tipo  $ID\_DAT$  são emitidos pelo árbitro do barramento para iniciar qualquer transação através da rede. Todos os quadros de pergunta, sejam para transações de variáveis iden-

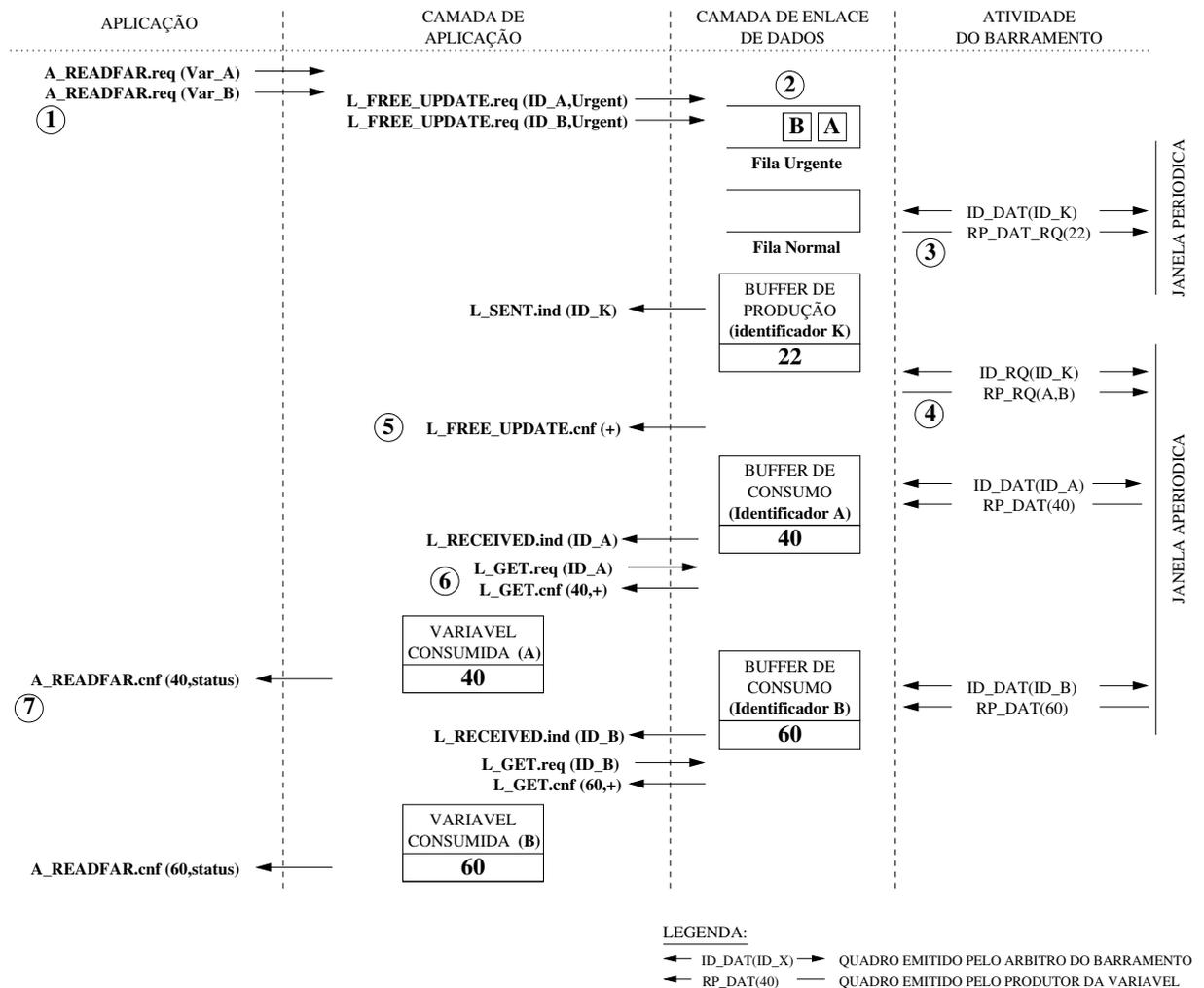


Figura 3.13: Leitura de Variável Remota - Transferência Aperiódica

tificadas (ID\_DAT, ID\_RQ) ou de mensagens (ID\_MSG), são do tipo ID\_DAT. A distinção entre eles é resolvida através do código contido no byte de controle do campo CAD (Fig. 3.14).

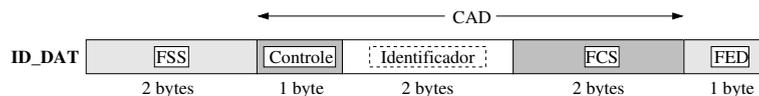


Figura 3.14: Estrutura do Quadro ID\_DAT

Os quadros ID\_DAT podem ser transmitidos de maneira autônoma pela rede, como é o caso das transações de variáveis periódicas, ou por solicitação explícita do usuário, através dos serviços da camada de aplicação (ver seção 3.2.4). Portanto os quadros ID\_DAT implementam as duas abordagens de captura do estado do ambiente descritas na subseção 2.2.3: disparo por tempo e disparo por evento.

Cada quadro ID\_DAT transporta um objeto de 16 bits correspondente ao identificador de uma

variável, empacotado com as informações de controle adicionadas pela camada de enlace e pela camada física. Todos os quadros ID\_DAT têm uma largura fixa, igual a **64 bits**.

### 3.2.6 Estrutura dos Quadros Quadros RP\_DAT (Resposta)

Os quadros transmitidos por uma rede WorldFIP em resposta aos quadros ID\_DAT transportam as PDUs recebidas da camada de aplicação encapsuladas com as informações de controle acrescentadas pelas camadas de enlace e física. Especificamente, os dados enviados da camada de enlace para a camada física, contidos no campo CAD, são colocados entre uma seqüência FSS e um delimitador FED (Fig. 3.6).

A largura do quadro RP\_DAT, cuja estrutura é ilustrada pela Fig. 3.15, corresponde a **48 bits** mais os bits correspondentes à PDU recebida da camada de aplicação.

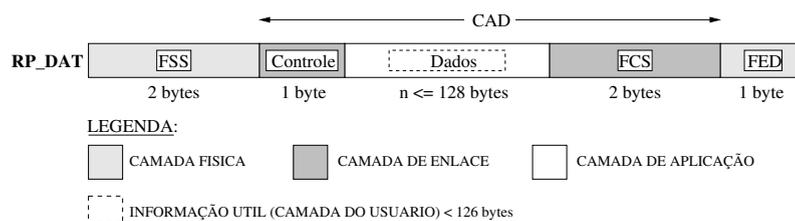


Figura 3.15: Estrutura do Quadro RP\_DAT

As respostas das estações a requisições de transmissão de listas de variáveis, efetuadas através de quadros ID\_RQ, são codificados em quadros RP\_RQ, que podem transportar até 64 identificadores de 16 bits (Fig. 3.16).

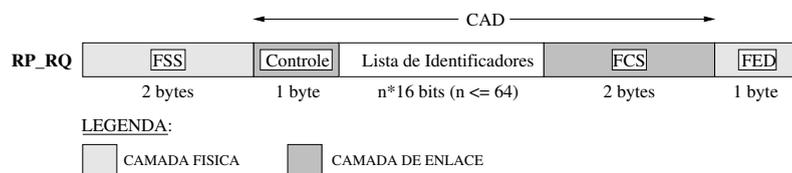


Figura 3.16: Estrutura do Quadro RP\_RQ

O protocolo WorldFIP dispõe ainda de quadros específicos para respostas a requisições de transmissão de mensagens, denominados RP\_MSG, RP\_ACK e RP\_FIN.

### 3.2.7 Durações das Transferências de Buffers

O esquema de controle do tráfego descrito na subseção 3.2.4 (Fig. 3.12) mostra que cada transferência de *buffer* é efetuada a partir da difusão, pelo árbitro do barramento, de um quadro

ID\_DAT(*ID\_X*), seguido de um quadro RP\_DAT(*valor*) enviado pela estação produtora da variável requisitada. A duração de uma transação corresponde, portanto, ao tempo de tráfego dos quadros ID\_DAT e RP\_DAT, acrescido do tempo de trânsito dos sinais pelo meio físico, denominado **tempo de circulação** ( $t_r$ ) (*turnaround time*) (Fig. 3.17). O padrão WorldFIP estabelece uma faixa de 10 a  $70 t_{mac}$  para o tempo de circulação, onde  $t_{mac}$  corresponde ao tempo de transmissão de um símbolo pela camada física (ver Fig. 3.5).

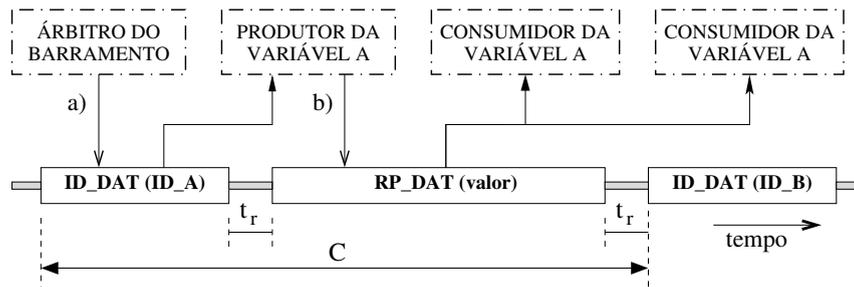


Figura 3.17: Duração da Transferência de *Buffers*

A duração da transação da variável *A*, por exemplo, pode ser calculada usando-se a equação (3.1) na qual  $tx\_rate$  representa a taxa de transmissão da rede em bit/s,  $len()$  corresponde à largura do quadro (em número de bits) e  $t_r$  é o tempo de circulação medido em segundos [1].

$$C = \frac{len(ID\_DAT) + len(RP\_DAT)}{tx\_rate} + 2.t_r \quad (3.1)$$

### 3.2.8 Controle do Tráfego Periódico

No que concerne à sincronização entre os ciclos elementares e macrociclos, a rede WorldFIP tem diversos modos de operação, em função da configuração adotada para o árbitro do barramento. Neste trabalho está sendo considerado que a rede opera no **modo síncrono**, caracterizado pela duração constante dos ciclos elementares.

O árbitro do barramento controla o tráfego periódico na rede WorldFIP a partir de uma tabela estática (TAB) que adota o modelo de escalonamento executivo cíclico (ver subseção 2.5.2). Esta tabela contém, em suas entradas, o identificador e o período de cada uma das variáveis que são varridas sequencialmente em tempo de execução. A escalonabilidade das variáveis periódicas do sistema é automaticamente verificada durante a montagem da grade de tempo que define os instantes de liberação de cada variável para transmissão (ver subseção 2.5.2). Portanto torna-se necessário, também, que seja conhecida a duração de cada transação (subseção 3.2.7).

O padrão de tráfego resultante da natureza periódica das transações é definido pelo ciclo elementar e pelo macro-ciclo do sistema, que são calculados com base nos períodos das variáveis. Para isto poderão ser empregadas as equações (2.6) e (2.7).

A grade de tempo representada pela Fig. 3.18 corresponde à TAB de um sistema cujas variáveis ( $A$ ,  $B$ ,  $C$ ,  $D$  e  $E$ ) têm períodos iguais a 1, 2, 3, 4 e  $4 \times e$ , respectivamente, onde  $e$  quantifica a largura do ciclo elementar. O macro-ciclo deste sistema tem duração  $m = 12e$ .

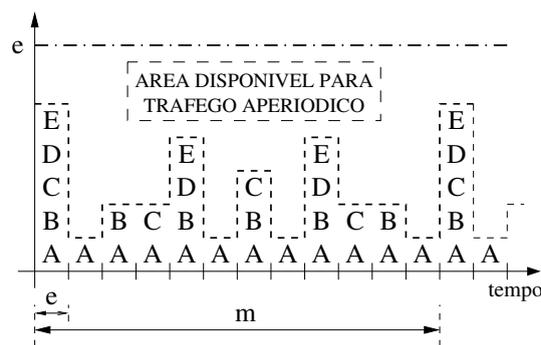


Figura 3.18: Grade de Tempo Referente ao Tráfego de Variáveis Periódicas

Deve ainda ser mencionado que, conforme mostrado no gráfico ilustrado pela Fig. 3.18, a área situada entre a demanda de tráfego periódico e a linha horizontal que define o limite do tempo disponível para o tráfego de dados em cada ciclo elementar ( $e$ ) pode ser utilizada para atendimento do tráfego aperiódico (subseção 3.2.11).

### 3.2.9 Montagem da TAB Usando a Abordagem Taxa Monotônica

A montagem da tabela do árbitro do barramento usando o algoritmo RM (ver subseção 2.4.3) baseia-se nos seguintes passos: 1) as requisições iniciais de transmissão de todas as variáveis periódicas são ativadas no início do macro-ciclo (instante crítico); 2) em cada ciclo elementar é verificado se a demanda na rede excede a duração ( $e$ ); 3) as variáveis são escalonadas na ordem inversa de seus períodos; 4) se a demanda da rede (*workload*) for maior que o parâmetro  $e$  então a variável será escalonada para o próximo ciclo elementar, obedecendo à sua prioridade; 5) se for excedido o período de qualquer das variáveis então o conjunto das variáveis não será escalonável. Neste caso os parâmetros das variáveis deverão ser reajustados [41].

Os dados mostrados na Tab. 3.1 representam um sistema distribuído no qual seis variáveis periódicas com largura de 126 bytes são transacionadas por uma rede WorldFIP com banda de

2,5 Mb/s e  $t_r = 8 \times 10^{-6} s$ . Na tabela também estão discriminados os custos de transmissão das variáveis, calculados através da equação (3.1).

Variável	Período (ms)	Custo ( $\mu s$ )
$v_{p1}$	4	470,4
$v_{p2}$	8	470,4
$v_{p3}$	8	470,4
$v_{p4}$	12	470,4
$v_{p5}$	12	470,4
$v_{p6}$	12	470,4

Tabela 3.1: Parâmetros das Variáveis Periódicas de um Sistema Distribuído

A Fig. 3.19 mostra a grade de tempo referente à escala do tráfego das variáveis descritas pelos atributos relacionados na Tab. 3.1. A análise desta grade permite ainda que sejam feitas as seguintes observações: 1) o ciclo elementar e o macro-ciclo têm durações de 4 ms e de 24 ms, respectivamente; 2) somente as variáveis  $v_{p1}$ ,  $v_{p2}$  e  $v_{p3}$  ocupam sempre a mesma fatia (slot) de tempo dentro de cada ciclo elementar. As liberações de todas as outras variáveis ocorrem em diferentes fatias de tempo dentro de cada CE, evidenciando a existência de *jitter* de transmissão (subseção 2.6.3).

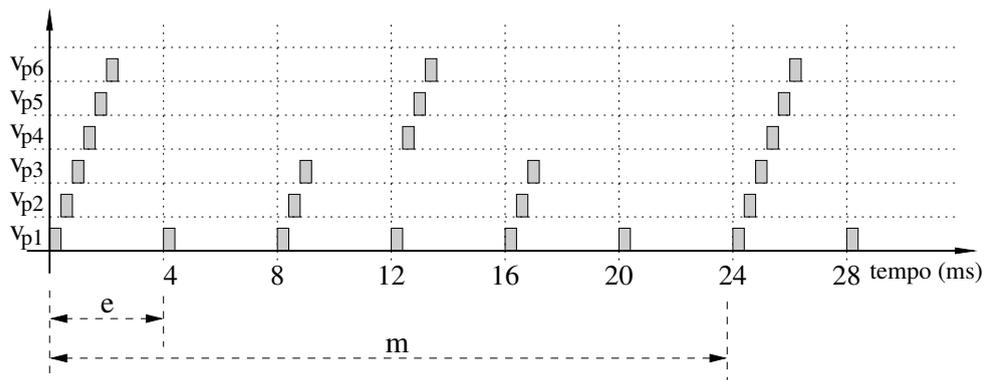


Figura 3.19: Tráfego de Variáveis Periódicas - Grade de Tempo da Abordagem RM

### 3.2.10 Montagem da TAB Usando a Abordagem *Earliest Deadline First* (EDF)

Na abordagem EDF (ver subseção 2.4.3) terá preferência de acesso ao meio físico a variável cujo prazo de transmissão estiver mais próximo de se expirar. Se duas ou mais variáveis tiverem o mesmo prazo elas serão escalonadas segundo a ordem de colocação da fila correspondente à sua prioridade (urgente ou normal), obedecendo o critério FCFS.

Os passos 1) e 2) da montagem da TAB empregando a abordagem RM (subseção 3.2.9) também serão adotados para a alternativa de abordagem de escalonamento EDF. O passo 3), entretanto, será diferente: a próxima variável a ser escalonada para transmissão será aquela cujo tempo restante até o início do ciclo elementar no qual ocorrerá a próxima ativação (ou tempo de vida) for menor. Os passos 4) e 5) também são análogos à abordagem de escalonamento RM.

O mecanismo de atribuição dinâmica de prioridades adotado pelo algoritmo EDF apresenta como principal vantagem em relação ao critério de estabelecimento de prioridades estáticas associado ao algoritmo RM o fato de ser capaz de escalonar certos conjuntos de variáveis que não são escalonáveis pelo algoritmo taxa monotônica.

Por exemplo, se a banda de transmissão da rede do sistema representado pelas variáveis listadas na Tab. 3.1 for modificada para 1 Mb/s, mantendo-se as mesmas periodicidades, a duração da transmissão de cada variável, calculada pela equação (3.1), será de  $1.176 \mu\text{s}$ . Neste caso o conjunto não será escalonável pelo algoritmo RM, pois  $v_{p6}$  “perde” o prazo no terceiro ciclo elementar (Fig. 3.20).

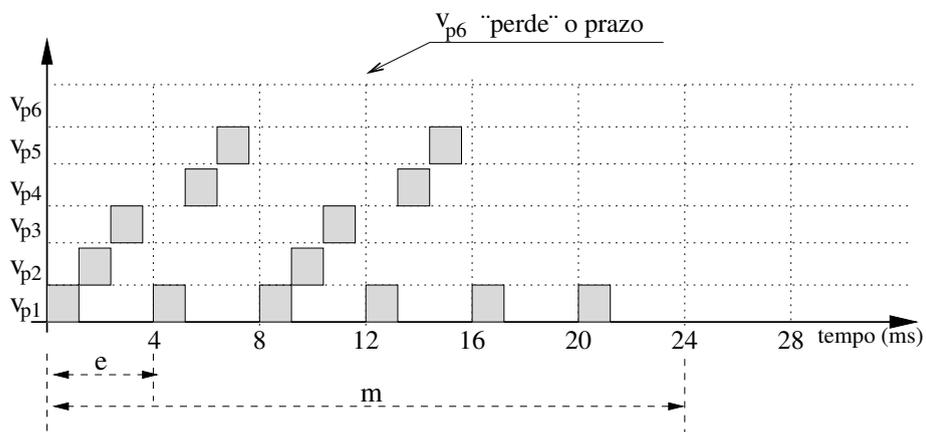


Figura 3.20: Tráfego de Variáveis Periódicas - Abordagem RM - Sistema não Escalonável

Entretanto, o mesmo conjunto de variáveis é escalonável pelo algoritmo EDF, conforme ilustra a grade de tempo mostrada na Fig. 3.21.

Pode ser observado, por outro lado, que existe uma maior variação entre os instantes de liberação das variáveis dentro de cada ciclo elementar, relativamente à escala produzida pelo algoritmo RM. Os instantes de transmissão das variáveis  $v_{p2}$  e  $v_{p3}$ , por exemplo, não apresentam a mesma regularidade se comparados com o padrão de liberações mostrado na Fig. 3.19, evidenciando a tendência de aumento do nível do “jitter” associada à abordagem EDF [41].

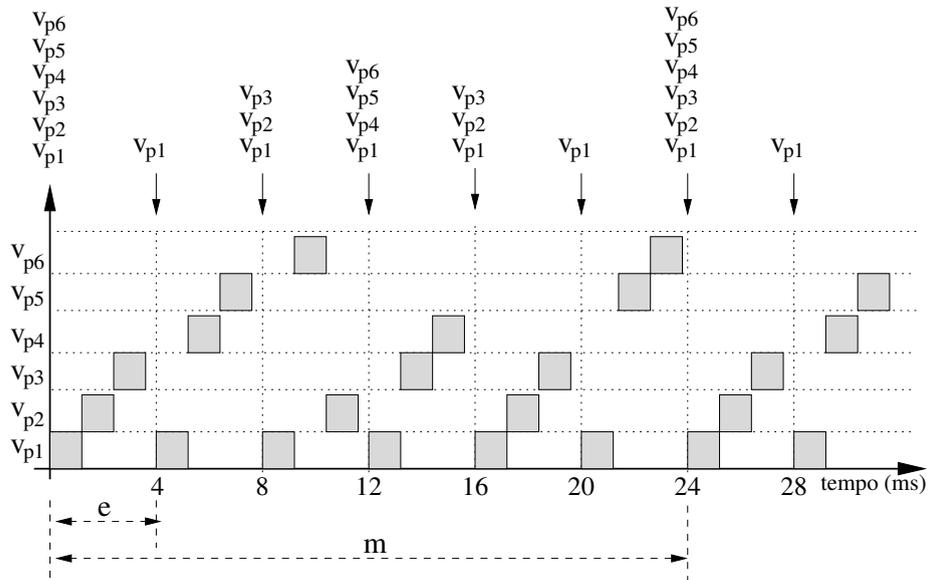


Figura 3.21: Tráfego de Variáveis Periódicas - Grade de Tempo da Abordagem EDF

### 3.2.11 Controle do Tráfego Aperiódico

O ciclo elementar típico de uma rede de campo compõe-se de uma **fase síncrona** e de uma **fase assíncrona** [33]. No ciclo elementar de duração  $e$ , representado pela Fig. 3.22, a fase síncrona é reservada para o processamento do tráfego das variáveis periódicas e portanto denomina-se **janela periódica** ( $PW$ ). A fase assíncrona, denominada **janela aperiódica** ( $AW$ ), acomoda as subjanelas de tempo que podem ser alocadas para o tráfego das variáveis aperiódicas com prioridades urgente e normal ( $UAW, NAW$ ) e uma subjanela na qual pode ser realizada, se necessário, a transmissão de mensagens ( $MW$ ) [1].

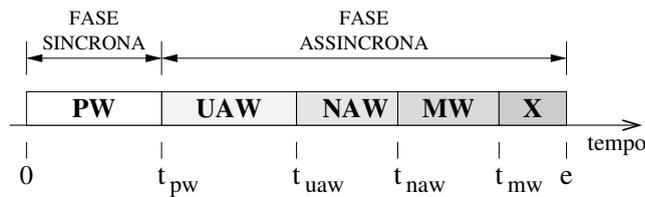


Figura 3.22: Janelas de Tempo do Ciclo Elementar

A janela  $PW$  e as subjanelas  $UAW, NAW$  e  $MW$  têm prioridades decrescentes, ou seja, uma vez concluídas as transações periódicas, as transmissões das variáveis aperiódicas ocorrem nas subjanelas subsequentes, se o tempo restante no ciclo elementar corrente o for suficiente. Portanto o escalonamento do tráfego em um ciclo elementar deverá obedecer à seguinte condição:

$$t_{pw} \leq t_{uaw} \leq t_{naw} \leq t_{mw} \leq e \tag{3.2}$$

No caso de  $t_{mw} < e$  uma subjanela de sincronização ( $X$ ) será criada no fim do ciclo elementar. Esta subjanela não será utilizada para transmissão de informações associadas às variáveis do sistema. Em vez disso a camada de enlace da rede a completará com um padrão de dados para preenchimento do tempo remanescente (*padding identifiers*), empregando a técnica de inserção de tempo ocioso (ver subseção 2.4.2).

O árbitro do barramento controla o tráfego aperiódico através do gerenciamento das filas de requisições residentes em sua camada de enlace, referentes às transferências pendentes (urgente e normal) e às requisições em andamento (Fig. 3.23).

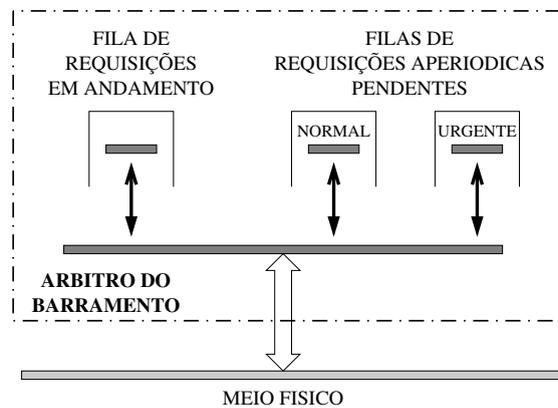


Figura 3.23: Filas de Requisições Aperiódicas no Árbitro do Barramento

De acordo com o mecanismo detalhado na subseção 3.2.4 (Fig. 3.13) a transferência de *buffers* entre estações produtoras e consumidoras de variáveis identificadas envolve a realização de uma série de transações atômicas consistindo de pares de quadros ID\_RQ / RP\_RQ referentes às listas de variáveis pendentes seguidos de pares de quadros ID\_DAT / RP\_DAT correspondentes a cada variável requisitada.

Este mecanismo, que ocorre na janela aperiódica de um ou de vários ciclos elementares consecutivos, encontra-se logicamente detalhado através do Algoritmo 1.

O cálculo do tempo de resposta de pior caso do tráfego aperiódico será efetuado com base nas seguintes premissas [41]:

1. Tanto a estação produtora como as estações consumidoras de uma determinada variável poderão emitir requisições de transferência aperiódica desta variável;

**Algoritmo 1** Controle do Tráfego Aperiódico pelo Árbitro do Barramento

---

```

Início da janela aperiódica
while [há tempo para processar transferência aperiódica de buffers] do
  if [“fila” de requisições em andamento está vazia] then
    if [há requisições de transferência aperiódica pendentes] then
       $ID\_RQ \Rightarrow$ 
       $RP\_RQ(lista) \Leftarrow$ 
    else
      while [há tempo para emitir identificadores de preenchimento] do
        [emite identificadores de preenchimento]
      end while
      [fim da janela aperiódica]
    end if
  else
     $ID\_DAT \Rightarrow$ 
     $RP\_DAT \Leftarrow$ 
  end if
end while

```

---

2. Cada estação poderá requisitar a transferência de uma única variável por quadro RP\_RQ transmitido. Portanto, se o sistema contiver  $na$  variáveis aperiódicas, para transmissão de cada variável será necessário o processamento de um par de quadros ID\_RQ / RP\_RQ seguido de outro par ID\_DAT / RP\_DAT, resultando em um número máximo de  $2 \times na$  requisições;
3. A transmissão com duração mais longa ( $Ca_{max}$ ), que será utilizada para calcular o limite superior do tempo de resposta, é definida a partir do seguinte conjunto de equações:

$$Ca_{RQ.i} = \max_{i=1, \dots, na} \left\{ \frac{\text{len}(ID\_RQ.x) + \text{len}(RP\_RQ.lista.i)}{tx\_rate} + 2t_r \right\} \quad (3.3)$$

$$Ca_{DAT.i} = \max_{i=1, \dots, na} \left\{ \frac{\text{len}(ID\_DAT.i) + \text{len}(RP\_DAT.valor.i)}{tx\_rate} + 2t_r \right\} \quad (3.4)$$

$$Ca_{max} = \max \{ Ca_{DAT.i}, Ca_{RQ.i} \} \quad (3.5)$$

4. Para cada variável aperiódica existe um intervalo mínimo entre requisições, que deverá ter como limite superior o tempo de resposta de pior caso da variável. Desta forma a transmissão será caracterizada como esporádica, de acordo com a definição adotada na subseção 2.6.2;
5. As requisições de transferências aperiódicas pendentes que estiverem aguardando na fila correspondente do árbitro do barramento serão processadas se a “fila” de transações em andamento estiver vazia e se houver tempo suficiente no ciclo elementar correspondente (Algoritmo 1);

6. O instante de ativação (ou de chegada) da requisição de transmissão de cada variável aperiódica no árbitro do barramento é tal que o quadro ID\_RP com o bit  $RQ$  ativado seja emitido no ciclo elementar que resulte na ocorrência do valor máximo do “*jitter*” da variável de maior frequência dentre as variáveis periódicas produzidas pela estação correspondente.

As metodologias de cálculo encontradas na literatura [1] [41] em geral decompõem o tempo de resposta da requisição de transmissão de uma variável aperiódica produzida pela estação  $k$  de uma rede WorldFIP em duas parcelas distintas: uma representa o limite superior do intervalo morto ( $\sigma^k$ ) e a outra leva em conta o tempo de ocupação do meio físico pelas variáveis aperiódicas escalonadas para transmissão antes da variável considerada (*aperiodic busy interval - ABI*).

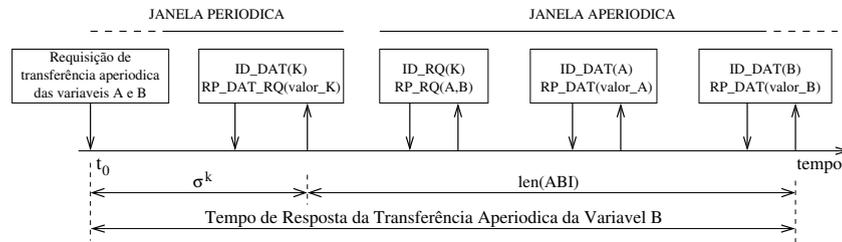


Figura 3.24: Tempo de Atendimento de Requisições Aperiódicas

A Fig. 3.24 representa graficamente o tempo de resposta ( $Ra^k$ ) da variável aperiódica  $B$ , modelado pela seguinte equação:

$$Ra^k = \sigma^k + len(ABI) \quad (3.6)$$

As parcelas que compõem a equação (3.6) têm os significados apresentados a seguir:

$\sigma^k$  - limite superior do intervalo morto, que depende do tempo de transmissão das variáveis periódicas, definido pela seguinte equação:

$$\sigma^k = Tp_j + J_{vp_j} + Cp_j \quad (3.7)$$

onde:

$Tp_j$  - período correspondente à variável  $vp_j$ , a qual tem a frequência de ativações mais elevada dentre as variáveis periódicas produzidas pela estação  $k$ ;

$J_{vp_j}$  - valor máximo do *jitter* de comunicação da variável  $vp_j$ ;

$Cp_j$  - duração de transmissão da variável periódica  $j$ . O valor de  $Cp_j$  pode ser calculado pela equação (3.1);

$len(ABI)$  - tempo de processamento de todas as requisições aperiódicas que estejam pendentes de transmissão na fila do árbitro do barramento. Este atraso, que corresponde a um ou mais ciclos elementares, é denominado pela literatura como **intervalo de ocupação aperiódica** (*aperiodic busy interval*). Tovar [41] apresenta um algoritmo, sob a forma de pseudo-código, que pode ser empregado para cálculo de  $len(ABI)$ .

Para que haja garantia de escalonabilidade das variáveis aperiódicas do sistema é necessário que estas sejam definidas como variáveis esporádicas (ver subseção 2.3.2). Ou seja, deverá ser especificado um intervalo de tempo mínimo entre cada duas requisições consecutivas de transmissão da variável aperiódica  $va_i$ , produzida pela estação  $k$ . Este tempo poderá ser calculado pela seguinte expressão:

$$Ta_i^k \geq Ra^k = \sigma^k + len(ABI) \quad (3.8)$$

### 3.2.12 Eficiência da Rede

A partir de informações contidas no documento descritivo do protocolo WorldFIP [43], pode ser deduzida uma equação matemática para o cálculo da eficiência da rede ( $\eta$ ), definida como a razão entre o tempo de transmissão da “informação útil” e o tempo total de execução da transação.

$$\eta = \frac{8 \times VAR\_AP}{8 \times VAR\_AP + 128 + 2.t_r} \quad (3.9)$$

Na equação (3.9)  $VAR\_AP$  representa a largura do dado enviado pela tarefa usuária para a camada de aplicação (em bytes) e  $t_r$  quantifica o tempo de circulação expresso com número de  $t_{mac}$ . Por exemplo uma rede com  $t_r = 20 t_{mac}$  terá um rendimento de 75% para transmissão de variáveis estruturadas como PDUs de 64 bytes, correspondendo a uma taxa de transferência líquida de *throughput* de 1875 kb/s.

## Capítulo 4

# Uma Metodologia de Cálculo do Tempo de Resposta da Rede WorldFIP

O limite superior do tempo de resposta no atendimento de requisições de transmissão de variáveis aperiódicas por uma rede WorldFIP pode ser determinado a partir da aplicação dos conceitos de sistemas de tempo real apresentados no Capítulo 2.

Com base nos trabalhos referentes à análise da escalabilidade do tráfego em redes WorldFIP encontrados na literatura [1] [41] e nos atributos, algoritmos e equações que caracterizam a aplicação do protocolo WorldFIP no contexto desta Dissertação, este Capítulo apresenta um método de cálculo do tempo de resposta de pior caso do tráfego aperiódico na rede de campo do sistema de controle de uma unidade geradora de energia elétrica que utiliza este protocolo como suporte de comunicação.

### 4.1 Principais Fatores de Influência para o Tempo de Resposta

O tempo de resposta de pior caso de atendimento de uma requisição de transferência de *buffers* referente a uma variável aperiódica emitida pela estação  $k$  de uma rede WorldFIP é função de diversos fatores, dentre os quais podem ser ressaltados:

1. Características físicas do *hardware* utilizado (banda de transmissão, tempo de circulação, meio físico adotado, componentes, etc.);
2. Frequências de ativação das variáveis cujos períodos definem as larguras do ciclo elementar e do macro-ciclo do sistema;

3. Instante de emissão da requisição de transmissão pela estação produtora ou consumidora da variável em relação ao macro-ciclo do sistema;
4. Carga da rede devido ao tráfego periódico;
5. Largura, em bytes, das variáveis periódicas e aperiódicas;
6. Posição da requisição de transmissão da variável na fila correspondente do árbitro do barramento.

## 4.2 Modelos das Variáveis

### 4.2.1 Variáveis Periódicas

Uma vez que a rede WorldFIP considerada neste trabalho adota o modo de tráfego síncrono (subseção 3.2.8), o prazo relativo fim-a-fim das variáveis periódicas será igual aos seus respectivos períodos. Portanto o modelo adotado para as variáveis periódicas ( $vp_i$ ) considera apenas três parâmetros.

$$vp_i = (k, Cp_i, T_i) \quad (4.1)$$

onde:

$k$  - identificador da estação produtora da variável;

$Cp_i$  - duração da transação, calculada pela equação (4.6);

$T_i$  - período da variável;

### 4.2.2 Variáveis Aperiódicas

A expressão (4.2) descreve o modelo matemático das variáveis aperiódicas, onde  $k$  representa o identificador da estação produtora da variável,  $Ca_i$  a duração da transação,  $Pa_i$  a prioridade e  $Da_i$  o prazo máximo de atendimento (*deadline*).

$$va_i = (k, Ca_i, Pa_i, Da_i) \quad (4.2)$$

onde:

$k$  - identificador da estação produtora da variável;

$Ca_i$  - duração da transação, calculada pela equação (4.6);

$Pa_i$  - prioridade atribuída à variável;

$Da_i$  - prazo de atendimento da variável.

### 4.3 Teste de Escalonabilidade para o Algoritmo Executivo Cíclico

A verificação da escalonabilidade de um conjunto de variáveis periódicas, efetuada durante a montagem da escala de transmissão de um sistema que adota o algoritmo executivo cíclico (subsecção 2.5.2), baseia-se nas seguintes considerações:

- Início do macro-ciclo no instante  $t_0 = 0$  (instante crítico);
- Largura do ciclo elementar ( $e$ ) calculada pela equação (2.7);
- Número de ciclos elementares contidos em um macro-ciclo ( $N$ ) calculado pela equação (2.8);
- Ciclo elementar  $CE_j$  definido no intervalo  $[a; b]$ , onde:

$$b = t_0 + j \times e \quad (j = 1, 2, \dots, N)$$

$$a = b - e$$

- Relação entre os instantes de ativação dos elementos do subconjunto  $S$  de variáveis periódicas ( $vp_i$ ) que estejam aguardando atendimento e seus respectivos prazos, tal que  $A_i \leq a < D_i$ . Além disso, os  $n$  elementos do subconjunto  $S$  têm prazo  $D_i = b$ .

Se pelo menos uma variável do subconjunto  $S$  não puder ser escalonada no ciclo elementar  $CE_j$  então, com os parâmetros correntes, o conjunto de variáveis não será escalonável.

Esta é uma condição necessária e suficiente de escalonabilidade que poderá ser expressa por:

$$\sum_{i=0}^n Cp_i \leq e \quad (4.3)$$

Em termos de utilização do recurso a expressão (4.3) poderá ser apresentada como um teste exato com o seguinte formato:

$$U = \sum_{i=0}^n \frac{Cp_i}{e} \leq 1 \quad (4.4)$$

## 4.4 Matriz de Escalonamento do Tráfego Periódico

A escala de tráfego das variáveis periódicas na rede WorldFIP pode ser representada matematicamente como uma matriz de dimensão  $i \times j$ , onde  $i$  varia de 1 até  $np$  (número de variáveis periódicas) e  $j$ , representando o número de ciclos elementares contidos em um macro-ciclo, varia de 1 até  $N$ .

Os algoritmos de cálculo do ciclo elementar e do macro-ciclo de um conjunto de variáveis periódicas estão codificados em linguagem C e disponibilizados no Apêndice B (seção B.1 - função `mdc_mmc.c`).

### 4.4.1 Matriz de Escalonamento Gerada pelo Algoritmo RM

Devido à natureza estática das prioridades atribuídas às variáveis com base no algoritmo taxa monotônica, se for adotado o critério de ordenação crescente em função dos períodos, os elementos da matriz de escalonamento do tráfego periódico poderão ser representados como valores booleanos [41]. Neste caso o elemento  $a_{ij}$  da matriz  $mtp_{RM}$  será verdadeiro (**1 lógico**) se a variável  $vp_i$  for escalonada para transmissão no ciclo elementar  $j$ ; caso contrário o valor será falso (**0 lógico**).

Como exemplo, a grade de transmissão mostrada na Fig. 3.19 (subseção 3.2.9) corresponde à seguinte **matriz de escalonamento**:

$$\mathbf{mtp}_{RM} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

A montagem da matriz de escalonamento  $mtp_{RM}$  pode ser realizada através da utilização da função `mr.m.c` cujo código fonte, em linguagem C, encontra-se no Apêndice B (seção B.2).

### 4.4.2 Matriz de Escalonamento Gerada pelo Algoritmo EDF

O mecanismo de atribuição dinâmica de prioridades com base nos prazos de transmissão adotado pelo algoritmo de escalonamento EDF proporciona o aumento da escalonabilidade de determinados conjuntos de variáveis periódicas, tomando-se como base de comparação o algoritmo RM (ver subseção 3.2.10).

Por outro lado poderá ser observado um aumento relativo do nível de “*jitter*” de transmissão causado pela alteração da ordem de colocação das variáveis na respectiva fila de requisições pendentes no árbitro do barramento pois, neste caso, devido à ordenação das requisições em cada ciclo elementar em função dos prazos correntes, poderá ocorrer a inversão das posições de algumas variáveis na fila de atendimento. Para cumprir este requisito cada elemento da matriz  $mtp_{EDF}$  será representado por um par ordenado  $(a_{ij}; b_{ij})$ , onde:

$a_{ij}$  - valor booleano que indica a transmissão da variável  $vp_i$  no ciclo elementar  $j$ .

$b_{ij}$  - número inteiro não negativo que define a prioridade de acesso ao meio, ou seja, a posição relativa de atendimento da variável  $vp_i$  no ciclo elementar  $j$ .

O valor  $(a_{ij}; b_{ij}) = (0;0)$  indica que a variável  $vp_i$  não será transmitida no ciclo elementar  $j$ .

A matriz de escalonamento mostrada a seguir corresponde ao sistema representado graficamente pela Fig. 3.21 (subsecção 3.2.10).

$$\mathbf{mtp}_{EDF} = \begin{bmatrix} (1, 1) & (1, 1) & (1, 1) & (1, 1) & (1, 1) & (1, 1) \\ (1, 2) & (0, 0) & (1, 3) & (0, 0) & (1, 2) & (0, 0) \\ (1, 3) & (0, 0) & (0, 0) & (1, 2) & (1, 3) & (0, 0) \\ (0, 0) & (1, 2) & (0, 0) & (1, 3) & (0, 0) & (0, 0) \\ (0, 0) & (1, 3) & (0, 0) & (0, 0) & (0, 0) & (1, 2) \\ (0, 0) & (0, 0) & (1, 2) & (0, 0) & (0, 0) & (1, 3) \end{bmatrix}$$

O código fonte, em linguagem C, da função `medf.c` encontra-se no Apêndice B (seção B.3). Esta função poderá ser empregada para geração da matriz  $mtp_{EDF}$ .

## 4.5 Cálculo da Duração das Transações

O protocolo WorldFIP define uma faixa de valores para o tempo de circulação ( $t_r$ ) em função do tempo de transmissão de um símbolo físico (subsecção 3.2.1). Com base nesta consideração a equação (3.1) será reescrita para a forma apresentada a seguir, que recebe como parâmetros a taxa de transmissão da rede ( $tx\_rate$ ) em Mb/s, a largura de cada um dos quadros transmitidos em número de bits e o tempo de circulação ( $t_r$ ) também em número de bits ( $t_{mac}$ ). O resultado produzido terá como

unidade o microssegundo ( $\mu s$ ).

$$C = \frac{\text{len}(ID\_DAT) + \text{len}(RP\_DAT) + 2.t_r}{tx\_rate} \quad (4.5)$$

Considerando-se que o quadro  $ID\_DAT$  tem largura fixa de 64 bits (subseção 3.2.5) e o quadro  $RP\_DAT$  acrescenta 48 bits à PDU recebida pela camada de enlace, a equação (4.5) pode ser apresentada sob uma forma mais simples:

$$C = \frac{8 \times VAR\_AP + 128 + 2.t_r}{tx\_rate} \quad (4.6)$$

Na equação (4.6)  $VAR\_AP$  (em bytes) representa a largura da estrutura de dados contida na PDU correspondente à variável identificada (Fig. 3.9 - subseção 3.2.3).

## 4.6 Determinação do Tempo de Resposta de Pior Caso do Tráfego Aperiódico

O tempo de resposta da requisição de transmissão de uma variável aperiódica depende do tráfego periódico alocado em cada ciclo elementar e das requisições de tráfego aperiódico que estiverem enfileiradas no árbitro do barramento, aguardando atendimento (subseção 3.2.11).

Considerando-se que, em uma rede WorldFIP, as variáveis periódicas têm prioridade mais elevada que as variáveis aperiódicas, o tempo que as requisições de tráfego aperiódico estiverem aguardando a liberação do meio físico de transmissão devido ao tráfego periódico de fato representa uma interferência, de acordo com a definição adotada nesta Dissertação (seção 2.5). Por esta razão o tempo de resposta de pior caso da requisição de transmissão de uma variável aperiódica produzida por uma estação  $k$  será modelado pela seguinte equação:

$$Ra^k = I_p^k + W_a^k \quad (4.7)$$

onde:

$I_p^k$  - limite superior da interferência causada pelo tráfego periódico;

$W_a^k$  - demanda na rede devida ao tráfego aperiódico.

#### 4.6.1 Cálculo da Interferência Causada pelo Tráfego Periódico ( $I_p^k$ )

Para que a requisição de transferência aperiódica seja informada ao árbitro do barramento no ciclo elementar corrente é necessário que ela esteja disponível na fila da camada de enlace da estação  $k$  antes da transmissão do quadro  $RP\_DAT(valor)$ , que assim poderá ter o bit  $RQ$  ativado. Caso contrário a requisição será transmitida na próxima ativação da variável de menor período produzida pela estação. Isto significa que, no pior caso, deverão ser considerados este período e a carga devida às variáveis periódicas escalonadas para transmissão a partir da variável  $vp_i$ , na janela periódica do ciclo elementar no qual terá início o processamento da transação aperiódica requisitada. A equação (4.8) poderá ser utilizada para o cálculo desta interferência:

$$I_p^k = T_i^k + J_{vp_i} + w_p^l \quad (4.8)$$

onde:

$T_i^k$  - período da variável  $vp_i$  que tem a freqüência de ativações mais elevada dentre as variáveis produzidas pela estação  $k$ ;

$J_{vp_i}$  - valor máximo do *Jitter* de transmissão de  $vp_i$ ;

$l$  - ciclo elementar a partir do qual serão processadas as transações aperiódicas requisitadas;

$w_p^l$  - parcela da janela periódica do ciclo elementar  $l$  utilizada para transmissão das variáveis periódicas escalonadas para transmissão a partir da variável  $vp_i$ .

O valor máximo do *jitter* de transmissão de  $vp_i$  assim como o ciclo elementar a partir do qual serão processadas as transações aperiódicas que estiverem aguardando liberação na fila do árbitro do barramento ( $l$ ) poderão ser determinados através da utilização do algoritmo codificado em linguagem C (função `j i t . c`) que encontra-se no Apêndice B (seção B.4).

##### a) Cálculo de $w_p^l$ para o Caso de Adoção do Algoritmo RM

No caso do escalonamento com base no algoritmo RM o valor de  $w_p^l$  é quantificado pela seguinte equação:

$$w_p^l = \sum_{h=i}^{np} a_{hl} \times Cp_h \quad (4.9)$$

$a_{hl}$  - elemento da matriz  $mtp_{RM}$  (subseção 4.4.1);

$Cp_h$  - duração de transmissão da variável periódica  $h$ .  $Cp_h$  poderá ser calculada pela equação (4.6);

$np$  - número de variáveis periódicas.

### b) Cálculo de $w_p^l$ para o Caso de Adoção do Algoritmo EDF

Caso seja adotado o algoritmo EDF para escalonamento das variáveis o primeiro passo a ser efetuado para o cálculo de  $w_p^l$  deverá ser a ordenação dos elementos da coluna  $l$  da matriz  $mtp_{EDF}$  de forma crescente em relação às prioridades. O valor 1 será atribuído à prioridade da primeira variável a ser escalonada.

$$w_p^l = \sum_{h=b_{il}}^{b_{max}} a_{r(h),l} \times Cp_{r(h)} \quad (4.10)$$

onde:

$r(h)$  - índice da variável com prioridade  $h$  no ciclo elementar  $l$ ;

$a_{r(h),l}$  - elemento da matriz  $mtp_{EDF}$  que indica o escalonamento da variável  $vp_{r(h)}$  no ciclo elementar  $l$  (subseção 4.4.2);

$b_{max}$  - prioridade correspondente à última variável periódica a ser escalonada no ciclo elementar  $l$ .

### 4.6.2 Cálculo da Demanda na Rede Devida ao Tráfego Aperiódico ( $W_a^k$ )

O tempo de processamento de todas as requisições aperiódicas pendentes de atendimento na fila do árbitro do barramento é calculado a partir da determinação do número de variáveis que podem ser transmitidas sequencialmente em cada ciclo elementar.

A largura da janela periódica de um ciclo elementar, ou seja, a demanda devida à transmissão de todas as variáveis periódicas escalonadas para transmissão no ciclo elementar  $j$ , é dada pela equação (4.11), onde  $a_{ij}$  é um elemento da matriz de escalonamento periódico ( $1 \leq j \leq N$ ).

$$pw(j) = \sum_{i=1}^{np} a_{ij} \times Cp_i \quad (4.11)$$

A janela aperiódica de um ciclo elementar  $m$  qualquer, inclusive para  $m \geq N$ , tem largura definida pela seguinte expressão:

$$aw(m) = e - pw(m^*) \quad (4.12)$$

onde:

$e$  - duração do ciclo elementar, calculada pela equação (2.7);

$m^*$  - posição relativa do ciclo elementar  $m$  no macro-ciclo. Na equação (4.13)  $N$  representa o número de ciclos elementares contidos no macro-ciclo, fornecido pela equação (2.8).

$$m^* = m - \lfloor \frac{m-1}{N} \rfloor \times N \quad (4.13)$$

O número de transações aperiódicas que podem ser processadas no ciclo elementar  $m$  pode ser calculado a partir dos resultados da aplicação das equações (3.5) e (4.12):

$$nap(m) = \lfloor \frac{aw(m)}{Ca_{max}} \rfloor \quad (4.14)$$

#### 4.6.3 Algoritmo de Cálculo do Tempo de Resposta

O algoritmo 2 apresenta um procedimento que, com base nas equações (4.11) a (4.14), determina o número de ciclos elementares a serem ocupados para o atendimento de todas as requisições aperiódicas pendentes ( $n_{CE}$ ), contados a partir do ciclo elementar  $l$ . Este algoritmo também identifica o ciclo elementar no qual ocorrerá a transmissão da última variável aperiódica da fila de requisições pendentes ( $m$ ).

---

#### Algoritmo 2 Cálculo do número de ciclos elementares contidos em $W_a$

---

```

 $n_{rest} = 2na$ 
 $m = l$ 
while [ $n_{rest} > 0$ ] do
  [calcula  $m^*$  - equação (4.13)]
  [calcula  $pw(m^*)$  - equação (4.11)]
  [calcula  $aw(m)$  - equação (4.12)]
  [calcula  $nap(m)$  - equação (4.14)]
  [ $n_{rest} = n_{rest} - nap(m)$ ]
  [ $m = m + 1$ ]
end while
 $n_{CE} = m - l$ 

```

---

Então, o intervalo de ocupação aperiódica é dado por:

$$W_a^k = (n_{CE} - 1) \times e + pw(m^*) + nap(m^*) \times Ca_{max} \quad (4.15)$$

Uma vez determinados o limite superior da interferência periódica  $I_p^k$ , através da equação (4.8), e a demanda na rede devida ao tráfego aperiódico ( $W_a^k$ ) pela aplicação da equação (4.15), na qual

$n_{CE}$  e  $m$  são calculados pelo algoritmo 2, o tempo de resposta de pior caso da requisição de transmissão de uma variável aperiódica produzida pela estação  $k$  poderá ser encontrado empregando-se a equação (4.7).

Deve aqui ser reiterada a restrição apontada na subseção 3.2.11 quanto à necessidade de manutenção de um intervalo mínimo de tempo entre ativações sucessivas das variáveis aperiódicas para que estas sejam consideradas esporádicas. Neste caso a seguinte relação deve ser incluída no conjunto de condições de garantia de escalonabilidade:

$$Da_i \geq R_a^k \quad (4.16)$$

## **Capítulo 5**

# **Controle do Processo de Geração de Energia Elétrica**

A geração de energia elétrica é um processo contínuo, com rígidos requisitos de segurança e confiabilidade, que refletem a importância deste insumo para todos os ramos de atividade da sociedade humana, seja industrial, comercial ou doméstico. O crescimento econômico, a evolução tecnológica e o avanço das legislações sociais e ambientais têm sido fatores de influência fundamentais para a qualidade dos serviços de eletricidade, com impacto direto no aperfeiçoamento dos sistemas de supervisão e controle das atividades relacionadas à produção industrial de energia elétrica. Como consequência, a operação dos equipamentos e processos passou a ser suportada por um nível crescente de automação e inteligência, resultando na elevação dos padrões de qualidade dos serviços prestados pelas empresas geradoras, transmissoras e distribuidoras de energia e também para a melhoria dos níveis de produtividade, em que pese a eventual redução do emprego de mão-de-obra humana.

### **5.1 O Processo Industrial de Geração de Energia**

#### **5.1.1 Notas Históricas**

A automação de processos tem uma longa história. Entretanto, apenas nas três últimas décadas o uso intensivo de sistemas computacionais nos níveis de produção das organizações industriais tornou-se técnica e economicamente viável, graças ao emprego de processadores digitais como um dos componentes centrais dos dispositivos programáveis, que puderam então ser integrados e aplicados ao controle e supervisão de processos, desde os escalões hierárquicos mais elevados até o nível de chão de fábrica das empresas. Uma característica que exerceu forte influência na decisão de utilizar-se

dispositivos microprocessados para suporte de atividades produtivas, além da versatilidade creditada à sua capacidade de programação foi, sem dúvida, sua inerente capacidade de comunicação.

Os sistemas de supervisão e controle da maioria das estações geradoras, transmissoras e distribuidoras de energia elétrica construídas até o final dos anos setenta no Brasil adotavam uma arquitetura tipicamente centralizada, com seus componentes interligados segundo o modelo de topologia em estrela (Fig. 5.1). As funções lógicas eram realizadas em painéis empregando tecnologia convencional, nos quais relés eletrônicos ou eletromecânicos executavam suas tarefas a partir de uma sala de controle, em resposta a sinais enviados pelos dispositivos de campo (sensores e atuadores) ou a comandos emitidos pelo operador, através de conexões ponto-a-ponto. Esta solução era conhecida como **lógica cabeada**.

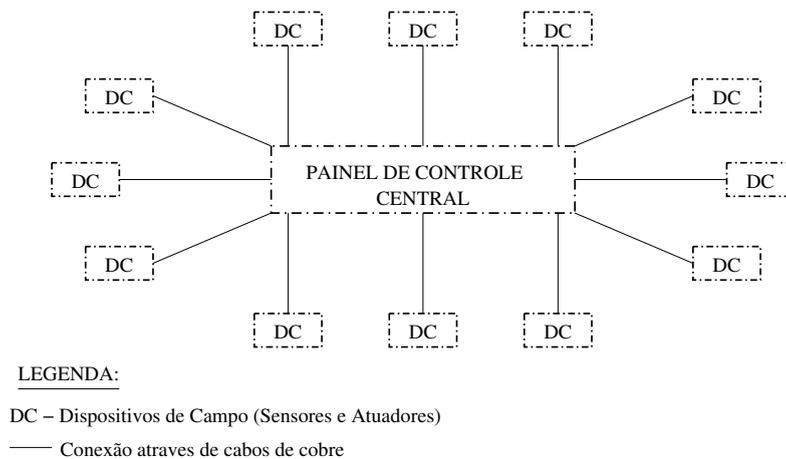


Figura 5.1: Controle Convencional - Arquitetura Centralizada

A análise desta arquitetura permite observar algumas características do modelo, dentre as quais destacam-se:

1. Uma complexa e extensa malha de cabos e fios de cobre, empregada para realizar a interligação entre os sensores e atuadores associados aos equipamentos de campo e os dispositivos do sistema de controle e supervisão.
2. Instrumentos e dispositivos auxiliares de supervisão, controle e alarmes instalados em painéis localizados em uma sala de controle centralizada.
3. Estados operacionais dos equipamentos do nível de campo codificados como sinais lógicos (*on-off*) ou sinais analógicos proporcionais às grandezas do processo, adotando níveis de tensões e correntes padronizados (ou não).

4. Limitação da quantidade e dos detalhes das informações disponibilizadas pelas variáveis do processo, em função da baixa capacidade de comunicação, decorrente do emprego de tecnologia convencional, em termos relativos.
5. Dificuldade de implementação de redundância no nível dos painéis de controle. Conseqüentemente o sistema de supervisão e controle representa, ele mesmo, um ponto de falha de modo comum.
6. Elevado custo de implementação de atividades de manutenção, atualização e expansão do sistema de supervisão de controle.

A exemplo da Usina Hidrelétrica de Itaipu, cujos detalhes técnicos do projeto foram registrados por Cotrim *et al.* [11], as plantas de geração de energia elétrica ocupam, via de regra, áreas relativamente extensas, onde equipamentos com características funcionais e operacionais diversas são interligados lógica e fisicamente para efetuar a composição e a integração das unidades produtivas.

### 5.1.2 Sistemas Computacionais de Controle Aplicados a Usinas Geradoras e Subestações de Energia Elétrica no Brasil [4]

A filosofia norteadora da aplicação da tecnologia digital a sistemas de controle de processos de usinas geradoras e subestações das principais empresas do setor elétrico brasileiro começou a ser estabelecida pela ELETROBRÁS nos fins da década de 80, a partir da criação do Comitê de Informática Industrial e Automação (COMIIA).

Um dos resultados da iniciativa da ELETROBRÁS foi a produção, pelo Grupo de Automação e Digitalização de Usinas e Subestações (GADUS), de um documento contendo uma descrição geral dos aspectos físicos e funcionais dos processos inerentes a esta classe de instalação industrial. Encontra-se também, neste documento, a definição dos requisitos básicos para o modelamento dos sistemas de controle dos processos de geração e de transmissão de energia elétrica.

## 5.2 Processo Físico (Objeto Controlado)

A geração de energia em usinas hidrelétricas constitui-se, basicamente, de um processo de conversão de energia mecânica em energia elétrica. O conjunto formado pela **turbina hidráulica, gerador, equipamentos agregados e sistemas auxiliares** é denominado **unidade geradora**.

O funcionamento de uma unidade geradora caracteriza-se por estados operacionais bem definidos e pelas transições entre estes estados, tendo como principais variáveis de referência para controle do processo a tensão gerada, a velocidade (ou frequência) e a potência elétrica (Fig. 5.2).

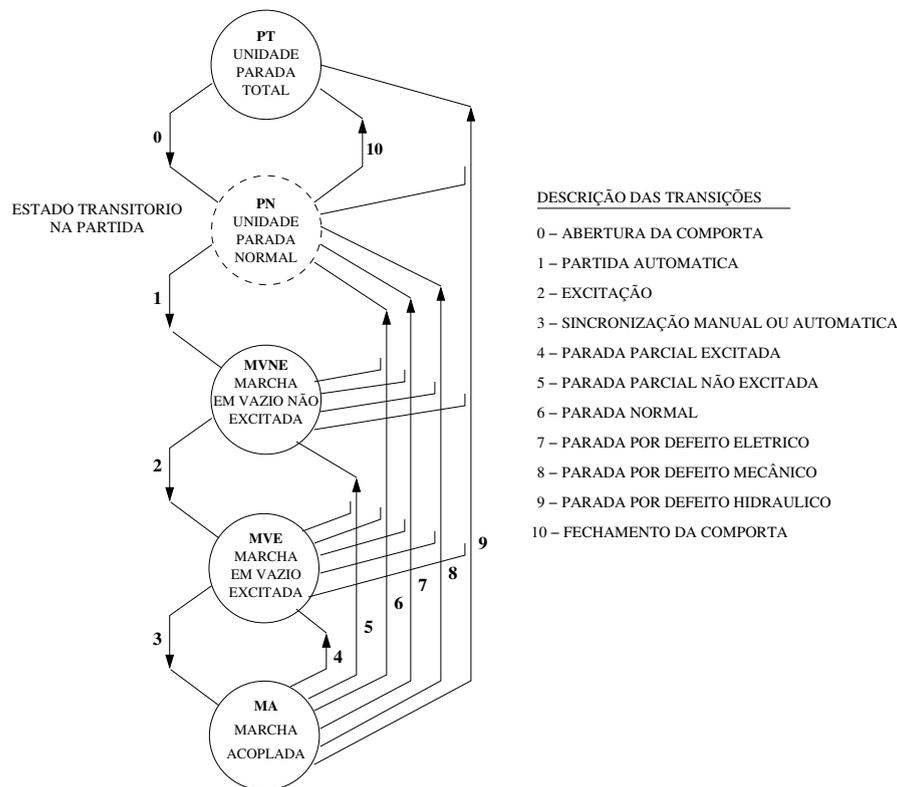


Figura 5.2: Estados e Transições Operacionais da Unidade Geradora

Cumprir ainda registrar que a análise do funcionamento de uma unidade geradora considera duas condições de operação, denominadas como **regime transitório** e **regime permanente**, que definem o seu comportamento eletro-mecânico. O resultado desta análise e a experiência adquirida com a operação de equipamentos similares representam as principais fontes de informações para a definição dos requisitos do sistema.

A seguir é apresentada uma breve descrição dos principais equipamentos existentes em uma planta de geração de energia elétrica, como a Usina Hidrelétrica de Itaipu [11], cujas características e requisitos têm influência no projeto do suporte de comunicação do sistema de controle do processo.

### 5.2.1 Turbina Hidráulica

A potência ativa da unidade geradora é assegurada pela turbina hidráulica que tem como função converter a energia potencial da água em trabalho mecânico. A turbina serve como máquina motriz

primária do gerador e depende do **regulador de velocidade** para controlar as variáveis do processo tratadas por um **controlador**, que poderá ser **analógico** ou **digital**. Os sensores, atuadores e dispositivos auxiliares eletro-hidráulicos e mecânicos empregam um conjunto de sinais de comando, controle e de supervisão para garantir a correta e segura resposta da unidade às solicitações do ambiente, de acordo com os parâmetros aplicados pelo sistema de controle.

A **comporta da tomada d'água**, o conduto forçado, o distribuidor e alguns dispositivos auxiliares formam o conjunto hidráulico responsável pelo controle do fluxo de água que aciona a turbina. O distribuidor e a comporta da tomada d'água constituem-se ainda em equipamentos de segurança pois, através do seu acionamento em determinadas condições anormais de operação, a unidade geradora poderá ser desligada, de forma automática ou manual, minimizando os riscos para a integridade física das instalações.

A turbina responde pelo comportamento mecânico da unidade geradora, tendo como variáveis de interesse do sistema de controle a velocidade angular (rotação) e a demanda de potência solicitada pelos consumidores.

### 5.2.2 Gerador

O gerador é uma máquina elétrica rotativa que realiza a conversão eletromecânica de energia com base na lei de Faraday [14]. O gerador é o componente da unidade geradora que, em última análise, representa a razão da existência da planta - a produção de energia elétrica.

Os geradores de corrente alternada são denominados **alternadores**, **geradores síncronos** ou **máquinas síncronas** e têm, como elementos diretamente associados à produção de energia elétrica, o **estator** e o **rotor**. Os geradores empregados em usinas hidrelétricas são mecanicamente acionados por turbinas hidráulicas que geralmente operam a baixas velocidades (valores inferiores a 100 rpm) se comparadas com os turbogeradores, cujas velocidades podem atingir valores nominais superiores a 1000 rpm.

Para realizar a sua função de maneira segura e adequada - disponibilização da potência elétrica requerida em seus terminais sob a forma de tensões e correntes, caracterizando o comportamento elétrico da unidade - o gerador depende de um certo número de **equipamentos agregados** e **sistemas auxiliares**, que operam sob coordenação do sistema de supervisão e controle da planta.

### 5.2.3 Equipamentos Agregados e Sistemas Auxiliares

Os equipamentos agregados e sistemas auxiliares da unidade geradora geralmente encontram-se distribuídos pela planta. Em virtude da necessidade de intercâmbio de informações de estado e de sinais de controle, comando e supervisão entre eles e os demais componentes do sistema, são necessários os serviços de um suporte de comunicação que atenda aos requisitos de previsibilidade, segurança e disponibilidade. Dentre estes equipamentos merecem destaque, em função de sua essencialidade para a operação da planta, os seguintes:

**Sistema de excitação** - O sistema de excitação é responsável pelo controle da tensão e da potência reativa da máquina síncrona. Do ponto de vista do sistema elétrico o gerador é o objeto controlado e o sistema de excitação é o controlador que atua sobre a corrente que é aplicada aos enrolamentos de campo do rotor para criar e manter o campo eletromagnético através do qual é realizada a conversão da energia mecânica fornecida pela turbina na energia elétrica disponibilizada nos terminais do gerador.

**Sistema de proteção elétrica** - As condições operacionais dos equipamentos e sistemas elétricos devem ser permanentemente monitoradas de modo a garantir que as variáveis do processo permaneçam dentro dos limites especificados. Esta tarefa é desempenhada pelo sistema de proteção elétrica, o qual pode ser modelado como um conjunto de sensores e atuadores que, com base em algoritmos especiais, são capazes de identificar condições anormais de operação, tais como sobrecargas, curtos-circuitos, subtensões, sobretensões, sobrecorrentes e demais ocorrências causadas por defeitos internos ou externos aos equipamentos elétricos.

Os algoritmos de proteção são tarefas executadas por dispositivos dedicados que, tendo como variáveis de entrada tensões e correntes medidas em determinados pontos do sistema elétrico, detectam as condições que representam riscos para a integridade dos equipamentos, instalações, ou mesmo para a continuidade do fornecimento de energia elétrica aos consumidores. Neste caso estes algoritmos, denominados também como **funções de proteção**, produzem os sinais de comando que iniciam os procedimentos para isolar automaticamente os componentes defeituosos dentro dos limites de tempo especificados nas definições de requisitos do sistema.

A proteção de geradores síncronos é um assunto de elevado nível de especialização em função da complexidade e da diversidade de condições anormais de operação possíveis de ocorrência [31].

No contexto dos sistemas de tempo real as funções de proteção elétrica pertencem ao conjunto das tarefas de missão crítica.

**Transformador elevador** - Geralmente os recursos hidráulicos que podem ser aproveitados para produção de energia elétrica situam-se em locais distantes dos grandes centros consumidores. Além do mais, por razões técnico-econômicas, a geração de energia elétrica é feita em níveis de tensão que não são os mais adequados para transmissão a grandes distâncias. Por estas razões a tensão de geração deve ser elevada para os patamares de transmissão, que podem atingir valores 50 vezes superiores aos níveis de tensão de geração.

O transformador elevador (geralmente instalados como um banco de três transformadores monofásicos) é o equipamento elétrico que tem por função elevar a tensão de geração para o nível de transmissão.

**Subestação elevadora** - A potência elétrica produzida pela planta, sob a forma de tensões e correntes fornecidas pelo transformador elevador de cada unidade, é concentrada em barras de alta tensão para então ser entregue ao sistema elétrico através das linhas de transmissão.

A subestação elevadora localiza-se o mais próximo possível dos geradores e inclui, entre seus componentes, equipamentos de manobra (disjuntores e seccionadoras), pára-raios, transformadores de instrumentos, etc.

Atualmente as subestações elevadoras, em especial as localizadas no interior das casas de força das usinas, empregam como meio isolante o hexafluoreto de enxofre, um gás pesado conhecido como **SF<sub>6</sub>**, o que permite a compactação das instalações e a redução substancial da área necessária à instalação física dos equipamentos.

**Sistemas de resfriamento** - As inevitáveis perdas que ocorrem durante o processo de conversão de energia produz calor que eleva a temperatura de funcionamento dos equipamentos. Este calor deve então ser trocado com o ambiente para evitar superaquecimentos que têm como consequência a redução da vida útil dos equipamentos. Cada equipamento elétrico deve então dispor de algum recurso, mais ou menos sofisticado, com o papel de manter a temperatura dentro dos limites aceitáveis pelos materiais empregados em sua construção.

**Equipamentos de sincronização** - A unidade geradora assume diversos estados operacionais antes de estar pronta para contribuir com a sua cota de energia para o sistema elétrico (Fig. 5.2). A

determinação do instante de fechamento do disjuntor da unidade para interligar o gerador com o sistema representa uma condição crítica pois a tensão nos terminais do gerador deverá estar sincronizada com a tensão do sistema elétrico. Para que isto aconteça deverão ser obedecidas as conhecidas “quatro condições essenciais para sincronização” - o módulo, a frequência, a seqüência de fases da tensão nos terminais do gerador e o ângulo de fase entre as tensões do gerador e do sistema - deverão estar abaixo dos valores máximos especificados. A sincronização do gerador representa uma condição de tempo real crítica.

**Serviços auxiliares** - Todos os equipamentos que contribuem para a operação da unidade geradora tais como sensores, atuadores, posicionadores, bombas, ventiladores, iluminação, dispositivos de controle e supervisão, etc., necessitam de energia para funcionar. Portanto cada planta dispõe de um sistema elétrico independente, composto de transformadores, quadros elétricos, baterias de acumuladores e conversores com a função de prover tensões elétricas para alimentação dos dispositivos e equipamentos em diversos níveis. Os sistemas de serviços auxiliares são essenciais para o funcionamento das plantas de geração de energia elétrica.

#### 5.2.4 Requisitos Ambientais do Processo Físico

Os processos industriais de geração de energia elétrica apresentam características ambientais que devem ser consideradas na definição de requisitos do suporte de comunicação do sistema de supervisão e controle. Dentre estas características, destacam-se as seguintes:

- Equipamentos e dispositivos distribuídos em área relativamente extensa, com elevado grau de intercâmbio de informações entre eles;
- Dispositivos localizados mais próximos ao processo, tais como os sensores e atuadores, sujeitos a condições ambientais severas, sob influência de campos eletromagnéticos intensos, vibrações, umidade e temperaturas elevadas;
- Variáveis periódicas e esporádicas, algumas com restrições temporais críticas;
- Demanda de serviços pelos usuários, tais como atualizações de ordens de ajustes (*set-points*), emissões de relatórios, consultas a valores correntes das variáveis do processo assim como a dados históricos, cálculos diversos e atividades de manutenção do sistema. Estas facilidades geralmente não estão associadas a prazos críticos, podendo ser atendidas com base em estratégias de melhor esforço, através de tarefas aperiódicas.

O ambiente caracterizado pelos requisitos acima mencionados definem uma aplicação típica de controle de processo distribuído, cujo desempenho depende de um suporte de comunicação baseado em redes locais que podem ser adequadamente mapeadas aos níveis hierárquicos modelados pela estrutura apresentada na subseção 2.6.1.

### 5.3 Sistema de Controle do Processo

O conjunto de recursos de *hardware* e de *software* empregados para executar as funções lógicas que têm como argumentos as variáveis do ambiente e os parâmetros que influenciam o comportamento da unidade geradora, sob todas as condições de serviço e modos de operação, compõe o sistema de supervisão e controle do processo.

#### 5.3.1 Funções Lógicas de Controle

Os estados e transições operacionais ilustrados pela Fig. 5.2 são controlados a partir da execução de tarefas tais como: preparação de partida, partida, sincronização, tomada de carga, controle de velocidade e da potência ativa, controle da tensão gerada e da potência reativa, parada normal, parada de emergência, etc.

Além das funções de controle propriamente ditas o sistema provê serviços adicionais sob a forma de aplicativos que, tendo como base o resultado do processamento das variáveis do ambiente, indicam os estados dos equipamentos, gerenciam a base de dados, realizam o tratamento de alarmes, efetuam cálculos e produzem relatórios diversos sob demanda.

A intervenção do operador humano para supervisão e controle do funcionamento dos equipamentos é efetuada por meio da interface homem-máquina (IHM). Do ponto de vista do sistema de controle o operador é parte do ambiente e atua sobre o processo físico aplicando sinais de comando e ordens de ajuste, efetuando leituras de valores correntes de variáveis, etc. A IHM pode também ser empregada como ferramenta de configuração do sistema ou ainda como recurso para demais atividades de manutenção.

#### 5.3.2 Níveis de Supervisão e Comando

A definição de requisitos do sistema estabelece três níveis físicos e lógicos a partir dos quais poderão ser efetuados o controle e supervisão do processo. Em cada um destes locais encontram-

se recursos e ferramentas de suporte para as tomadas de decisão capazes de influenciar direta ou indiretamente o comportamento operacional das unidades geradoras.

No nível de comando mais próximo do processo físico (**nível do processo**) encontram-se os recursos necessários para suporte dos trabalhos realizados durante a fase de implantação do empreendimento (comissionamento) e, posteriormente, para atividades de manutenção dos equipamentos.

Os equipamentos, acessórios e demais facilidades disponíveis na sala de controle local, inclusive a IHM, proporcionam os recursos que permitem o controle da operação da unidade a partir do **nível local**. Tais facilidades utilizam os serviços do nível do processo e, por sua vez, oferecem um conjunto de serviços mais sofisticados que aqueles disponibilizados pelo nível de comando mais próximo do processo físico.

O gerenciamento técnico da planta é efetuado pela equipe de operadores que, durante a maior parte do tempo, permanece na sala de controle central, com as unidades funcionando no modo de **controle remoto**. A partir deste nível as unidades podem ser controladas coletivamente (**controle conjunto**) ou de forma separada das demais máquinas da planta (**controle individual**).

### 5.3.3 Suporte de Comunicação de Dados

A Fig. 5.3 representa, de forma simplificada, a arquitetura de *hardware* adotada para o suporte de comunicação do Sistema Digital de Supervisão e Controle Local (SDSC) a ser instalado na Usina Hidrelétrica de Itaipu. A topologia adotada possibilita o mapeamento das redes do processo aos níveis hierárquicos definidos no modelo do sistema de comunicação (ver subseção 2.6.1).

O nível 0 compreende a interface de instrumentação, que suporta o intercâmbio físico dos sinais que trafegam entre os controladores de entrada e saída (**controladores de E/S**) e os equipamentos do processo. Neste nível as conexões entre os dispositivos podem ser efetuadas da forma ponto-a-ponto ou através de conexões seriais, empregando os recursos disponibilizados por certos protocolos de redes de campo.

Os controladores de E/S realizam o tratamento primário dos sinais oriundos do processo, assim como decodificam, em sinais elétricos que são aplicados aos equipamentos de campo, os dados que são transmitidos pelo **controlador de célula** através da rede de campo. A interconexão dos controladores de E/S através da rede de campo resulta na implementação de um banco de dados de tempo real permitindo a cooperação entre os dispositivos de controle do processo físico. Algumas tarefas com

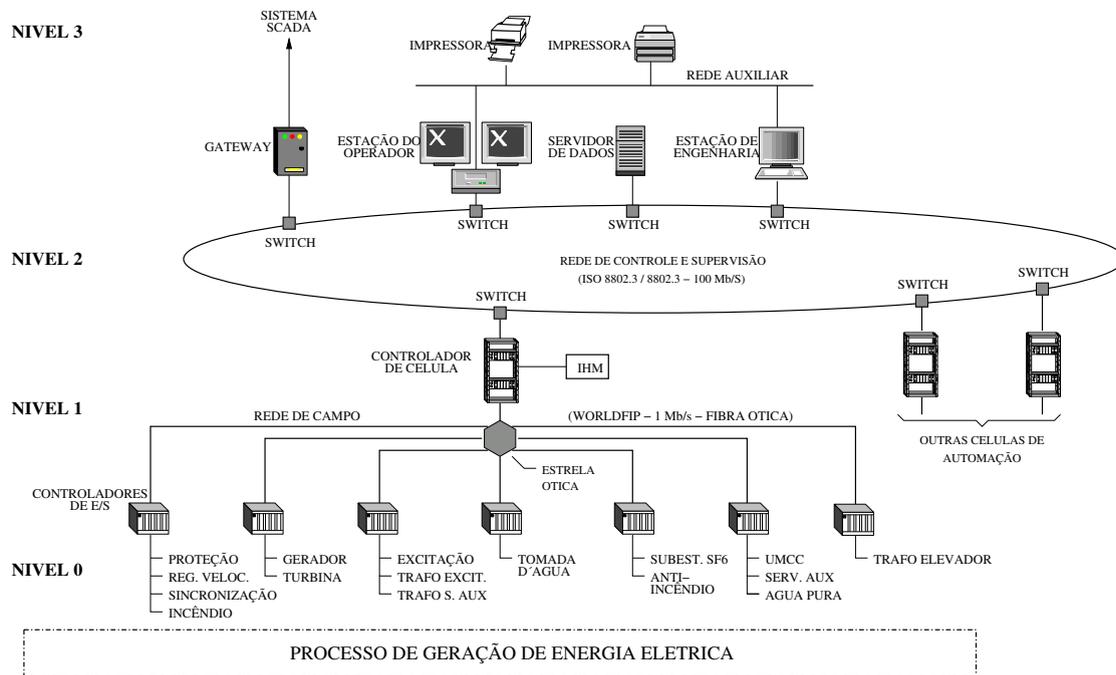


Figura 5.3: Arquitetura de *Hardware* do SDSC

restrições de tempo de resposta mais rigorosas podem ser executadas localmente nos controladores de E/S.

A estação configurada como controlador de célula é dotada de recursos suficientes para prover os seguintes serviços:

- integração dos componentes da interface de instrumentação através da rede de campo;
- implementação das tarefas que realizam as funções lógicas de controle da unidade geradora (seção 5.3.1);
- *gateway* para intercâmbio de dados entre as redes de campo (nível 1) e a rede de controle e supervisão (nível 2).

O controlador de célula dispõe ainda de todos os recursos necessários para o suporte da operação autônoma da unidade geradora, incluindo uma IHM local (nível 1). Portanto, durante uma eventual indisponibilidade dos recursos do nível 2 (sala de controle local), a operação da unidade poderá ser comandada, no modo individual e local, a partir da IHM do nível 1.

No nível 2 localizam-se as estações de operação empregadas para supervisão e controle local das células produtivas da planta que, neste caso, são as unidades geradoras. Encontram-se também neste nível o servidor de dados, impressoras, o *gateway* entre a rede de controle e supervisão e os

equipamentos da sala de controle central (nível 3) e demais componentes de suporte à operação e às atividades de engenharia.

O nível 3 implementa o modo de **supervisão e comando remoto** através do SCADA (sistema de supervisão, controle e aquisição de dados da planta).

### 5.3.4 Tempos de Resposta do Sistema

O documento de requisitos do sistema [23] especifica os limites para os tempos de tráfego dos sinais de comando e controle pelo suporte de comunicação (Fig. 5.3).

**Sinais gerados pelo processo:** Na Tabela 5.1 estão discriminadas as classes de sinais gerados pelo processo e os tempos máximos para sua disponibilização nos componentes do nível 2 (estações de operação, servidores de dados, *gateway*, etc.) a partir do instante de sua aquisição pelos controladores de E/S (nível 0);

Classe de Sinal Gerado pelo Processo	Tempo Máximo Admissível Para Disponibilização (s)	
	Na Estação de operação	No <i>Gateway</i> de Comunicação
Mudança de estado	1,000	0,500
Medição analógica	1,000	0,500
Alarmes	1,000	0,500
Seqüência de eventos (SOE)	2,000	2,000

Tabela 5.1: Atrasos Máximos Admissíveis para os Sinais Gerados pelo Processo

**Sinais emitidos pelo operador:** As ordens de comando e os sinais de controle (ordens de ajustes) enviadas para o processo a partir da interface homem-máquina deverão estar disponibilizados nos terminais dos controladores de E/S em um tempo máximo de 0,500 s, medido a partir de sua entrega ao suporte de comunicação pelos equipamentos do nível 2 (estação de operação ou *gateway* de comunicação com o SCADA).

Deve ainda ser registrado que, para atender às restrições temporais de certas tarefas de tempo real residentes no controlador de célula (nível 1) ou nos controladores de E/S, algumas variáveis que trafegam pela rede de campo têm prazos de entrega rígidos, com valores da ordem a 100 milissegundos.

## Capítulo 6

# Estudo de Caso - Cálculo do Tempo de Resposta da Rede de Campo do SDSC

Este Capítulo apresenta uma aplicação do método de cálculo do tempo de resposta de pior caso da rede WorldFIP cujo desenvolvimento foi detalhado no Capítulo 4. Para tal são utilizados os dados levantados durante a fase preliminar do projeto do sistema digital de supervisão e controle (SDSC) das duas unidades geradoras que estão sendo instaladas na Usina Hidrelétrica de Itaipu.

O objeto controlado está descrito, de maneira sucinta, na seção 5.2 e a estrutura hierárquica do suporte de comunicação do SDSC encontra-se representada esquematicamente pela Fig. 5.3.

Este estudo de caso tem como finalidade a análise do escalonamento das variáveis diretamente associadas às funções de controle de processo. Portanto os cálculos dos tempos de resposta consideram apenas os sinais com requisitos de entrega em tempo limitado. Isto restringe o alcance do seu escopo às variáveis periódicas e às variáveis aperiódicas com prioridade urgente. As variáveis aperiódicas com prioridade normal, que podem ser atendidas pelo suporte de comunicação segundo uma estratégia de melhor esforço, não serão tratadas nesta Dissertação.

### 6.1 Esquema Lógico de Interconexão dos Controladores

A Fig. 6.1 mostra, de forma simplificada, a distribuição lógica dos controladores conectados à rede WorldFIP. Neste diagrama destaca-se o controlador de célula (**A000**), que localiza-se física e logicamente entre a rede de supervisão e controle (nível 2) e a rede de campo (nível 1), e os controladores de E/S (**A001** a **A007**) que constituem-se nos componentes da interface de instrumentação.

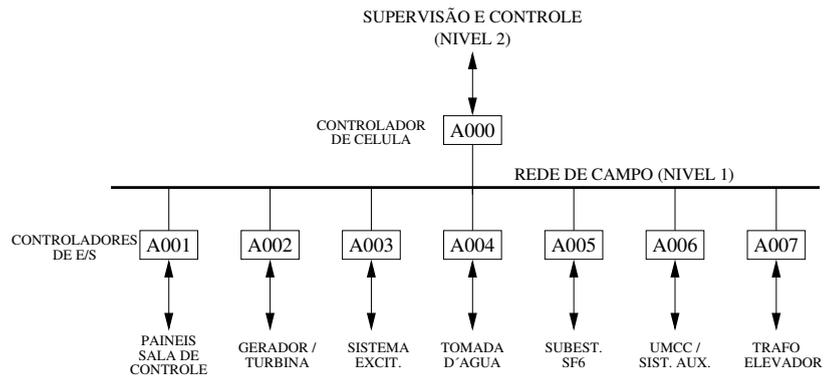


Figura 6.1: Arranjo Lógico dos Controladores de Entrada e Saída

## 6.2 Sinais Físicos do Processo

A principal fonte dos dados utilizados para o dimensionamento dos componentes do sistema de controle é um documento técnico denominado **lista de pontos** que contém a descrição dos atributos dos sinais cujos objetos digitais serão transportados pelas variáveis transacionadas através da rede de comunicação.

O estado corrente dos equipamentos e demais componentes do objeto controlado é disponibilizado, pela instrumentação do sistema (sensores), sob a forma de sinais elétricos que são adquiridos pelos controladores de E/S (Fig. 6.1) e convertidos nos objetos digitais representando a imagem do processo. Estes objetos são transmitidos através da rede de campo e lidos pelos nós consumidores que os utilizam como argumentos das tarefas da aplicação distribuída.

De maneira similar os sinais de comando e as ordens de ajuste geradas nos níveis superiores da estrutura hierárquica, e também os sinais produzidos pelas funções lógicas residentes nos controladores, são empacotados como quadros WorldFIP (seção 3.2.1) e enviados para os atuadores através da rede de campo sob a forma de variáveis que são consumidas pelos nós que realizam a interface com o processo. Os conversores D/A dos controladores de E/S então efetuam o acondicionamento dos dados para a forma de sinais elétricos em níveis de tensão compatíveis com os equipamentos de campo.

### 6.2.1 Sinais Discretos

Os estados dos equipamentos, ordens de comando e os resultados do processamento de certas tarefas lógicas executadas nos controladores são eventos que podem ser representados por sinais de

natureza binária (*on-off*) denominados **sinais discretos**, **sinais booleanos** ou **sinais lógicos**.

Para fins de atribuição de prioridade de acesso ao meio de comunicação os sinais discretos são classificados, neste trabalho, em três grupos:

**Controle** (*CTL<sub>d</sub>*) - sinais discretos utilizados como argumentos das funções que executam tarefas de controle do processo, em certos casos com realimentação em malha fechada e com restrições de tempo real críticas. Os sinais de controle são processados de forma **periódica** [33].

**Sinalização** (*SIN<sub>d</sub>*) - sinais representativos dos estados dos equipamentos, que entretanto não são empregados como parâmetros das funções de controle do processo. Estes dados devem ser transmitidos para as tarefas residentes nos níveis mais elevados da estrutura de controle no menor tempo possível por razões de consistência de informação. Apesar de não terem requisitos de tempo real crítico é conveniente que as variáveis que transportam estes sinais tenham garantia de entrega em tempo limitado. Portanto eles são tratados, pela rede de campo, como sinais **aperiódicos** com nível de prioridade **urgente**.

**Alarme** (*ALM<sub>d</sub>*) - informações sobre as condições anormais de operação que devem ser reportadas ao operador, através das funções de supervisão, no menor prazo possível. Os sinais classificados neste grupo geralmente não são utilizados pelas tarefas que executam funções de controle do processo. Portanto eles não têm restrições de tempo real crítico e podem ser atendidas, pelo sistema de comunicação, em uma base de melhor esforço. Por razão estas informações de alarme são consideradas, pelo suporte de comunicação, como sinais **aperiódicos** com nível de prioridade **normal**.

A Tabela 6.1 apresenta a quantificação dos sinais discretos classificados de acordo com os atributos funcionais adotados como critérios de atribuição das prioridades de acesso ao meio de transmissão. Os sinais discretos são codificados, no nível dos controladores de E/S, como objetos de **3 bits**.

### 6.2.2 Sinais Analógicos

Os atributos do processo que variam de forma contínua no tempo e que podem ser modelados por funções do tipo  $y = f(t)$ , onde  $y, t \in R$ , são denominados *sinais analógicos*. Para fins de atribuição de prioridade de acesso ao meio de transmissão os sinais analógicos são classificados nos quatro grupos seguintes, dos quais os três primeiros são periódicos e o último é processado de forma aperiódica:

Estação Produtora	Sinais Discretos		
	Controle ( <i>CTL_d</i> )	Sinalização ( <i>SIN_d</i> )	Alarme ( <i>ALM_d</i> )
A000	125	-	-
A001	20	16	170
A002	112	117	146
A003	18	83	45
A004	10	20	22
A005	60	60	30
A006	20	60	120
A007	60	40	80
Total	425	380	613

Tabela 6.1: Quantificação dos Sinais Discretos

**Grandeza elétrica** (*MGE\_a*) - valores das tensões e correntes elétricas medidos nos terminais da unidade geradora e dos equipamentos de serviços auxiliares disponibilizados pelos transformadores de potencial e transformadores de corrente. Antes de serem aquisitados, de forma **periódica**, pelos controladores de E/S estes sinais são submetidos a um pre-tratamento em dispositivos denominados **medidores de grandezas elétricas**.

**Transdutor** (*TRD\_a*) - grandezas de natureza não elétrica produzidas pelo processo, traduzidas por valores de pressão, fluxo, posição, nível, etc., convertidas em sinais elétricos proporcionais por **transdutores de sinal**. No caso em estudo os transdutores fornecem sinais de corrente contínua na faixa de 4 – 20mA. Os sinais do grupo TRD são aquisitados pelos controladores da interface de instrumentação de forma **periódica**.

**Termômetro de Resistência** (*RTD\_a*) - em certos casos é conveniente que a medição da temperatura em determinados pontos dos equipamentos seja efetuada através de resistores construídos com materiais cujas curvas de variação temperatura  $\times$  resistência são conhecidas. Então, os valores de temperatura são inferidos a partir da leitura **periódica** dos valores de resistência fornecidos por estes sensores, também conhecidos como **termômetros de resistência** (*resistance thermometers*) [35].

**Ordem de ajuste** (*STP\_a*) - comandos recebidos dos níveis superiores da estrutura de controle contendo os valores de referência para os pontos de operação desejados para os equipamentos (potência ativa, potência reativa, tensão, etc.). Estes sinais, denominados **ordens de ajuste** (*set-points*), em geral não variam no tempo em uma base contínua. Portanto eles são processados como sinais **aperiódicos** com nível de prioridade **urgente**.

A Tabela 6.2 apresenta os sinais analógicos considerados no cálculo do tempo de resposta da rede, classificadas por grupo funcional e com discriminação das respectivas estações produtoras.

A palavra binária dos conversores A/D e D/A dos controladores de E/S têm largura de 14 bits. Por conta dos bits de validação adicionados pelo *hardware*, os objetos digitais correspondentes aos sinais analógicos são codificados em **22 bits**.

Estação Produtora	Grupos dos Sinais Analógicos			
	Grandeza Elétrica ( <i>MGE_a</i> )	Transdutor ( <i>TRD_a</i> )	Termômetro de Resistência ( <i>RTD_a</i> )	Ordem de Ajuste ( <i>STP_a</i> )
A000	-	-	-	34
A001	-	6	-	-
A002	9	26	32	-
A003	-	8	-	-
A004	-	3	-	-
A005	-	5	-	-
A006	8	7	-	-
A007	-	9	20	-
Total	17	64	52	34

Tabela 6.2: Quantificação dos Sinais Analógicos

### 6.3 Critérios de Formação das Variáveis e Quantificação dos Parâmetros da Rede

A análise da escalonabilidade do tráfego de variáveis em uma rede WorldFIP visa determinar a faixa de valores a ser atribuídos a determinados parâmetros de maneira que seja garantido o atendimento das restrições temporais especificadas para o suporte de comunicação.

Dentre os fatores enumerados na seção 4.1 serão considerados, como parâmetros de controle que influenciam direta ou indiretamente os tempos de resposta da rede, as frequências de ativação dos sinais, as larguras das estruturas de dados recebidas pela camada de enlace, a taxa de transmissão de dados e o tempo de circulação. São adotados os seguintes valores para esses dois últimos parâmetros:

- Taxa de transmissão (*tx\_rate*): 1 Mb/s e 2,5 Mb/s;
- Tempo de circulação ( $t_r$ ):  $20 t_{mac}$ ;

Os critérios de atribuição dos valores dos períodos das variáveis ( $T_i$ ) e das larguras das estruturas de dados ( $VAR_{AP_i}$ ) estão detalhados nas subseções 6.3.1 e 6.3.2 a seguir.

### 6.3.1 Atributos das Variáveis Periódicas

Os períodos mostrados na Tabela 6.3 são atribuídos de forma a atender as restrições e requisitos funcionais associados aos grupos de sinais descritos na subseção 6.2.1.

Natureza do Sinal	Valores dos Períodos	
	Grupo	Período (ms)
Controle	$CTL_d$	$T_1$
Grandeza Elétrica	$MGE_a$	20
Transdutor	$TRD_a$	50
Termômetro de Resistência	$RTD_a$	250

Tabela 6.3: Valores dos Períodos - Grupos de Sinais Periódicos

O período dos sinais pertencentes ao grupo  $CTL_d$  ( $T_1$ ), referente às variáveis com maior frequência de ativações, será utilizado como um dos parâmetros de entrada nas simulações para determinação dos tempos de resposta do suporte de comunicação. Serão adotados, para este parâmetro, valores de **5**, **10** e **20** milissegundos. Os períodos das variáveis pertencentes aos grupos  $MGE_a$ ,  $TRD_a$  e  $RTD_a$  são considerados como constantes.

A largura da estrutura de dados transportada pela PDU, juntamente com o tempo de circulação e a taxa de transmissão, influi na duração das transações, cujo valor pode ser calculado através da equação (4.6). Serão adotados, para avaliação dos tempos de resposta da rede, valores de **12**, **24**, **36**, **48**, **64**, **96** e **126** bytes para o parâmetro  $VAR_{AP}$ . A equação (3.9) fornece o valor de 36,4% para o rendimento teórico da rede no caso da adoção de  $VAR_{AP}$  de 12 bytes. A utilização de estruturas de dados de 126 bytes resulta em uma eficiência de 85,7%, de acordo com a equação (3.9).

### 6.3.2 Atributos das Variáveis Aperiódicas

As ordens de ajuste ( $STP_a$ ), as informações de mudanças de estado de equipamentos ( $SIN_d$ ) e as condições anormais de operação ( $ALM_d$ ) são notificadas ao sistema de supervisão e controle através de sinais aperiódicos. Os níveis de prioridade definidos para as variáveis que transportam os objetos referentes a estes grupos de sinais estão descritos nas subseções 6.2.1 e 6.2.2.

A exemplo das variáveis periódicas, as durações das transações de variáveis aperiódicas serão calculadas com base em estruturas de dados com larguras de **12**, **24**, **36**, **48**, **64**, **96** e **126** bytes.

### 6.3.3 Sinais Periódicos e Aperiódicos × Estações Produtoras

A Tabela 6.4 apresenta o resumo numérico dos grupos de sinais periódicos e aperiódicos associados às suas respectivas estações produtoras. A partir destas informações poderá ser quantificado o número de variáveis produzidas em cada estação levando-se em conta o tamanho de cada objeto (3 bits para os sinais discretos e 22 bits para os sinais analógicos) e o valor de VAR\_AP adotado para caso simulado.

Estação Produtora	Grupos de Sinais Periódicos				Grupos de Sinais Aperiódicos		
	<i>CTL_d</i>	<i>MGE_a</i>	<i>TRD_a</i>	<i>RTD_a</i>	<i>STP_a</i>	<i>SIN_d</i>	<i>ALM_d</i>
A000	125	-	-	-	34	-	-
A001	20	-	6	-	-	16	170
A002	112	9	26	32	-	117	146
A003	18	-	8	-	-	83	45
A004	10	-	3	-	-	20	22
A005	60	-	5	-	-	60	30
A006	20	8	7	-	-	60	120
A007	60	-	9	20	-	40	80

Tabela 6.4: Grupos de Sinais Periódicos e Aperiódicos × Estações Produtoras

## 6.4 Aplicação do Método de Cálculo

A análise da escalonabilidade do tráfego periódico e o cálculo dos tempos de resposta a requisições de transmissão de variáveis aperiódicas pela rede WorldFIP objeto do presente estudo de caso foram efetuados aplicando-se o método apresentado no Capítulo 4. Os testes e verificações iniciais das equações e dos algoritmos foram executados tendo como base as funções cujos códigos-fonte encontram-se disponibilizados no Apêndice B.

### 6.4.1 O Programa ANAFIP

Tendo em vista a portabilidade e a independência de plataforma de execução foi desenvolvida uma ferramenta de *software*, implementada em Java, para suporte das simulações. Esta ferramenta, denominada **ANAFIP**, tem como base as funções listadas no Apêndice B, portadas de linguagem C para Java. O programa ANAFIP dispõe de uma interface gráfica que permite a realização das seguintes operações (ver Apêndice C):

- carregamento dos parâmetros da rede e dos atributos variáveis;

- escolha do algoritmo de escalonamento (RM ou EDF);
- apresentação dos tempos de resposta calculados para cada estação produtora de variáveis.

A ferramenta ANAFIP, antes de efetuar o cálculo dos tempos de resposta a requisições de tráfego aperiódico, executa uma etapa preparatória, usando como dados os atributos das variáveis periódicas do sistema. Esta etapa preparatória consiste, principalmente, dos seguintes passos:

- determinação do ciclo elementar e do macro-ciclo do sistema;
- verificação da escalonabilidade das variáveis periódicas;
- montagem da matriz de escalonamento periódico com base no algoritmo escolhido (RM ou EDF);

#### 6.4.2 Resultados Numéricos das Simulações

Os tempos de resposta produzidos pelo programa ANAFIP, relativos aos sinais mostrados na Tabela 6.4, considerando os parâmetros da rede WorldFIP e atributos das variáveis discriminados na seção 6.3, encontram-se organizados sob a forma de tabelas no Apêndice A. Os resultados mostrados nas Tabelas A.1 a A.14 servem como base para os seguintes comentários:

1. O principal fator de influência no tempo de resposta das variáveis aperiódicas é a taxa de transmissão de dados da rede. Observa-se, no caso sob análise, que a rede com  $tx\_rate = 1$  Mb/s não consegue escalonar o conjunto de variáveis com um período mínimo de 5 ms, independente do algoritmo utilizado (RM ou EDF). Exceto para uma condição mostrada na Tabela A.14 ( $T_1 = 5$ ms,  $VAR\_AP = 126$  bytes), todos os casos simulados serão escalonáveis se a taxa de transmissão da rede WorldFIP for igual a 2,5 Mb/s.
2. O período da variável de maior frequência produzida em cada estação ( $T_1$ ) exerce uma forte influência no tempo de resposta. A consistência desses resultados pode ser inferida pela análise do método de cálculo da interferência causada pelo tráfego periódico, efetuado pela equação (4.8).
3. Na maior parte dos casos, os tempos de resposta são decrescentes em relação à posição lógica da estação na rede, para cada valor atribuído a  $T_1$ . Ou seja, de uma maneira geral as variáveis produzidas pela estação A000 têm tempos de resposta superiores às demais estações. Isto tem como causa o critério de ordenação das variáveis pelo algoritmo de escalonamento empregado. Deve ainda ser observado que, em função do algoritmo de escalonamento adotado, poderão ocorrer “oscilações” de alguns resultados devido ao “jitter” de transmissão.

4. Há um compromisso entre o valor de  $VAR\_AP$ , a largura da janela aperiódica e os tempos de resposta calculados. Os melhores resultados foram obtidos utilizando-se estruturas de dados de 24 bytes. Adicionalmente pode ser verificado que a atribuição de valores muito pequenos para  $VAR\_AP$ , ou aproximando-se do limite superior admissível para este parâmetro (126 bytes), reduz a escalabilidade do conjunto de variáveis e aumenta o tempo de resposta da rede.
5. O algoritmo EDF tende a produzir tempos de resposta iguais ou melhores que aqueles produzidos pelo algoritmo RM. Os resultados obtidos nas simulações mostram que quanto maior for a dificuldade de escalonamento pelo algoritmo RM maiores serão as diferenças entre os valores produzidos por este algoritmo e os produzidos pelo algoritmo EDF, com vantagem para este último. Além disso, é muito importante frisar que, caso sejam adotados determinados valores para os parâmetros  $T_1$  ou  $VAR\_AP$ , o algoritmo EDF consegue escalonar o conjunto de variáveis que, nas mesmas condições, não é escalonável pelo algoritmo RM.

### 6.4.3 Apresentação Gráfica dos Resultados

A título de ilustração, a seguir são apresentados graficamente os tempos de resposta encontrados para variáveis aperiódicas produzidas pelas estações A000 e A007, em função das larguras das estruturas de dados enviadas da camada de aplicação para a camada de enlace da rede.

Em cada figura são superpostos os resultados obtidos a partir dos algoritmos de escalonamento adotados, RM ou EDF. As Figuras 6.2 a 6.5 correspondem à rede configurada para operar com  $tx\_rate = 1\text{ Mb/s}$  e período  $T_1 = 10$  e 20ms. Os tempos de resposta encontrados considerando-se a taxa de transmissão de 2,5 Mb/s e  $T_1 = 5, 10$  e 20ms são mostrados nas Figuras 6.6 a 6.11.

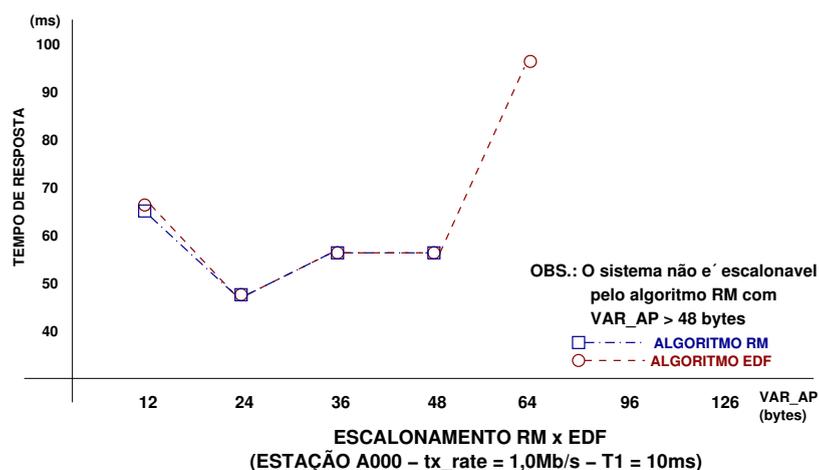
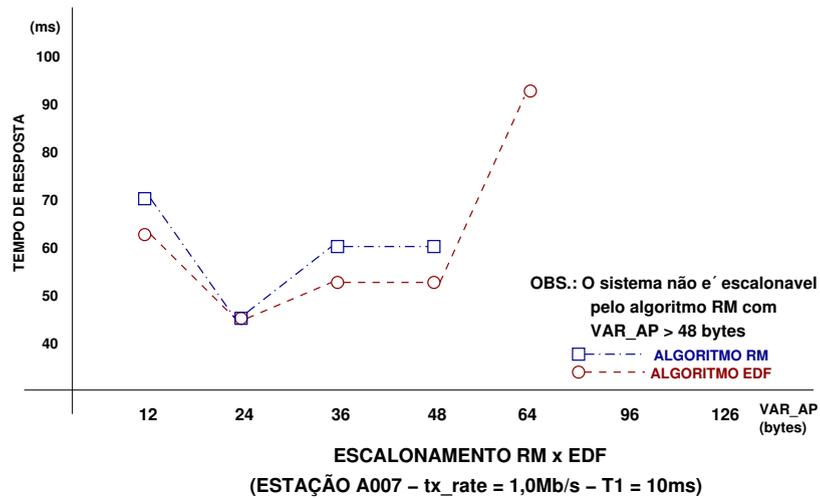
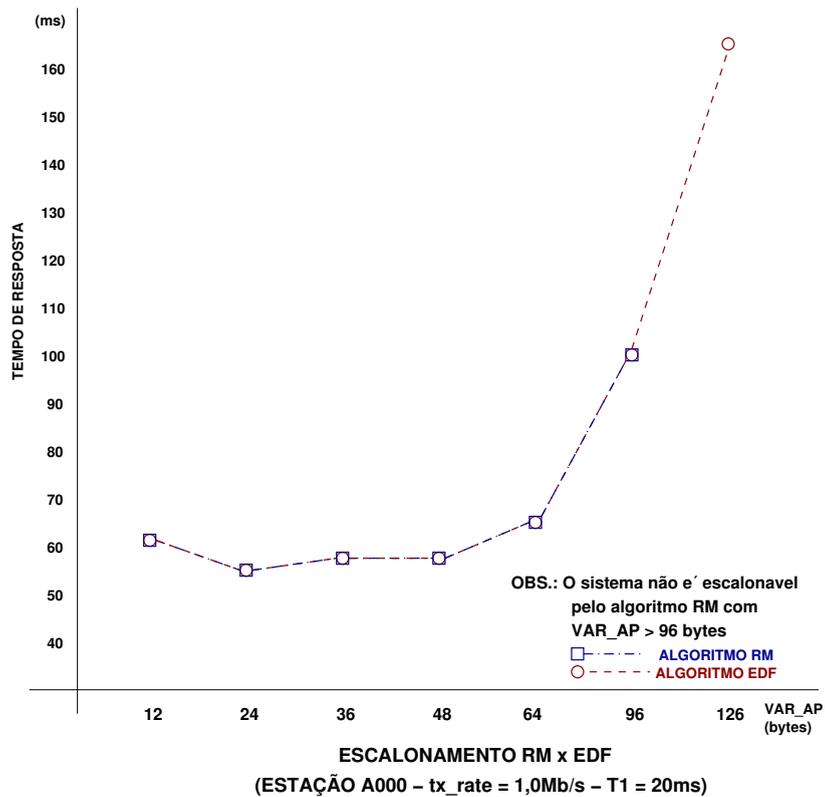


Figura 6.2: Tempos de Resposta - RM × EDF - Estação A000 (1,0 Mb/s -  $T_1 = 10\text{ms}$ )

Figura 6.3: Tempos de Resposta - RM × EDF - Estação A007 (1,0 Mb/s - T<sub>1</sub> = 10ms)Figura 6.4: Tempos de Resposta - RM × EDF - Estação A000 (1,0 Mb/s - T<sub>1</sub> = 20ms)

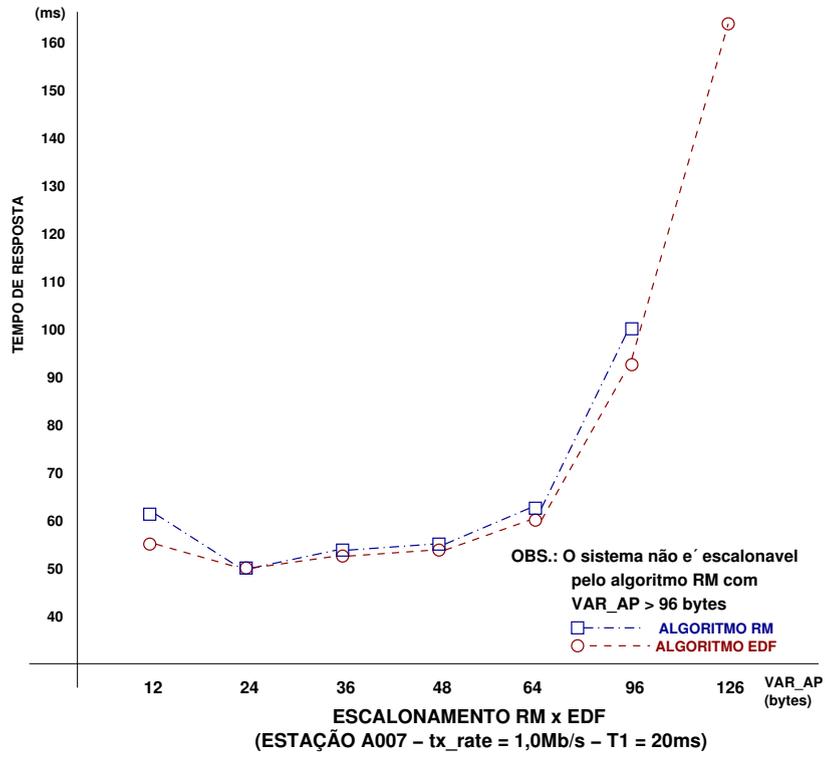


Figura 6.5: Tempos de Resposta - RM × EDF - Estação A007 (1,0 Mb/s -  $T_1 = 20ms$ )

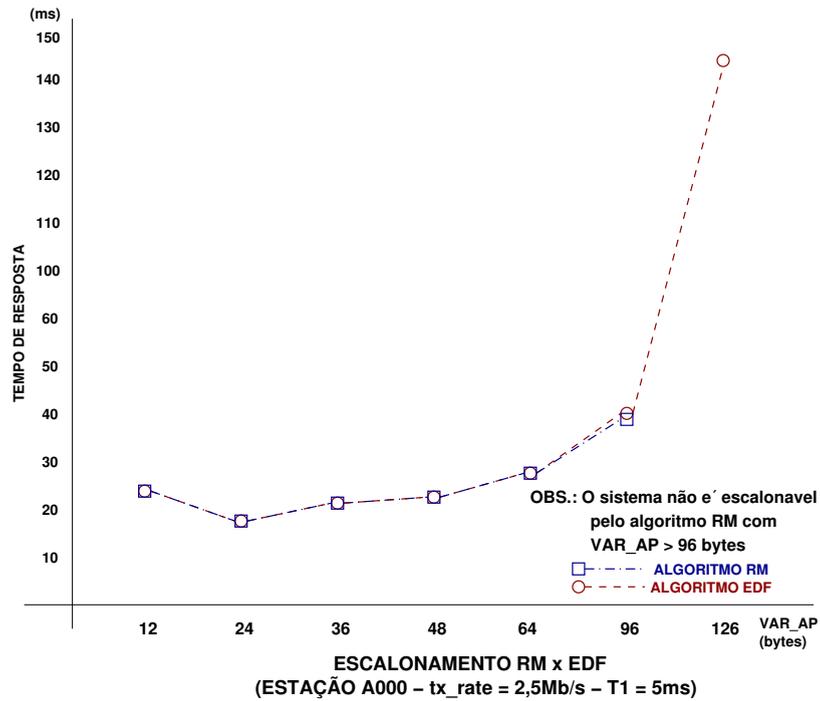
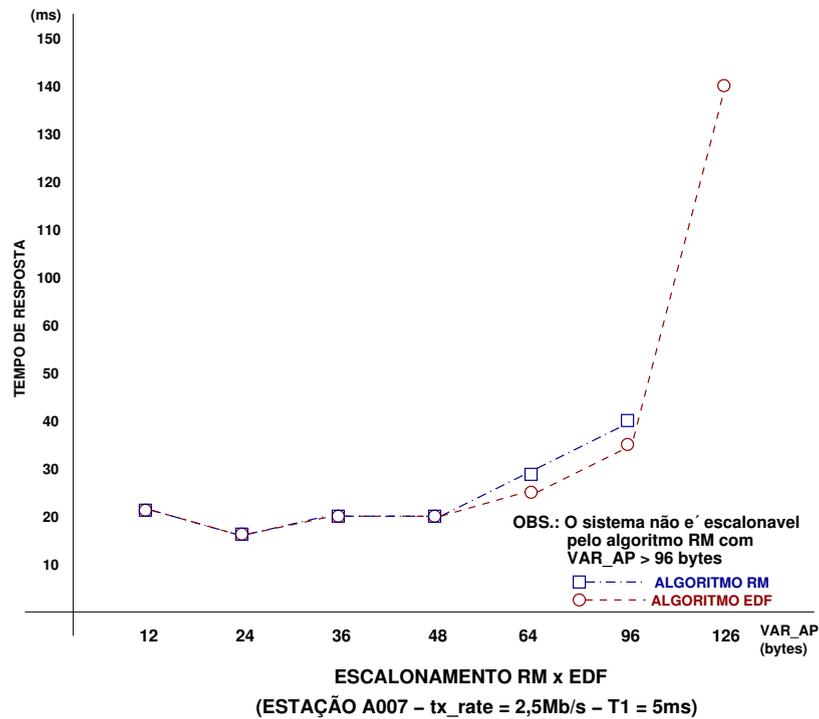
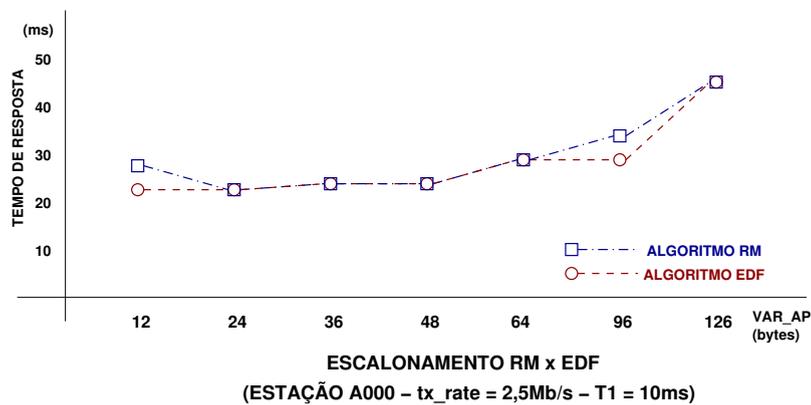
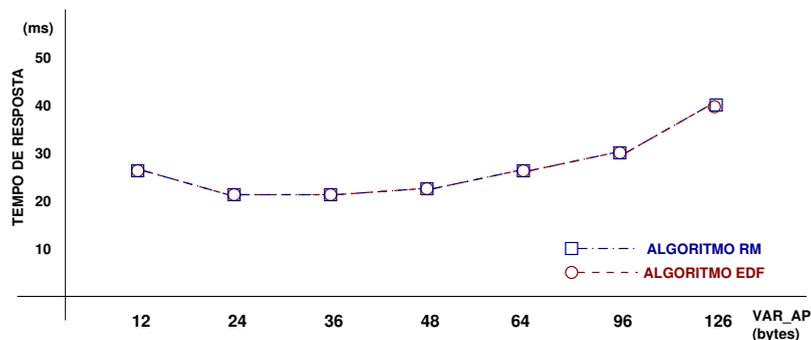
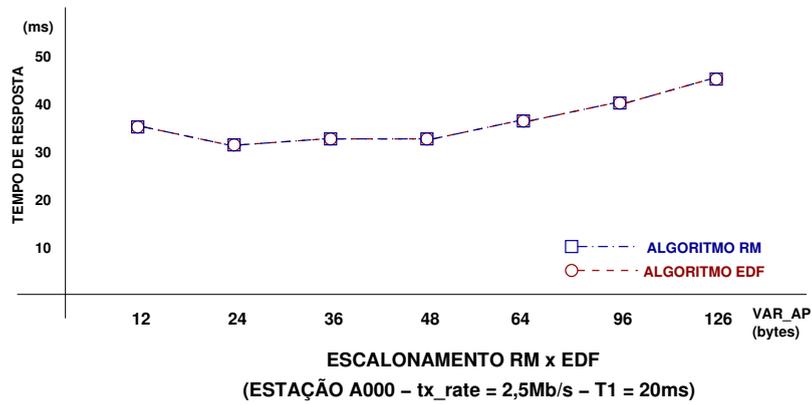
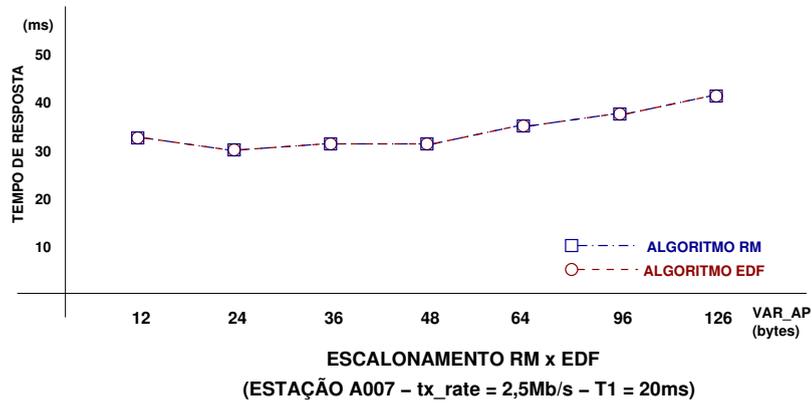


Figura 6.6: Tempos de Resposta - RM × EDF - Estação A000 (2,5 Mb/s -  $T_1 = 5ms$ )

Figura 6.7: Tempos de Resposta - RM × EDF - Estação A007 (2,5 Mb/s - T<sub>1</sub> = 5ms)Figura 6.8: Tempos de Resposta - RM × EDF - Estação A000 (2,5 Mb/s - T<sub>1</sub> = 10ms)Figura 6.9: Tempos de Resposta - RM × EDF - Estação A007 (2,5 Mb/s - T<sub>1</sub> = 10ms)

Figura 6.10: Tempos de Resposta - RM × EDF - Estação A000 (2,5 Mb/s - T<sub>1</sub> = 20ms)Figura 6.11: Tempos de Resposta - RM × EDF - Estação A007 (2,5 Mb/s - T<sub>1</sub> = 20ms)

## Capítulo 7

### Conclusões

O limite superior do tempo de resposta no atendimento de requisições de transmissão de variáveis aperiódicas por uma rede WorldFIP empregada em sistemas de controle de processos pode ser determinado a partir da aplicação dos conceitos de sistemas de tempo real encontrados na literatura.

O controle do acesso ao meio adotado pelo protocolo WorldFIP é efetuado com base no escalonamento executivo cíclico, cuja grade de tempo de liberação de variáveis periódicas, que serve como referência para montagem da tabela do árbitro do barramento, é definida em tempo de projeto. Uma das grandes dificuldades de implementação de sistemas baseados no algoritmo executivo cíclico é a construção dessa grade, cuja complexidade aumenta exponencialmente em relação ao número de tarefas que disputam o direito de acesso ao recurso compartilhado. Por esta razão são empregadas heurísticas sub-ótimas como ferramentas de análise e construção da grade de tempo.

A metodologia de cálculo desenvolvida no Capítulo 4 pode ser aplicada para análise da escalonabilidade de uma rede WorldFIP configurada para operação no modo síncrono. São disponibilizadas duas alternativas de construção da tabela do árbitro do barramento: uma baseada no algoritmo taxa monotônica (*rate monotonic* - RM) e a outra que utiliza o algoritmo *earliest deadline first* (EDF). É apresentado, também, um conjunto de equações utilizadas para compor o algoritmo de cálculo do tempo de resposta de pior caso no atendimento de requisições de transmissão de variáveis aperiódicas.

O objeto de aplicação da metodologia apresentada é o sistema de supervisão e controle de uma unidade geradora de energia elétrica de grande porte. Por esta razão no Capítulo 5 são descritas, de maneira sucinta, algumas características técnicas dos equipamentos principais e auxiliares normalmente encontrados em uma planta de geração de energia elétrica. Estes equipamentos são os

principais produtores e consumidores de dados processados pelos sistemas de supervisão e controle típicos de Usinas Hidrelétricas.

Um estudo de caso é então apresentado como exemplo de aplicação do método apresentado, usando os dados coletados durante a fase preliminar do projeto do sistema de supervisão e controle das duas unidades que estão sendo instaladas na Usina Hidrelétrica de Itaipu. Com base nos resultados mostrados nas tabelas A.1 até A.14 (Anexo A) podem ser feitas as seguintes observações:

1. O principal fator de influência no tempo de resposta das variáveis aperiódicas é a taxa de transmissão de dados da rede. Dentre os valores de  $tx\_rate$  considerados no estudo quase todos os casos analisados foram escalonáveis com a taxa de 2,5 Mb/s. Uma rede com  $tx\_rate = 1$  Mb/s não consegue escalonar o conjunto de variáveis com um período ( $T_1$ ) de 5 ms.
2. O período da variável de maior frequência produzida em cada estação ( $T_1$ ) tem uma forte influência no tempo de resposta para atendimento de requisições de transações aperiódicas.
3. Os tempos de resposta são decrescentes em relação à posição lógica da estação na rede, para cada valor atribuído a  $T_1$ . Isto deve-se ao critério de ordenação das variáveis pelo algoritmo de escalonamento. Deve ser observado, entretanto, que em função do algoritmo de escalonamento adotado poderá ocorrer a inversão de alguns resultados devido ao “jitter” gerado.
4. Há um compromisso entre o valor de  $VAR\_AP$  e a largura da janela aperiódica, para fins de escalonamento de variáveis. Foram efetuadas simulações com estruturas de dados de 12, 24, 36, 48, 64, 96 e 126 bytes. Os melhores tempos de resposta foram obtidos, para cada valor de  $tx\_rate$  adotado, com  $VAR\_AP = 24$  bytes. Foi também verificado que a atribuição de valores muito pequenos para  $VAR\_AP$ , além de reduzir a eficiência na utilização da rede, reduz também a escalonabilidade do conjunto de variáveis em função do aumento excessivo do tráfego pela rede.
5. O algoritmo EDF tende a produzir tempos de resposta melhores que aqueles produzidos pelo algoritmo RM. Os resultados obtidos nas simulações mostram que quanto maior for a dificuldade de escalonamento pelo algoritmo RM maiores serão as diferenças entre os valores produzidos por este algoritmo e os produzidos pelo algoritmo EDF com vantagem para este último. Além disso existem casos que, em função dos valores de certos atributos, o conjunto de variáveis não é escalonável pelo algoritmo RM e o é pelo algoritmo EDF.

O método de cálculo do tempo de resposta da rede WorldFIP, conforme foi apresentada nesta Dissertação, poderá ser refinado de forma a melhorar a previsibilidade do comportamento e a eficiência na utilização deste recurso. Por exemplo, pode ser considerada a extensão do modelo nos seguintes pontos:

- Explicitar a relação existente entre o número de variáveis periódicas produzidas por uma estação e a duração da interferência aperiódica. Esta relação, de natureza inversa, é função da quantidade de variáveis produzidas por determinada estação subscritora da rede e escalonadas para transmissão em um mesmo ciclo elementar.
- Desenvolver uma equação para o cálculo da duração de transações de mensagens. Assim será possível aumentar a largura do objeto transportado pela PDU para 254 bytes, valor duas vezes maior que o limite de 126 bytes imposto pelo protocolo para transações de variáveis identificadas.

Há ainda outros pontos importantes a serem abordados em futuros trabalhos, tais como:

- Inclusão da influência de falhas de componentes do suporte de comunicação no modelo matemático da rede.
- Execução de testes da rede WorldFIP em plataforma com vistas à validação final do modelo desenvolvido, em particular quando for adotado o algoritmo de escalonamento EDF para construção da tabela do árbitro do barramento.

## Apêndice A

# Tempos de Resposta da Rede WorldFIP

Nas seções A.1 e A.2 deste Apêndice encontram-se os resultados das simulações realizadas com suporte do programa ANAFIP (subseção 6.4.1). As Tabelas A.1 a A.7 contém os tempos de resposta da rede WorldFIP configurada para funcionar com taxa de transmissão de 1,0 Mb/s. As Tabelas A.8 a A.14 mostram os valores encontrados considerando-se a rede operando com  $tx\_rate=2,5$  Mb/s.

### A.1 Rede com Taxa de Transmissão de 1 Mb/s

#### A.1.1 Resultados com VAR\_AP = 12 bytes

Estação Produtora	Tempos de Resposta (ms)					
	$T_i = 5$ ms		$T_i = 10$ ms		$T_i = 20$ ms	
	RM	EDF	RM	EDF	RM	EDF
A000	ne	ne	65,58	66,10	61,88	61,88
A001	ne	ne	64,52	65,05	60,82	60,82
A002	ne	ne	64,26	64,78	60,56	60,56
A003	ne	ne	65,84	63,73	60,82	58,71
A004	ne	ne	66,10	63,46	60,56	58,45
A005	ne	ne	66,10	63,20	60,30	58,18
A006	ne	ne	69,77	62,67	61,85	57,66
A007	ne	ne	69,50	62,41	61,06	56,86

obs.: **ne** - não escalonável

Tabela A.1: Tempos de Resposta - RM  $\times$  EDF ( $tx\_rate = 1$  Mb/s - VAR\_AP = 12 bytes)

**A.1.2 Resultados com VAR\_AP = 24 bytes**

Estação Produtora	Tempos de Resposta (ms)					
	$T_i = 5$ ms		$T_i = 10$ ms		$T_i = 20$ ms	
	RM	EDF	RM	EDF	RM	EDF
A000	ne	ne	48,72	48,72	55,12	55,12
A001	ne	ne	48,00	48,00	54,40	54,40
A002	ne	ne	47,64	47,74	54,04	54,04
A003	ne	ne	47,64	46,92	52,60	52,60
A004	ne	ne	47,28	46,56	52,24	52,24
A005	ne	ne	46,92	46,20	51,88	51,88
A006	ne	ne	46,56	45,84	51,52	51,52
A007	ne	ne	46,56	45,48	50,80	50,80

obs.: **ne** - não escalonável

Tabela A.2: Tempos de Resposta - RM  $\times$  EDF ( $tx\_rate = 1$  Mb/s - VAR\_AP = 24 bytes)

**A.1.3 Resultados com VAR\_AP = 36 bytes**

Estação Produtora	Tempos de Resposta (ms)					
	$T_i = 5$ ms		$T_i = 10$ ms		$T_i = 20$ ms	
	RM	EDF	RM	EDF	RM	EDF
A000	ne	ne	56,87	56,87	57,33	57,33
A001	ne	ne	55,96	55,96	56,42	57,42
A002	ne	ne	55,50	55,50	55,96	55,05
A003	ne	ne	55,50	54,59	55,50	54,59
A004	ne	ne	55,49	54,14	55,05	54,14
A005	ne	ne	54,59	53,68	54,59	53,68
A006	ne	ne	54,12	53,22	54,14	53,22
A007	ne	ne	59,12	52,77	54,56	52,31

obs.: **ne** - não escalonável

Tabela A.3: Tempos de Resposta - RM  $\times$  EDF ( $tx\_rate = 1$  Mb/s - VAR\_AP = 36 bytes)

**A.1.4 Resultados com VAR\_AP = 48 bytes**

Estação Produtora	Tempos de Resposta (ms)					
	$T_i = 5$ ms		$T_i = 10$ ms		$T_i = 20$ ms	
	RM	EDF	RM	EDF	RM	EDF
A000	ne	ne	56,56	56,56	57,66	57,66
A001	ne	ne	56,01	56,01	57,11	57,11
A002	ne	ne	55,46	55,46	56,56	56,56
A003	ne	ne	56,01	54,90	55,56	55,46
A004	ne	ne	55,46	54,35	56,01	54,90
A005	ne	ne	54,90	53,80	55,46	54,35
A006	ne	ne	54,35	53,25	54,35	53,80
A007	ne	ne	58,83	52,70	54,97	52,97

obs.: **ne** - não escalonável

Tabela A.4: Tempos de Resposta - RM  $\times$  EDF ( $tx\_rate = 1$  Mb/s - VAR\_AP = 48 bytes)

**A.1.5 Resultados com VAR\_AP = 64 bytes**

Estação Produtora	Tempos de Resposta (ms)					
	$T_i = 5$ ms		$T_i = 10$ ms		$T_i = 20$ ms	
	RM	EDF	RM	EDF	RM	EDF
A000	ne	ne	ne	97,00	64,28	64,28
A001	ne	ne	ne	96,32	63,60	63,60
A002	ne	ne	ne	95,64	62,92	62,92
A003	ne	ne	ne	94,96	62,92	61,56
A004	ne	ne	ne	94,28	62,24	60,88
A005	ne	ne	ne	93,60	62,24	60,20
A006	ne	ne	ne	92,92	64,08	59,52
A007	ne	ne	ne	92,24	63,40	58,18

obs.: **ne** - não escalonável

Tabela A.5: Tempos de Resposta - RM  $\times$  EDF ( $tx\_rate = 1$  Mb/s - VAR\_AP = 64 bytes)

**A.1.6 Resultados com VAR\_AP = 96 bytes**

Estação Produtora	Tempos de Resposta (ms)					
	$T_i = 5$ ms		$T_i = 10$ ms		$T_i = 20$ ms	
	RM	EDF	RM	EDF	RM	EDF
A000	ne	ne	ne	ne	101,23	101,23
A001	ne	ne	ne	ne	100,30	100,30
A002	ne	ne	ne	ne	100,30	99,36
A003	ne	ne	ne	ne	100,30	97,49
A004	ne	ne	ne	ne	100,30	96,55
A005	ne	ne	ne	ne	104,68	95,62
A006	ne	ne	ne	ne	103,74	94,68
A007	ne	ne	ne	ne	101,87	92,81

obs.: **ne** - não escalonável

Tabela A.6: Tempos de Resposta - RM  $\times$  EDF ( $tx\_rate = 1$  Mb/s - VAR\_AP = 96 bytes)

**A.1.7 Resultados com VAR\_AP = 126 bytes**

Estação Produtora	Tempos de Resposta (ms)					
	$T_i = 5$ ms		$T_i = 10$ ms		$T_i = 20$ ms	
	RM	EDF	RM	EDF	RM	EDF
A000	ne	ne	ne	ne	ne	165,29
A001	ne	ne	ne	ne	ne	164,11
A002	ne	ne	ne	ne	ne	162,94
A003	ne	ne	ne	ne	ne	160,58
A004	ne	ne	ne	ne	ne	159,41
A005	ne	ne	ne	ne	ne	158,23
A006	ne	ne	ne	ne	ne	157,06
A007	ne	ne	ne	ne	ne	164,11

obs.: **ne** - não escalonável

Tabela A.7: Tempos de Resposta - RM  $\times$  EDF ( $tx\_rate = 1$  Mb/s - VAR\_AP = 126 bytes)

## A.2 Rede com Taxa de Transmissão de 2,5 Mb/s

### A.2.1 Resultados com VAR<sub>AP</sub> = 12 bytes

Estação Produtora	Tempos de Resposta (ms)					
	$T_i = 5$ ms		$T_i = 10$ ms		$T_i = 20$ ms	
	RM	EDF	RM	EDF	RM	EDF
A000	23,77	23,77	27,60	23,60	35,91	35,91
A001	23,34	23,34	27,18	27,18	35,49	35,49
A002	23,24	23,24	26,86	27,08	35,39	35,39
A003	23,13	22,81	26,65	26,65	34,65	34,65
A004	23,03	22,71	26,55	26,55	34,54	34,54
A005	22,92	22,60	26,44	26,44	34,33	34,44
A006	22,71	22,39	26,23	26,23	34,22	34,01
A007	22,81	22,29	26,02	26,13	33,91	33,91

Tabela A.8: Tempos de Resposta - RM  $\times$  EDF ( $tx\_rate = 2,5$  Mb/s - VAR<sub>AP</sub> = 12 bytes)

### A.2.2 Resultados com VAR<sub>AP</sub> = 24 bytes

Estação Produtora	Tempos de Resposta (ms)					
	$T_i = 5$ ms		$T_i = 10$ ms		$T_i = 20$ ms	
	RM	EDF	RM	EDF	RM	EDF
A000	18,93	18,93	22,38	22,38	32,38	32,38
A001	18,64	18,64	22,10	22,10	32,10	32,10
A002	18,50	18,50	21,95	21,95	31,95	31,95
A003	18,21	18,21	21,66	21,66	31,38	31,38
A004	18,06	18,06	21,52	21,52	31,23	31,23
A005	17,92	17,92	21,38	21,38	31,09	31,09
A006	17,78	17,78	21,23	21,23	30,80	30,94
A007	17,63	17,63	21,09	21,09	30,66	30,66

Tabela A.9: Tempos de Resposta - RM  $\times$  EDF ( $tx\_rate = 2,5$  Mb/s - VAR<sub>AP</sub> = 24 bytes)

**A.2.3 Resultados com VAR<sub>AP</sub> = 36 bytes**

Estação Produtora	Tempos de Resposta (ms)					
	$T_i = 5$ ms		$T_i = 10$ ms		$T_i = 20$ ms	
	RM	EDF	RM	EDF	RM	EDF
A000	21,93	21,93	23,13	23,13	33,13	33,13
A001	21,57	21,57	22,77	22,77	32,77	32,77
A002	21,20	21,38	22,59	22,59	32,57	32,40
A003	21,02	21,02	22,22	22,22	32,04	32,04
A004	20,84	20,84	22,04	22,04	31,86	31,86
A005	20,65	20,65	21,86	21,86	31,67	31,67
A006	20,47	20,47	21,67	21,67	31,49	31,49
A007	20,29	20,29	21,49	21,49	31,13	31,13

Tabela A.10: Tempos de Resposta - RM  $\times$  EDF ( $tx\_rate = 2,5$  Mb/s - VAR<sub>AP</sub> = 36 bytes)**A.2.4 Resultados com VAR<sub>AP</sub> = 48 bytes**

Estação Produtora	Tempos de Resposta (ms)					
	$T_i = 5$ ms		$T_i = 10$ ms		$T_i = 20$ ms	
	RM	EDF	RM	EDF	RM	EDF
A000	22,51	22,51	23,69	23,69	33,69	33,69
A001	22,29	22,29	23,47	23,47	33,47	33,47
A002	22,07	22,07	23,25	23,25	33,25	33,25
A003	21,85	21,85	23,03	23,03	32,81	32,81
A004	21,62	21,62	22,81	22,81	32,59	32,59
A005	21,40	21,40	22,59	22,59	32,37	32,37
A006	21,18	21,18	22,37	22,37	32,14	32,14
A007	20,96	20,96	22,14	22,14	31,70	31,70

Tabela A.11: Tempos de Resposta - RM  $\times$  EDF ( $tx\_rate = 2,5$  Mb/s - VAR<sub>AP</sub> = 48 bytes)

**A.2.5 Resultados com VAR\_AP = 64 bytes**

Estação Produtora	Tempos de Resposta (ms)					
	$T_i = 5$ ms		$T_i = 10$ ms		$T_i = 20$ ms	
	RM	EDF	RM	EDF	RM	EDF
A000	27,34	27,34	28,70	28,70	36,53	36,53
A001	27,07	27,07	28,43	28,43	36,26	36,26
A002	26,80	26,80	28,16	28,16	35,98	35,98
A003	27,07	26,53	27,88	27,88	35,44	35,44
A004	26,80	26,26	27,62	27,62	35,17	35,17
A005	26,53	25,98	27,34	27,34	34,90	34,90
A006	26,26	25,71	27,07	27,07	34,62	34,62
A007	28,26	25,44	26,80	26,80	34,08	34,08

Tabela A.12: Tempos de Resposta - RM  $\times$  EDF ( $tx\_rate = 2,5$  Mb/s - VAR\_AP = 64 bytes)**A.2.6 Resultados com VAR\_AP = 96 bytes**

Estação Produtora	Tempos de Resposta (ms)					
	$T_i = 5$ ms		$T_i = 10$ ms		$T_i = 20$ ms	
	RM	EDF	RM	EDF	RM	EDF
A000	38,99	39,36	33,48	30,48	40,48	40,48
A001	38,61	38,99	33,10	33,10	40,11	40,11
A002	38,24	38,61	32,73	32,73	39,36	39,73
A003	38,61	38,24	32,36	32,36	38,99	38,99
A004	38,61	37,86	31,98	31,98	38,61	38,61
A005	38,61	37,49	31,61	31,61	38,24	38,24
A006	40,99	37,11	31,23	31,23	37,49	37,86
A007	40,99	36,74	30,86	30,86	37,11	37,11

Tabela A.13: Tempos de Resposta - RM  $\times$  EDF ( $tx\_rate = 2,5$  Mb/s - VAR\_AP = 96 bytes)

**A.2.7 Resultados com VAR\_AP = 126 bytes**

Estação Produtora	Tempos de Resposta (ms)					
	$T_i = 5$ ms		$T_i = 10$ ms		$T_i = 20$ ms	
	RM	EDF	RM	EDF	RM	EDF
A000	ne	143,94	44,12	44,12	45,52	45,52
A001	ne	143,47	43,64	43,64	45,05	45,05
A002	ne	143,00	43,17	43,17	44,11	44,58
A003	ne	142,53	42,70	42,70	43,64	43,64
A004	ne	142,06	42,23	42,23	43,17	43,17
A005	ne	141,59	41,76	41,76	42,70	42,70
A006	ne	141,12	41,29	41,29	42,23	42,23
A007	ne	140,66	40,82	40,82	41,29	41,29

Tabela A.14: Tempos de Resposta - RM  $\times$  EDF ( $tx\_rate = 2,5$  Mb/s - VAR\_AP = 126 bytes)

## Apêndice B

# Códigos-Fonte dos Algoritmos Básicos

Nas seções a seguir estão disponibilizadas as funções que podem ser empregadas para determinação de alguns parâmetros utilizados na metodologia de cálculo apresentada no Capítulo 4. Estas funções estão codificadas em linguagem ANSI C para possibilitar a execução de testes ou mesmo sua integração a outros sistemas.

### B.1 Função `mdc_mmc.c`

```
/* mdc_mmc.c - calcula a largura do ciclo elementar e do macro-ciclo */

/* calcula: e = mdc(Ti) (largura do ciclo elementar) */
/*          m = mmc(Ti) (largura do macro-ciclo) */
/*          N = m/e (numero de ciclos elementares em um macro-ciclo) */
/* Ver subsecao 2.5.2 da Dissertacao */

/* os parametros das variaveis sao lidos do arquivo varPer.dat */

#include<stdio.h>
#define FALSE 0
#define TRUE 1
typedef int boolean;

/* modelo das variaveis periodicas - secao 4.2.1 da Dissertacao */

typedef struct {int k; float C; int T;} var;

/*
k -> produtor da variavel
C -> custo (ou duracao da transmissao
T -> periodo
*/

void leArqVar(FILE *arq, var v[100], int *n);
int calc_mdc(var v[100], int n);
int calc_mmc(var v[100], int mdc, int n, int *N);
```

```
/* teste do algoritmo */

int main()
{
    int i, e, min, np, m, N;
    var vp[100];
    FILE *arqEntrada;

    leArqVar(arqEntrada, vp, &np);
    e = calc_mdc(vp,np);
    m = calc_mmc(vp,e,np,&N);
    printf("\ne = MDC(Ti)=%d\n", e);
    printf("\nm = MMC(Ti) = %d\n N = %d\n", m, N);
    return(0);
}

/* funcao leArqVar - le atrib. das variaveis periodicas (arq. varPer.dat) */
/* retorna um array de variaveis do tipo "var" e o num. de variaveis lidas */

void leArqVar(FILE *arq, var v[100], int *n)
{
    boolean aux;
    int i,s,b;
    float a;
    arq = fopen("varPer.dat", "r");

    if(arq == NULL)
    {
        printf("\nerro na abertura do arquivo varPer.dat\n");
        exit(1);
    }

    aux = TRUE;
    i=0;
    while(aux)
    {
        fscanf(arq, "%d\t %f\t %d\n", &s,&a,&b);
        v[i].k=s; /* estacao produtora da variavel */
        v[i].C=a; /* custo de transmissao da variavel */
        v[i].T=b; /* periodo da variavel */
        if(v[i].C ==0)
        {
            aux = FALSE;
            *n=i;
            fclose(arq);
        }
        i++;
    }
}
```

```
/* calc_mdc calcula o mdc dos periodos das variaveis periodicas */

int calc_mdc(var v[100], int n)
{
    int i, min, mdc;
    boolean aux;
    min=v[0].T;          /* inicializa variavel auxiliar */

    for(i=0; i<n; i++)  /* encontra o menor periodo da serie */
    {
        if(v[i].T<min)
            min=v[i].T;
    }

    mdc=min;
    if(!mdc)
    {
        printf("\n O MDC nao pode ser zero\n ");
        return(1);
    }

    aux=TRUE;
    i=0;
    while(aux)         /* encontra o mdc */
    {
        while(i<n)
        {
            if(v[i].T%mdc)
            {
                mdc=mdc-1;
                aux=TRUE;
                i=0;
            }
            else
            {
                i++;
                aux=FALSE;
            }
        }
    }
    return(mdc);
}
```

```
/* funcao calc_mmc -> calcula largura do macro-ciclo (e) e o num. de */
/*                                ciclos elementares (N) em um macro-ciclo */

int calc_mmc(var v[100], int mdc, int n, int *N)
{
    int i, min, max, mmc, M;
    boolean aux;

    max=v[0].T;          /* inicializa variavel auxiliar */

    for(i=0; i<n; i++)   /*encontra o menor periodo da serie */
    {
        if(v[i].T>max)
            max=v[i].T;
    }

    M=max;
    i=0;
    while(aux) /* encontra N (num. de ciclos elementares no macro-ciclo) */
    {
        while(i<n)
        {
            if(M%((v[i].T/mdc)))
            {
                M++;
                aux=TRUE;
                i=0;
            }
            else
            {
                i++;
                aux=FALSE;
            }
        }
    }

    mmc=M*mdc;
    *N=M;
    return(mmc);
}
```

**B.2 Função mrm.c**

```

/* mrm.c - gera a matriz de escalonamento mtpRM - subsecao 4.4.1 */

/* codigo para geracao da matriz de escalonamento do trafego
   periodico usando a abordagem Taxa Monotonica (RM) */

/* Parametros de entrada: - variaveis periodicas (arquivo varPer.dat)      */
/*                          - largura do ciclo elementar (e), numero de ciclos */
/*                          elementares no macro-ciclos (N) fornecidos pela */
/*                          funcao mdc_mmc.c (Anexo I)                       */

#include<stdio.h>
#define TRUE 1
#define FALSE 0
typedef int boolean;

/* modelo das variaveis periodicas - secao 4.2.1 */
typedef struct {int k; float C; int T;} var;
/*
   k -> produtor da variavel
   C -> custo (ou duracao da transmissao)
   T -> periodo
*/

/* prototipos das funcoes */
void leArq(FILE *arq, var v[100], int *n);
int escala_rm(var v[100], int np, int e, int N, boolean mtp[100][100]);
void gravaArq(FILE *arq, int n, int N, boolean a[100][100]);
void leArqMtp(FILE *arq, int n, int N, boolean a[100][100]);
void inicializa(int np, int N,boolean a[100][100]);

/* teste da implementacao do algoritmo */

int main(void)
{
  FILE *arqSaida, *arqEntrada, *arqMtp;
  boolean mtp[100][100], aux;
  float a;
  int b,i,np,r,s;
  var vp[100];
  int N, e;
  N=12;
  e=1;

  /* parametros e, N - fornecidos pela funcao mdc_mmc.c (Apêndice B) */

  leArq(arqEntrada, vp, &np);
  inicializa(np, N, mtp);
  r = escala_rm(vp, np, e, N, mtp);
  gravaArq(arqSaida, np, N, mtp);
  leArqMtp(arqMtp, np, N, mtp);
}

```

```

/* funcao escala_rm - produz a matriz de escalonamento mtpRM */
int escala_rm(var v[100], int np, int e, int N, boolean mtp[100][100])
{
    int i,j,k,l,aux;
    float w[100];
    for(i=0; i<=N; i++)
        w[i]=0;

    for(i=0; i<np; i++)
    {
        l=0;
        while(l<N)
        {
            if((w[l]+v[i].C)<=e)
            {
                mtp[i][l]=TRUE;      /* v[i] eh escalonada no ce atual */
                w[l]=w[l]+v[i].C;   /* atualiza a carga do ce */
                l=l+v[i].T/e;       /* avanca ateh a proxima chegada de v[i] */
            }
            else
            {
                k=l+1;              /* vai para proximo ce */
                aux=TRUE;
                while(aux)
                {
                    if((w[k]+v[i].C)<=e) /* se ce nao estah sobrecarregado */
                    {
                        mtp[i][k]=TRUE; /* escalona v[i] */
                        w[k]=w[k]+v[i].C; /* atualiza carga do ce */
                        l=l+v[i].T/e; /* avanca ateh a proxima chegada de v[i] */
                        aux=FALSE;
                    }
                    else
                    {
                        if((k*e)>(l+v[i].T/e))
                        {
                            printf("\n\n\tv[%d] nao eh escalonavel\n", i+1);
                            printf("\ta matriz de escalonamento nao eh valida\n\n");
                            aux=FALSE;
                            return(1);
                            l=N+1; /* forca saida do loop de ciclos elementares */
                            i=np+1; /* forca saida do loop de variaveis */
                        }
                        else
                        {
                            k=k+1;
                            aux=TRUE;
                        }
                    }
                }
            }
        }
    }
    return(0);
}

```

```
/* funcao leArq - le atrib. das variaveis periodicas (arq. varPer.dat) */  
/* retorna um array de variaveis do tipo "var" e num. de variaveis lidas */
```

```
void leArq(FILE *arq, var v[100], int *n)  
{  
    boolean aux;  
    int i,s,b;  
    float a;  
    arq = fopen("varPer.dat", "r");  
    if(arq == NULL)  
    {  
        printf("\nerro na abertura do arquivo varPer.dat\n");  
        exit(1);  
    }  
    aux = TRUE;  
    i=0;  
    while(aux)  
    {  
        fscanf(arq, "%d\t%f\t%d\n", &s,&a,&b);  
        v[i].k=s; /* estacao produtora da variavel */  
        v[i].C=a; /* custo de transmissao da variavel */  
        v[i].T=b; /* periodo da variavel */  
        if(v[i].C ==0)  
        {  
            aux = FALSE;  
            *n=i;  
            fclose(arq);  
        }  
        i++;  
    }  
}
```

```
void gravaArq(FILE *arq, int n, int N, boolean a[100][100])  
{  
    int i,j,r;  
  
    arq = fopen("mtpRM.dat", "w");  
    if(arq == NULL)  
    {  
        printf("\nErro na abertura do arquivo mtpRM.dat\n");  
        exit(1);  
    }  
    for(i=0; i<n; i++)  
        for(j=0; j<N; j++)  
        {  
            r=a[i][j];  
            fwrite(&r,sizeof(int),1,arq);  
        }  
    fclose(arq);  
}
```

```
void leArqMtp(FILE *arq, int n, int N, boolean a[100][100])
{
    int i,j,r;

    arq = fopen("mtpRM.dat", "r");
    if(arq == NULL)
    {
        printf("\nErro na abertura do arquivo mtpRM.dat\n");
        exit(1);
    }
    for(i=0; i<n; i++)
        for(j=0; j<N; j++)
            {
                fread(&r,sizeof(int),1,arq);
                a[i][j]=r;
            }
        fclose(arq);

    /* teste dos valores lidos */
    for(i=0; i<n; i++)
    {
        printf("vp[%d]\t", i+1);
        for(j=0; j<N; j++)
            printf("%d ", a[i][j]);
        printf("\n");
    }
}

void inicializa(int np, int N, boolean a[100][100])
{
    int i,j;
    for(i=0; i<np; i++)
        for(j=0; j<=N; j++)
            a[i][j]=0;
}
```

### B.3 Função medf.c

```

/* medf.c - gera matriz de escalonamento mtpEDF.dat (subsecao 4.4.2) */
/* codigo para geracao da matriz de escalonamento do trafego */
/* periodico usando a abordagem Earliest Deadline First (EDF) */

#include<stdio.h>
#define TRUE 1
#define FALSE 0
#define MAX 1000
typedef int boolean;
/* modelo das variaveis periodicas - secao 4.2.1 */
typedef struct {int k; float C; int T;} var;
/*      k -> produtor da variavel
      C -> custo (ou duracao da transmissao)
      T -> periodo */

/* atributos da requisicao de transmissao da variavel */
typedef struct {boolean req; int prazo; int ciclo;} cand;

/* Param. de entrada: - variaveis periodicas (arquivo varPer.dat) */
/*                    - largura do ciclo elementar (e), numero de ciclos */
/*                    elementares no macro-ciclos (N) fornecidos pela */
/*                    funcao mdc_mmc.c (Anexo I) */

struct rec {int k,a,b;}; /* estrutura dos registros do arquivo mtpEDF.dat */

/* prototipos das funcoes */
void leArq(FILE *arq, var v[100], int *n);
int escala_edf(var v[100], int np, int e, int N, boolean mtp[100][100],
              int prio[100][100]);
void gravaArq(FILE *arq, int n, int N, boolean a[100][100],
              int prio[100][100], struct rec z);
void inicializa(int n, int N, boolean a[100][100],int p[100][100]);
void leArqMtp(FILE *arq, int n, int N, boolean a[100][100],
              int p[100][100]);

/* teste da implementacao do algoritmo EDF */
int main(void)
{
    FILE *arqSaida, *arqEntrada;
    boolean mtp[100][100], aux;
    float a;
    int b,i,np,r,s, prio[100][100];
    var vp[100];
    struct rec z;

    /* parametros e, N - fornecidos pela funcao mdc_mmc.c (Apêndice B) */
    int e, N;
    e=1; /* valor usado para teste do algoritmo */
    N=12; /* valor usado para teste do algoritmo */

    leArq(arqEntrada, vp, &np);
    inicializa(np,N,mtp,prio);
    r = escala_edf(vp,np,e,N,mtp,prio);

```

```

    gravaArq(arqSaida, np, N, mtp, prio, z);
    leArqMtp(arqEntrada, np, N, mtp, prio);
}

/* funcao escala_edf - produz matriz mtpEDF - subsecao 4.4.2 */

int escala_edf(var v[100], int np, int e, int N, boolean mtp[100][100],
               int prio[100][100])
{
    cand tr[100];
    float w[100];
    boolean aux, carga, davez, reqpend;
    int i, p, edf, ce, prox;

    edf = e;      /* inicializa o prazo mais "proximo" */
    ce=0;

    /* inicializacao das variaveis w[] e tr[] */

    for (i=0;i<20;i++)
    {
        w[i]=0;
        tr[i].req=0;
        tr[i].prazo=MAX;
        tr[i].ciclo=0;
    }

    aux = TRUE;
    while((ce < N) && aux)
    {
        p=0; /* pos. (slot) de tempo dentro do ciclo elementar */
        /* determina os instantes de chegada de cada requisicao */
        i=0;
        while((i<np) && aux)
        {
            if(!(ce % v[i].T/e)) /* verifica chegada vp[i] no ce corrente */
            {
                if(tr[i].req) /* se ha requisicao pendente */
                {
                    printf("\nvp[%d] nao eh escalonavel pelo algoritmo EDF\n\n", i);
                    printf("a matriz de escalonamento mtp_EDF nao eh valida\n");
                    aux = FALSE;
                }
                else
                {
                    tr[i].req = TRUE; /* requisicao de vp[i] */
                    tr[i].prazo = ce + v[i].T/e; /* prazo de vp[i] */
                    tr[i].ciclo = ce; /* ciclo de chegada da req de vp[i] */
                }
            }
            i++;
        }
    }
}

```

```

/* escalonamento de variaveis no ciclo elementar ce */
carga = FALSE; /* inicializacao do carregamento do ce */
while(!carga) /* escolha da variavel com menor prazo */
{
    reqpend = FALSE;
    edf = MAX;
    davez = TRUE; /* variavel a ser escalonada no ce corrente */
    for(i=0; i<np; i++)
    {
        if(tr[i].req) /* se ha requisicao de vp[i] */
        {
            reqpend = TRUE;
            if(tr[i].prazo <= edf)
            {
                if(davez) /* proxima variavel a ser escalonada */
                {
                    edf = tr[i].prazo;
                    prox = i;
                    davez = FALSE;
                }
                else /* requisicao mais antiga -> maior prioridade */
                {
                    if(tr[i].ciclo < tr[prox].ciclo)
                    {
                        edf = tr[i].prazo;
                        prox = i;
                    }
                }
            }
        }
    }
}
/* carrega proxima variavel a ser escalonada no ce corrente */
if((w[ce]+v[prox].C<=e) && reqpend)/* testa se ce esta sobrecarregado */
{
    mtp[prox][ce]=TRUE;
    w[ce] = w[ce] + v[prox].C;
    tr[prox].req = FALSE;
    tr[prox].prazo = MAX;
    tr[prox].ciclo = 0;
    p=p+1;
    prio[prox][ce]=p; /* pos. de atendimento de vp[i] escolhida */
}
else
    carga = TRUE; /* ce corrente estah sobrecarregado */
}
ce++; /* vai para o prox ce */
}
}

/* funcao leArq - le atributos das variaveis periodicas (arq. varPer.dat) */
/* retorna um array de variaveis do tipo "var" e num. de variaveis lidas */

```

```
void leArq(FILE *arq, var v[100], int *n)
{
    boolean aux;
    int i,s,b;
    float a;
    arq = fopen("varPer.dat", "r");
    if(arq == NULL)
    {
        printf("\nerro na abertura do arquivo varPer.dat\n");
        exit(1);
    }
    aux = TRUE;
    i=0;
    while(aux)
    {
        fscanf(arq, "%d\t%f\t%d\n", &s,&a,&b);
        v[i].k=s; /* estacao produtora da variavel */
        v[i].C=a; /* custo de transmissao da variavel */
        v[i].T=b; /* periodo da variavel */
        if(v[i].C ==0)
        {
            aux = FALSE;
            *n=i;
            fclose(arq);
        }
        i++;
    }
}

void inicializa(int n, int N, boolean a[100][100],int p[100][100])
{
    int i,j;
    for(i=0; i<n; i++)
        for(j=0; j<N; j++)
        {
            a[i][j]=0;
            p[i][j]=0;
        }
}

/* grava a matriz gerada no arquivo mtpEDF.dat */
void gravaArq(FILE *arq, int n, int N, boolean a[100][100],
              int prio[100][100], struct rec z)
{
    int i,j;
    arq = fopen("mtpEDF.dat", "w");
    if(arq == NULL)
    {
        printf("\nErro na abertura do arquivo mtpEDF.dat\n");
        exit(1);
    }
    for(i=0; i<n; i++)
    {
        for(j=0; j<N; j++)
        {
```

```
        z.k=i;
        z.a=a[i][j];
        z.b=prio[i][j];
        fwrite(&z,sizeof(struct rec),1,arq);
    }
}
fclose(arq);
}

/* teste de leitura do arquivo mtpEDF.dat */
void leArqMtp(FILE *arq, int n, int N, boolean a[100][100], int p[100][100])
{
    int i,j;
    struct rec z;
    arq = fopen("mtpEDF.dat", "r");
    if(arq == NULL)
    {
        printf("\nErro na abertura do arquivo mtpEDF.dat\n");
        exit(1);
    }
    for(i=0; i<n; i++)
    {
        for(j=0; j<N; j++)
        {
            fread(&z,sizeof(struct rec),1,arq);
            a[i][j]=z.a;
            p[i][j]=z.b;
        }
    }
    fclose(arq);
    /* teste da matriz lida */
    printf("\t\t ciclo elementar\n");
    printf("      ");
    for(i=1; i<=N; i++)
        printf("    %d ", i);
    printf("\n");
    printf("      ");
    for(i=1; i<=N; i++)
        printf("    a b");
    printf("\n");
    for(i=0; i<n; i++)
    {
        printf("v[%d]", i+1);

        for(j=0; j<N; j++)
            printf("    %d %d", a[i][j], p[i][j]);
        printf("\n");
    }
}
```

**B.4 Função jit.c**

```

/* jit.c */
/* calculo do jitter de transmissao das variaveis periodicas */
/* escalonadas pelos algoritmos RM ou EDF por inspecao das matrizes */
/* mtpRM ou mtpEDF (subsecoes 4.4.1 ou 4.4.2 da Dissertacao) */

/* dados de entrada: */
/* - np  -> numero de variaveis periodicas
   - N    -> numero de ciclos elementares contidos no macro-ciclo
   - mtp  -> elem. "a" das matrizes mtpRM ou mtpEDF (funcoes mrm.c ou medf.c)
   - p    -> elem. "b" da matriz mtpEDF (funcao medf.c)
   - modo -> disciplina de escalonamento: RM (modo=1) ou EDF (modo=2)
   - ceIni-> vetor para armazenamento do ce inicial do escalonamento de cada var.
   - jMax -> vetor para armazenamento do jitter maximo de cada variavel */

/* modelo das variaveis periodicas - secao 4.2.1 da Dissertacao */
typedef struct {int k; float C; int T;} var;
/* k -> produtor da variavel
   C -> custo (ou duracao da transmissao
   T -> periodo */

void jit(int np, int N, var vp[100], boolean mtp[100][100],
         int p[100][100], int modo, int ceIni[100], float jMax[100])
{
    int ce, reqPrim, reqProx, prox, nce, aux, i, j, k;
    float w, distPrim, distProx, jitter;
    boolean continua;

    for(i=0; i<np; i++)
        jMax[i]=0;

    for(i=0; i<np; i++)
    {
        ce=0;
        while (ce < N)
        {
            reqPrim=ce;
            prox=ce+vp[i].T/e;
            reqProx=prox%N;

            /* encontra a primeira liberacao de vp[i] */
            nce=0;
            while(mtp[i][reqPrim+nce]==FALSE)
                nce++;

            w=0;
            if(modo==1)
            {
                for(j=0; j<i; j++)
                {
                    if(mtp[j][reqPrim+nce])
                        w = w + vp[i].C;
                }
            }
        }
    }
}

```

```

if(modo==2)
{
  for(j=p[i][reqPrim+nce]-1; j>=1; j--)
  {
    continua=TRUE;
    k=0;
    while((k<np) && continua)
    {
      if(k!=i)
      {
        if(j==p[k][reqPrim+nce])
        {
          w = w + vp[k].C;
          continua=FALSE;
        }
      }
      k++;
    }
  }
}
distPrim=nce+w;
if(!ce)
  aux=nce;

/*encontra a proxima liberacao de vp[i] */
nce=0;
while(mtp[i][reqProx+nce]==FALSE)
  nce++;

w=0;
if(modo==1)
{
  for(j=0; j<i; j++)
  {
    if(mtp[j][reqProx+nce])
      w = w + vp[i].C;
  }
}
if(modo==2)
{
  for(j=p[i][reqProx+nce]-1; j>=1; j--)
  {
    continua=TRUE;
    k=0;
    while((k<np) && continua)
    {
      if(k!=i)
      {
        if(j==p[k][reqProx+nce])
        {
          w = w + vp[k].C;
          continua=FALSE;
        }
      }
      k++;
    }
  }
}

```

```
        }
    }
}
distProx=nce+w;
ce=prox;
/* encontra o jitter atual */
jitter = distProx - distPrim;
if(jitter > jMax[i])
{
    jMax[i] = jitter;
    ceIni[i] = reqProx+nce;
}
}
if(!jMax[i])
    ceIni[i]=aux;
}
}
```

## Apêndice C

# ANAFIP - Interface com o Usuário

Neste Apêndice são mostradas, como ilustração, algumas telas da interface com usuário do programa ANAFIP.

### C.1 Tela Principal

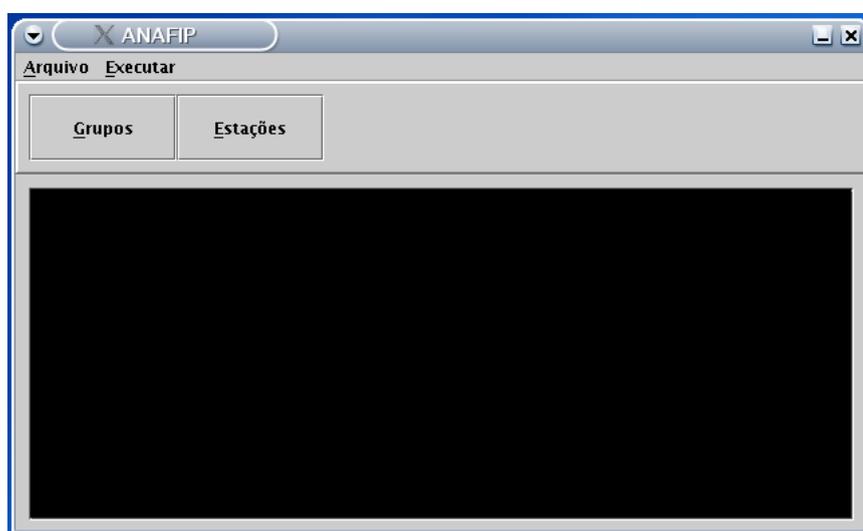


Figura C.1: ANAFIP - Tela principal

## C.2 Entrada de Parâmetros

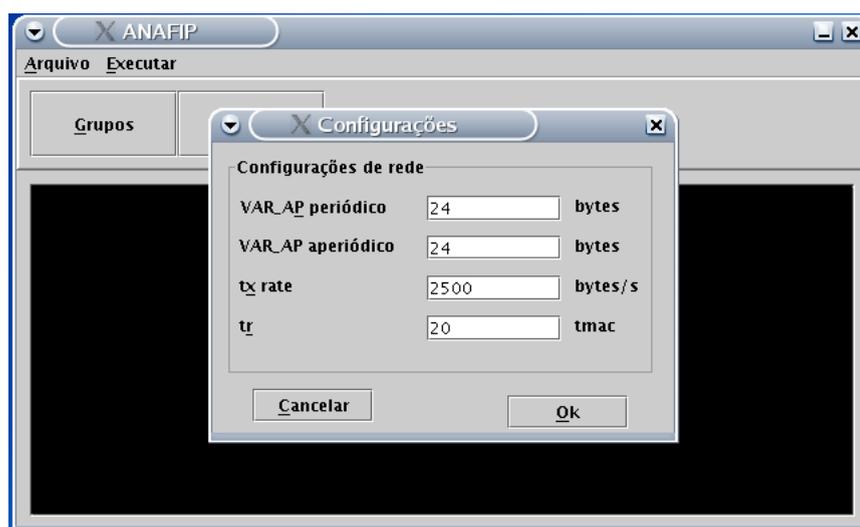


Figura C.2: ANAFIP - Entrada de Parâmetros

## C.3 Edição de Grupos

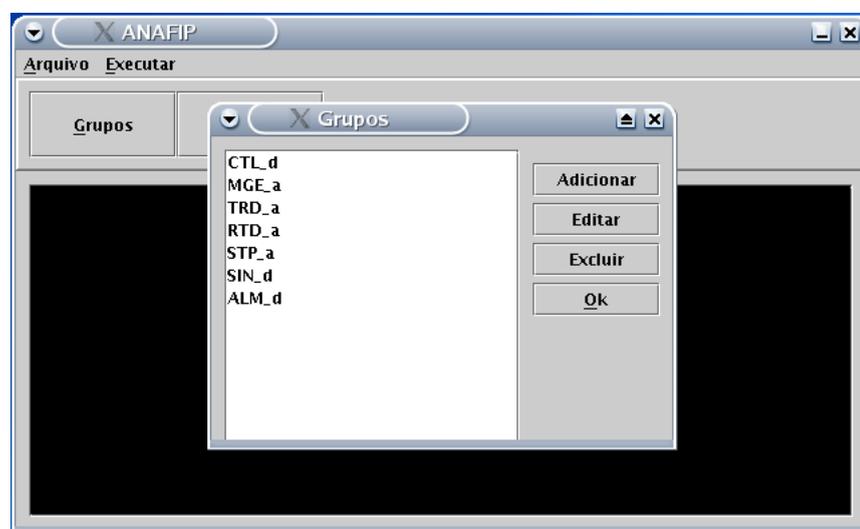


Figura C.3: ANAFIP - Edição de Grupos

## C.4 Edição de Estações

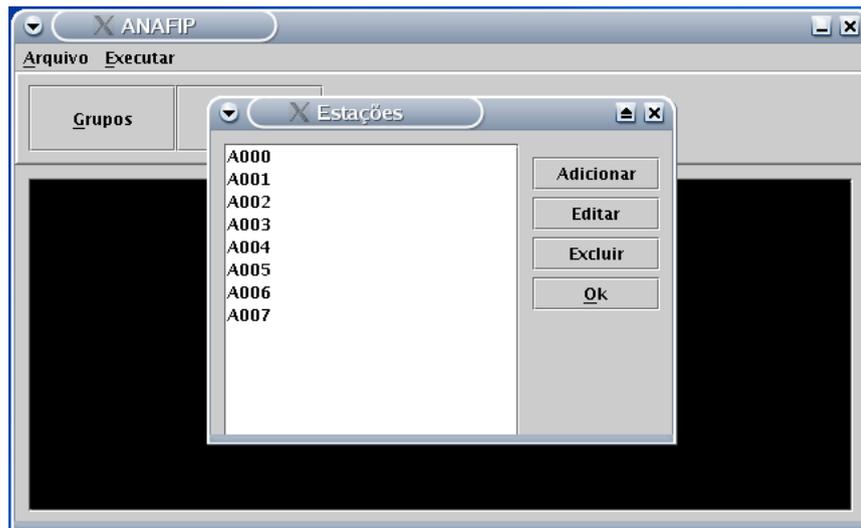


Figura C.4: ANAFIP - Edição de Estações

## C.5 Execução de Caso

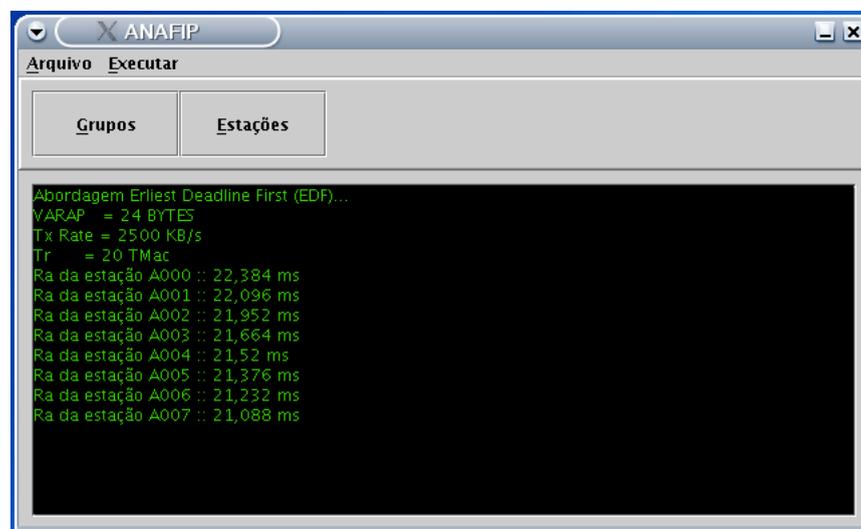


Figura C.5: ANAFIP - Execução de um Caso

# Referências Bibliográficas

- [1] ALMEIDA, Luís M.P. *Flexibility and Timeliness in Fieldbus-based Real-Time Systems*. Tese de Doutoramento, Universidade de Aveiro, Portugal, 1999.
- [2] ARAS, C. M.; KUROSE, J. F.; REEVES, D. S.; *et al.*. Real-time communications in packet-switched networks. In: *IEEE Proceedings*, volume 82. 1994 págs. 122–139.
- [3] AUDSLEY, N. *Deadline Monotonic Scheduling*. Rel. téc., Department of Computer Science, University of York, 1990.
- [4] AZEVEDO, Fernando R. H.; FILHO, Jorge M. O.; ALVES, José; *et al.*. *Análise da Aplicação de Tecnologia Digital a Controle de Processos de Usinas e Subestações*. 1<sup>a</sup> ed. Eletrobrás - Centrais Elétricas Brasileiras S.A., 1991.
- [5] BAKER, T.P.; SHAW, Alan. The cyclic executive model and ada. *IEEE Transactions in Software Engineering*, 1988. págs. 120–129.
- [6] BURNS, A. Scheduling hard real-time systems. In: *Special Issue on Real-Time Systems*. 1991 págs. 116–128.
- [7] BURNS, A.; AUDSLEY, N. *Real-Time Systems Schedulability*. Rel. téc., University of York, 1990.
- [8] BURNS, A.; WELLINGS, A. J. Criticality and utility in the next generation. *The Journal of Real-Time Systems*, 1991. volume 3, págs. 351–354.
- [9] BURNS, Alan; WELLINGS, Andy. *Real-Time Systems and Programming Languages*. 2<sup>a</sup> ed. Addison Wesley, 1997.
- [10] CARDEIRA, Carlos; MAMMERI, Zoubir. A schedulability analysis of tasks and network traffic in distributed real-time systems. *The Journal of the International Measurement Confederation IMEKO*, 1995. volume 2, n<sup>o</sup> 15, págs. 71–83.

- [11] COTRIM, John Reginald (Ed.). *Itaipu Hydroelectric Project - Engineering Features*. Brasil: Itaipu Binacional, 1994.
- [12] FARINES, Jean-Marie; FRAGA, Joni S.; DE OLIVEIRA, Rômulo Silva. *Sistemas de Tempo Real*. 1ª ed. São Paulo: IME-USP, Escola de Computação 2000, 2000.
- [13] FIDGE, C.J. *Real-Time Schedulability Tests for Preemptive Multitasking*. Rel. téc., Department of Computer Science, University of Queensland, 1996.
- [14] FITZGERALD, A.E.; KINGSLEY, Charles; KUSKO, Alexander. *Máquinas Elétricas - Conversão Eletromecânica da Energia - Processos, Dispositivos e Sistemas*. 1ª ed. McGraw-Hill do Brasil Ltda., 1975.
- [15] GOMAA, Hassan. *Designing Concurrent, Distributed, and Real-Time Applications with UML*. Addison-Wesley, 2000.
- [16] HAVERTY, N. Mil-std 1553 - a standard for data communications. *Communication and Broadcasting*, 1986. volume 10, nº 1, págs. 29–33.
- [17] HOWELL, Rodney R.; VENKATRAO, Muralidhar K. On non-preemptive scheduling of recurring tasks using inserted idle times. *Information and Computation*, 1995. , nº 117, págs. 50–62.
- [18] IEC-61158. *Fieldbus Standard for Use in Industrial Control Systems*. International Electrotechnical Commission, 1993.
- [19] INTEL. *The Bitbus Interconnect Serial Control Bus Specifications*. Intel Corporation, 1984.
- [20] ISO-11898. *Road Vehicle - Interchange or Digital Information - Controller Area Network (CAN) for High Speed Communication*. International Organization for Standardization, 1993.
- [21] ISO-8824. *Information Technology - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1)*. International Organization for Standardization, 1990.
- [22] ISO/IEC-7498-1. *Information Technology - Open Systems Interconnection - Basic Reference Model*. International Organization for Standardization, 1994.
- [23] ITAIPU (Ed.). *Itaipu Hydroelectric Power Station - Technical Specifications - 7028-20-15200-P - ESP 022/97 - Turbines - Generators and Accessories - Units 9A and 18A - Chapter VI - Digital Supervision and Control System*. Brasil: Itaipu Binacional, 1998.

- [24] JEFFAY, Kevin; STANAT, Donald F.; MARTEL, Charles U. On non-preemptive scheduling of periodic and sporadic tasks. *Proceedings of the Twelfth IEEE Real-Time Systems Symposium*, 1991. págs. 129–139.
- [25] KOPETZ, Hermann. *Real-Time Systems: Design Principles for Distributed Embedded Systems*. Kluwer Academic Publishers, 1997.
- [26] LEHOCZKY, J.; SHA, L.; DING, Y. The rate monotonic scheduling algorithm: Exact characterization and average case behavior. *Proceedings of 10th IEEE Real-Time Systems Symp.*, 1989. págs. 166–171.
- [27] LIU, C. L.; LAYLAND, James W. Scheduling algorithms for multiprogramming in a hard real-time environment. *Journal of the Association for Computing Machinery*, 1973. volume 20, nº 1, págs. 46–61.
- [28] LIU, Jane W. S. *Real-Time Systems*. Prentice Hall, 2000.
- [29] MALCOLM, Nicholas; ZHAO, Wei. Hard real-time communication in multiple-access networks. *Real-Time Systems*, 1995. volume 8, nº 1, págs. 35–77.
- [30] MCGUFFIN, L.J.; REID, L.O.; SPARKS, S.R. Map/top in cim distributed computing. *IEEE Network*, 1988. volume 2, nº 3, págs. 23–31.
- [31] MOZINA, Charles J. (Ed.). *IEEE Tutorial on the Protection of Synchronous Generators*. IEEE Power Engineering Society, 1995.
- [32] PIMENTEL, Juan R. *Communication Networks for Manufacturing*. Prentice Hall, 1990.
- [33] PLEINEVAUX, Patrick; DECOTIGNIE, Jean-Dominique. Time critical communication networks: Field buses. *IEEE Network*, 1988. volume 2, nº 3, págs. 55–63.
- [34] RAMAMRITHAM, K.; STANKOVIC, J. Scheduling algorithms and operating systems support for real-time systems. *Proceedings of the IEEE*, 1994. volume 82, nº 1, págs. 55–57.
- [35] RAMSAY, D.C. *Principles of Engineering Instrumentation*. 1<sup>a</sup> ed. Arnold, 1996.
- [36] ROSEMOUNT. *HART - Smart Communications Protocol Specification. Rev. 5.1.4*. Rosemount Inc., 1991.

- [37] SHA, Lui; GOODENOUGH, J.B. Real-time scheduling theory and ada. *IEEE Computer*, 1990. volume 23, nº 4, págs. 53–62.
- [38] SHA, Lui; SATHAYE, Shirish S. A systematic approach to designing distributed real-time systems. In: *Advances in Real-Time Systems*, ed. SON, Sang H. Prentice Hall, 1993 págs. 149–171.
- [39] STANKOVIC, J.A. *Real Time Computing Systems: The Next Generation*. Rel. téc., IEEE Tutorial on Hard Real-Time Systems, 1988.
- [40] TINDELL, Ken. *Analysis of Hard Real-Time Communications*. Rel. téc., Real-Time Systems Systems Research Group, University of York, 1995.
- [41] TOVAR, Eduardo M. M. *Supporting Real-Time Communications with Factory-Floor Networks*. Tese de Doutoramento, Universidade do Porto, Portugal, 1999.
- [42] VASQUES, Francisco; JUANOLE, Guy. *Pre-Run-Time Schedulability Analysis in Fieldbus Networks*. Rel. téc., LAAS du CNRS, Toulouse Cedex, França, 1994.
- [43] WORLDFIP. *WorldFIP Protocol*. WorldFIP Organization, Clamart, França, 1988.
- [44] XU, Jia; PARNAS, David L. On satisfying timing constraints in hard-real-time systems. *IEEE Transactions on Software Engineering*, 1993. volume 19, nº 1, págs. 70–84.