

**UNIVERSIDADE FEDERAL DE SANTA CATARINA  
DEPARTAMENTO DE AUTOMAÇÃO E SISTEMAS**

Tiago Semprebom

**EXPLORANDO DESCARTES DE ATIVAÇÕES  
PERIÓDICAS PARA PROVIMENTO DE QUALIDADE  
DE SERVIÇO EM REDES IEEE 802.15.4**

Florianópolis

2012



Tiago Semprebom

**EXPLORANDO DESCARTES DE ATIVAÇÕES  
PERIÓDICAS PARA PROVIMENTO DE QUALIDADE  
DE SERVIÇO EM REDES IEEE 802.15.4**

Tese submetida ao Programa de Pós-Graduação em Engenharia de Automação e Sistemas (PPGEAS) para a obtenção do Grau de Doutor em Engenharia de Automação e Sistemas.

Orientador: Carlos Barros Montez, Dr.

Florianópolis

2012

Ficha de identificação da obra elaborada pelo autor,  
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Semprebom, Tiago

Explorando Descartes de Ativações Periódicas para  
Provimento de Qualidade de Serviço em Redes IEEE 802.15.4  
[tese] / Tiago Semprebom ; orientador, Carlos Barros  
Montez - Florianópolis, SC, 2012.  
162 p. ; 21cm

Tese (doutorado) - Universidade Federal de Santa  
Catarina, Centro Tecnológico. Programa de Pós-Graduação em  
Engenharia de Automação e Sistemas.

Inclui referências

1. Engenharia de Automação e Sistemas. 2. Redes de  
Sensores sem Fios. 3. IEEE 802.15.4. 4. (m,k)-firm. 5.  
escalonamento tempo real. I. Montez, Carlos Barros. II.  
Universidade Federal de Santa Catarina. Programa de Pós-  
Graduação em Engenharia de Automação e Sistemas. III. Título.

Tiago Semprebom

**EXPLORANDO DESCARTES DE ATIVAÇÕES  
PERIÓDICAS PARA PROVIMENTO DE QUALIDADE  
DE SERVIÇO EM REDES IEEE 802.15.4**

Esta Tese foi julgada aprovada para a obtenção do Título de “Doutor em Engenharia de Automação e Sistemas”, e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia de Automação e Sistemas (PPGEAS).

Florianópolis, 16 de julho 2012.

---

Jomi Fred Hübner, Dr.  
Coordenador do PPGEAS

---

Carlos Barros Montez, Dr.  
Orientador

**Banca Examinadora:**

---

Prof. Carlos Barros Montez, Dr.  
Presidente

---

Prof. Nunzio Marco Torrisi, Dr.



---

Prof. Francisco Vasques, Dr.

---

Prof. Ricardo Alexandre Reinaldo de Moraes, Dr.

---

Prof. Rômulo Silva de Oliveira, Dr.

---

Prof. Alex Sandro Roschildt Pinto, Dr.





Aos meus amados pais, Dersi e Elizabeth  
e a minha noiva Fran...

## AGRADECIMENTOS

Já ouviu-se dizer que “a gratidão é a lembrança do coração”. Ao longo de minha jornada sempre pude contar com inúmeras pessoas especiais, verdadeiros ”anjos da guarda”, sem os quais meus objetivos seriam muito difíceis de alcançar ou seriam até inatingíveis. Por isso essa parte da minha tese é tão especial . Quero expressar de coração meu agradecimento às seguintes pessoas:

À Deus pelos momentos de felicidade que me dão força e inspiração e também pelos momentos de dificuldade que me ajudam a ser um ser humano melhor.

À minha família, o alicerce da minha vida: meus pais, Dersi e Elizabeth pelo amor incondicional; pelo apoio nos momentos de dificuldade e de inquietantes decisões. Agradeço por estarem sempre ao meu lado em todas as minhas conquistas e realizações, pois estes não teriam valor se vocês não estivessem comigo.

Ao meu amor, a Fran pelo companheirismo em todos os momentos, pelos sorrisos e por toda a paciência e dedicação que teve durante a realização desta empreitada. Te amo!

Ao Prof. Dr. Carlos Montez, por ser mais do que meu orientador, por acreditar em minha capacidade e no meu crescimento pessoal e profissional, pelo apoio em todos os momentos, principalmente pela amizade. A você Montez, meu muito obrigado, sem você este projeto não teria existido.

À todos os professores, funcionários e colegas do Departamento de Automação e Sistemas da UFSC, em especial ao Gustavo Zomer pela constante colaboração no desenvolvimento deste trabalho.

Ao Prof. Dr. Francisco Vasques (FEUP-UP), pela colaboração durante a realização desta obra e por ter me orientado durante o período de doutorado sanduíche na Universidade do Porto, em Portugal.

À todos os meus colegas professores do núcleo de telecomunicações do Instituto Federal de Santa Catarina, Campus São José, em especial agradeço aos seguintes amigos: Evandro Cantú, Emerson Ribeiro Mello, Márcio Doniak, Odilson Tadeu Valle, Maria Cláudia Castro, Marcelo Maia Sobral, Eraldo Silveira e Silva, Mário de Noronha Neto e Marcos Moecke. Agradeço também duas pessoas que sempre acreditaram nesse projeto e colaboraram diretamente para sua realização: Prof. Jesué Graciliano da Silva e a Profa. Maria Clara Kaschny Schneider.

À CAPES pela concessão de bolsas durante os dois primeiros anos do meu doutorado. Agradeço também ao consórcio *Erasmus Mundus* - *EWB* por fomentar minha estada na Universidade do Porto.

*“Aqueles que passam por nós, não vão  
sós, não nos deixam sós. Deixam um  
pouco de si, levam um pouco de nós.”*

Antoine de Saint-Exupéry



## RESUMO

Atualmente, tem se considerado cada vez mais o uso de redes sem fios em sistemas de automação industrial e controle de processos. Sistemas de Controle via Rede são exemplos de aplicações em que busca-se implantar em ambientes industriais. Para estes tipos de aplicações, o suporte à serviços de comunicação tempo real é uma das maiores necessidades. Tradicionalmente, aplicações de tempo real industriais assumem que nenhuma amostra será perdida durante os ciclos de controle. Contudo, alguns estudos realizados nos últimos anos indicam que os efeitos dos descartes de mensagens frente ao desempenho dos sistemas de controle podem ser compensados com emprego de outras técnicas. O padrão IEEE 802.15.4. têm se mostrado uma solução atrativa para uma vasta gama de aplicações no campo das Redes de Sensores sem Fios e mais recentemente no campo da instrumentação sem fios. Sendo assim, esta tese tem por objetivo principal propor abordagens de provimento de Qualidade de Serviço (QoS) em redes IEEE 802.15.4 para aplicações de tempo real que tolerem perdas de *deadline*. Primeiramente, um mecanismo de priorização de tráfego para dispositivos de tempo real, durante períodos de acesso com contenção (CAP) foi proposto. Este mecanismo, denominado DDBP, oferece de forma descentralizada aos dispositivos que desejam realizar suas transmissões durante o CAP, priorização de acesso ao meio de comunicação de acordo com a distância para falha dos dispositivos. Posteriormente, duas abordagens de escalonamento durante períodos de acesso sem contenção (CFP) foram propostas: abordagem SDBP, a qual oferece garantias de atendimento de *deadline* às aplicações, através de um teste de escalonabilidade determinístico e uma heurística *on-line* de priorização de mensagens; e abordagem  $(m,k)$ -*spin*, a qual apresenta uma análise de escalonabilidade baseada no conceito de período de nível de ocupação. As abordagens propostas voltadas para o escalonamento durante o CFP oferecem garantias temporais para as aplicações com restrições de QoS, modeladas de acordo com o modelo de tarefas  $(m,k)$ -*firm*. Dispositivos que não conseguem realizar suas transmissões durante períodos livres de contenção, poderão realizar suas transmissões utilizando a abordagem de priorização de tráfego tempo real desenvolvida para o CAP.

**Palavras-chave:** Redes de Sensores sem Fios, IEEE 802.14.5,  $(m,k)$ -*firm*, escalonamento tempo real.



## ABSTRACT

In the last few years, the use of wireless networks has been increasingly seen in industrial automation and process control. Network Control Systems are examples of driving applications in industrial environments. For these types of applications, support for real-time communications services is one of the major requirements. In traditional approaches, many of these real-time applications assume no data losses during the control cycle. However some more recent studies indicate that the effect of control messages discards upon the performance of the control systems may be significantly encompassed with the adoption of other techniques.

The IEEE 802.15.4 protocol is an attractive solution for a wide range of applications in the field of Wireless Sensors Networks and more recently in the field of wireless instrumentation. Thus, this thesis aims to propose approaches to providing Quality of Service (QoS) in IEEE 802.15.4 for real-time applications that tolerate deadline misses. First, a mechanism for prioritizing traffic for real time devices during Contention Access Periods (CAP) was proposed. This mechanism, called DDBP offers in a decentralized way, priority access in accordance to the distance to device failure. Subsequently, two approaches to scheduling during Contention Free Periods (CFP) were proposed: SDBP approach which provides deadlines guarantees by the adoption of a deterministic scheduling test and an on-line prioritization heuristic; and (m,k)-spin approach, which presents a schedulability analysis based on the busy period concept.

The proposed approaches applied to the CFP scheduling provide temporal guarantees for applications with QoS constraints, modelled according to the (m,k)-firm task model. Devices that cannot perform their transmissions during contention free periods, can try to perform their transmissions using the real-time traffic prioritization approach developed for the CAP.

**Keywords:** Wireless Sensor Networks, IEEE 802.15.4, (m,k)-firm, real-time scheduling.





## LISTA DE FIGURAS

Figura 1	Laço de controle de uma aplicação industrial. . . . .	26
Figura 2	Exemplo de sistema de tempo real e seu ambiente. . . . .	34
Figura 3	Parâmetros típicos de uma tarefa de tempo real. . . . .	36
Figura 4	Funções benefício associadas a uma tarefa tempo real. .	38
Figura 5	Parâmetros típicos para escalonamento DM. . . . .	44
Figura 6	Cálculo do período de ocupação de nível $i$ ( $D_i > T_i$ ). . .	45
Figura 7	Escalação melhor esforço. . . . .	48
Figura 8	Escalação garantido. . . . .	48
Figura 9	Escalação robusto. . . . .	49
Figura 10	Abordagens de escalação tempo real (adaptado de (OLIVEIRA, 1997)). . . . .	50
Figura 11	Topologia estrela. . . . .	55
Figura 12	Topologia ponto a ponto. . . . .	56
Figura 13	Agrupamento em árvore. . . . .	57
Figura 14	Modos de operação IEEE 802.15.4. . . . .	60
Figura 15	Estrutura do superquadro sem GTSs. . . . .	61
Figura 16	Exemplo de estrutura do superquadro com CAP e CFP. . . . .	64
Figura 17	Espaçamento entre quadros no IEEE 802.15.4. . . . .	64
Figura 18	Algoritmos CSMA/CA com e sem compartimentos. . . . .	66
Figura 19	Estrutura do quadro de solicitação de GTS. . . . .	69
Figura 20	Fragmentação e realocação de GTSs. . . . .	70
Figura 21	Utilização real na alocação de compartimentos de GTS. . . . .	71
Figura 22	Pilha de protocolos ZigBee/IEEE 802.15.4. . . . .	73
Figura 23	Elementos de uma rede WirelessHart. . . . .	75
Figura 24	Distância para falha para uma tarefa (2,3)- <i>firm</i> (baseado em (HAMDAOUI; RAMANATHAN, 1995a)). . . . .	81
Figura 25	Janela de atendimento de <i>deadlines</i> $k_i.P_i$ . . . . .	86
Figura 26	Modelo em topologia estrela. . . . .	103
Figura 27	Algoritmo Distributed DBP (executado em cada nodo). . . . .	109
Figura 28	Falhas dinâmicas <i>vs.</i> utilização. . . . .	113
Figura 29	Escalação de GTS segundo Slotted DBP. . . . .	121
Figura 30	Falhas dinâmicas <i>vs.</i> fator de utilização. . . . .	123

Figura 31 Padrão de classificação inicial para uma tarefa (2,3)- <i>firm</i> .	125
Figura 32 Classificador de tarefas (m,k)- <i>firm</i> .	126
Figura 33 Padrões gerados a partir de rotações em uma tarefa (2,3)- <i>firm</i> .	126
Figura 34 Cálculo do período de ocupação de nível $i$ .	129
Figura 35 Tempos de chegada da tarefa $\tau_j$ , com restrições (2,5)- <i>firm</i> .	133
Figura 36 Escala de execução (m,k)- <i>spin</i> .	135
Figura 37 Porcentagem de tarefas admitidas pelo (m,k)- <i>spin</i> , que foram rejeitas por <i>Ramanathan</i> (com carga=100%).	136
Figura 38 Porcentagem de tarefas admitidas pelo (m,k)- <i>rec-spin</i> , que foram rejeitas por <i>Ramanathan</i> (com carga=100%).	137
Figura 39 Porcentagem de tarefas admitidas, considerando apenas conjuntos harmônicos.	138
Figura 40 Cenário experimental.	140
Figura 41 Análise comparativa entre os trabalhos.	142

## LISTA DE TABELAS

Tabela 1	Frequências, modulações e taxas de dados IEEE 802.15.4.	58
Tabela 2	Valores de IFS.....	65
Tabela 3	Períodos do superquadro no IEEE 802.15.4 (sem período inativo). ....	71
Tabela 4	Carga útil no IEEE 802.15.4 (bytes). ....	72
Tabela 5	Precedência de execução das tarefas segundo DWCS. ...	84
Tabela 6	Abordagem tradicional e abordagem DDBP.....	112
Tabela 7	Falhas Dinâmicas e perdas de <i>deadlines</i> em diferentes fatores de utilização. ....	123
Tabela 8	Falhas dinâmicas considerando um número fixo de nodos.	124
Tabela 9	Aceitação para conjuntos de tarefas não harmônicos....	138
Tabela 10	Aceitação para conjuntos de tarefas harmônicos.....	139
Tabela 11	Tempos de resposta (ms). ....	141



## LISTA DE ABREVIATURAS E SIGLAS

ACK	Acknowledgment
BE	Backoff Exponent
BI	Beacon Interval
BPSK	Binary Phase Shift Keying
BO	Beacon Order
BWP	Blue When Possible
CA	Collision Avoidance
CAP	Contention Access Period
CCA	Clear Channel Assessment
CH	Cluster Head
CFP	Contention Free Period
CID	Cluster Identifier
COTS	Commercial-Of-The-Shelf
CSMA	Carrier Sense Multiple Access
CTS	Clear to Send
DDBP	Distributed DBP
DBP	Distance Based Priority
DM	Deadline Monotonic
DWCS	Dynamic Window-Constrained Scheduling
ED	Energy Detection
EDF	Earliest Deadline First
FDD	Full Function Device
FDM	Frequency Division Multiplexing
FIFO	First-in, First-out
GDPA	Guaranteed Dynamic Priority Assignment
GTS	Guaranteed Time Slot
HART	Highway Addressable Remote Transducer
HC	Hybrid Coordinator
HCF	Hybrid Coordinator Function
HCCA	HCF Controlled Channel Access
EDCA	Enhanced Distributed Channel Access
IEEE	Institute of Electrical and Electronics Engineers

IFS	Interframe Space
ISA	Instrumentation Systems and Automation Society
ISM	Industrial, Scientific and Medical
LIFS	Long Interframe Space
LQI	Link Quality Indication
LR	Low-Rate
MAC	Medium Access Control
MMC	Mínimo Múltiplo Comum
NB	Number of Backoff
NCS	Networked Control Systems
O-QPSK	Offset Quadrature Phase-Shift Keying
PAN	Personal Area Network
PC	Point Coordinator
PCF	Point Coordinator Function
PHY	Physical Layer
QoS	Quality of Service
RFD	Reduced Function Device
RM	Rate Monotonic
RM-RTO	Rate Monotonic-RTO
RSSF	Rede de Sensores sem Fios
RTS	Request to Send
RTO	Red Task Only
SD	Superframe Duration
SDBP	Slotted DBP
SIFS	Short Interframe Space
SO	Superframe Order
STR	Sistemas de Tempo Real
TDMA	Time Division Multiple Access
VDS	Virtual Deadline Scheduling
WCET	Worst Case Execution Time
WPAN	Wireless Personal Area Network

# SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	25
1.1 MOTIVAÇÃO .....	26
1.2 OBJETIVOS DA TESE .....	28
1.3 CONTRIBUIÇÕES DA TESE .....	30
1.4 ORGANIZAÇÃO DO TEXTO .....	30
<b>2 FUNDAMENTAÇÃO TEÓRICA</b> .....	33
2.1 SISTEMAS DE TEMPO REAL .....	33
2.1.1 Modelo de tarefas .....	35
2.1.2 <i>Deadlines</i> e funções benefício .....	37
2.1.3 Escalonamento tempo real .....	39
2.1.4 Escalonamento de tarefas periódicas .....	41
2.1.5 Escalonamento de tarefas aperiódicas .....	47
2.1.6 Escalonamento na sobrecarga .....	47
2.1.7 Abordagens adaptativas .....	49
2.2 MODELO DE TAREFAS (M,K)- <i>FIRM</i> .....	52
2.3 ESPECIFICAÇÃO IEEE 802.15.4 .....	53
2.3.1 Descrição geral do IEEE 802.15.4 .....	53
2.3.2 Topologias de rede .....	54
2.3.3 Camada física IEEE 802.15.4 .....	58
2.3.4 Controle de Acesso ao Meio do IEEE 802.15.4 .....	59
2.3.4.1 Modos de operação do IEEE 802.15.4 .....	60
2.3.4.2 Estrutura do superquadro .....	62
2.3.4.3 Protocolo de acesso ao meio CSMA/CA .....	65
2.3.4.4 Alocação e gerência de GTS .....	68
2.3.5 Estudo dos limites físicos do padrão IEEE 802.15.4 .....	70
2.3.6 ZigBee .....	72
2.3.7 WirelessHart .....	73
2.3.7.1 Visão geral .....	74
2.3.7.2 Arquitetura de rede .....	75
2.4 CONCLUSÕES DO CAPÍTULO .....	76
<b>3 TRABALHOS RELACIONADOS</b> .....	79
3.1 ESCALONAMENTO DE TAREFAS QUE TOLERAM DES- CARTES DE ATIVAÇÕES .....	79
3.1.1 Escalonamento DBP .....	80
3.1.2 Escalonamento Skip-Over .....	81
3.1.3 Modelo de tarefas com <i>deadlines</i> (n,m)- <i>hard</i> .....	83
3.1.4 Escalonamento baseado em janelas .....	83

3.1.5 Escalonamento baseado em <i>deadlines</i> virtuais . . . . .	85
3.2 ESCALONAMENTO TDMA . . . . .	86
3.2.1 Protocolo de comunicação com compressão de compartimentos . . . . .	87
3.2.2 Escalonamento de mensagens em redes de sensores/atuadores . . . . .	88
3.2.3 Protocolo de acesso ao meio com priorização implícita . . . . .	89
3.3 ESCALONAMENTO DE GTS EM REDES IEEE 802.15.4 . . . . .	90
3.4 TESTES DE ESCALONABILIDADE . . . . .	92
3.4.1 Teste de escalonabilidade para o DBP . . . . .	92
3.4.2 Teste de escalonabilidade exato para tarefas (m,k)- <i>firm</i> . . . . .	93
3.5 CONSIDERAÇÕES SOBRE AS ABORDAGENS . . . . .	94
3.6 CONCLUSÕES DO CAPÍTULO . . . . .	96
4 MODELO DO SISTEMA E MÉTRICAS DE DESEMPENHO . . . . .	99
4.1 INTRODUÇÃO . . . . .	99
4.2 MODELO DO SISTEMA . . . . .	102
4.3 MÉTRICAS DE DESEMPENHO . . . . .	104
4.4 CONCLUSÕES DO CAPÍTULO . . . . .	105
5 ESCALONAMENTO NO PERÍODO DE ACESSO COM CONTENÇÃO . . . . .	107
5.1 MOTIVAÇÃO . . . . .	107
5.2 ALGORITMO DISTRIBUTED DBP . . . . .	108
5.3 RESULTADOS . . . . .	110
5.4 CONCLUSÕES DO CAPÍTULO . . . . .	114
6 ESCALONAMENTO EM PERÍODOS DE ACESSO SEM CONTENÇÃO . . . . .	115
6.1 ABORDAGENS PROPOSTAS . . . . .	115
6.1.1 SIGHT: explorando descartes de tarefas periódicas no escalonamento de GTSs . . . . .	116
6.1.1.1 Algoritmo Slotted DBP . . . . .	117
6.1.1.2 Corretude do Slotted DBP . . . . .	119
6.1.1.3 Exemplo de escalonamento Slotted DBP . . . . .	120
6.1.1.4 Resultados de experimentos por simulação . . . . .	121
6.1.2 (m,k)- <i>spin</i> : alocação de GTSs com giro no padrão de classificação (m,k)- <i>firm</i> . . . . .	124
6.1.2.1 Classificador de tarefas . . . . .	125
6.1.2.2 Análise de escalonabilidade . . . . .	128
6.1.2.3 Prova formal do teste de escalonabilidade . . . . .	130
6.1.2.4 Exemplo de aplicação . . . . .	134



6.1.2.5 Resultados .....	135
6.2 COMPARAÇÃO COM TRABALHOS RELACIONADOS ..	141
6.3 CONCLUSÕES DO CAPÍTULO .....	142
<b>7 CONCLUSÕES</b> .....	145
7.1 REVISÃO DAS MOTIVAÇÕES E OBJETIVOS .....	145
7.2 VISÃO GERAL DO TRABALHO .....	146
7.3 CONTRIBUIÇÕES DA TESE .....	148
7.4 TRABALHOS FUTUROS .....	150
<b>REFERÊNCIAS</b> .....	153



# 1 INTRODUÇÃO

Avanços recentes das tecnologias de comunicação e computação têm propiciado o surgimento de dispositivos multifuncionais com baixo custo e baixo consumo de energia, os quais podem observar e reagir a mudanças no ambiente físico que estão inseridos.

Uma Rede de Sensores sem Fios (RSSFs) típica consiste em um conjunto de dispositivos nodos sensores em um ambiente a ser monitorado e controlado. Uma característica destas redes é a possibilidade de coexistir um conjunto heterogêneo de nodos, onde alguns nodos são tipicamente mais poderosos do que outros, possuem diferentes conjuntos de recursos disponíveis e suportam múltiplos tipos de tráfego, incluindo dados periódicos, intermitentes e de tempo real.

O padrão IEEE 802.15.4 têm se mostrado uma solução atrativa para uma vasta gama de aplicações no campo das RSSFs e mais recentemente no campo da instrumentação sem fios (e.g. controle de processos).

Atualmente, um número significativo de trabalhos de pesquisa estão sendo efetuados no desenvolvimento de redes sem fios que ofereçam Qualidade de Serviço (QoS) (ZHANG et al., 2011; RUIZ et al., 2011) às aplicações. Esta tendência é uma consequência da crescente utilização de comunicações sem fio em ambientes de escritório e doméstico. Portanto, é provável que num futuro próximo, a ampla disponibilidade de soluções de redes sem fios irá também gerar um padrão *de facto* para comunicação sem fios em ambientes industriais, onde o conjunto de protocolos normalizados IEEE 802.11 (802.11, 2007) e IEEE 802.15.4 (802.15.4, 2006) são os principais candidatos.

No entanto, os requisitos de comunicação para ambientes industriais são muito específicos. Para além do tráfego genérico (por exemplo, o tráfego de dados e multimídia) similar àquele encontrado em ambientes de escritório/doméstico, existe tráfego com requisitos de tempo real. Desta forma se faz necessária a implementação de novas soluções de comunicação para essas aplicações, obedecendo as especificações já implementados em dispositivos de prateleira – *Commercial-Of-The-Shelf* (COTS).

## 1.1 MOTIVAÇÃO

Atualmente, tem se considerado cada vez mais o uso de redes sem fios em sistemas de automação industrial e controle de processos (BELLO; TOSCANO, 2009; TOSCANO; BELLO, 2011, 2012; WILLIG, 2008). A motivação mais evidente é a necessidade de redução da quantidade de cabos nas plantas industriais, proporcionando a redução de custos de construção e manutenção dos sistemas; tornando mais fácil a configuração dinâmica das máquinas, oferecendo mobilidade e flexibilidade aos sistemas de controle industriais. Voz sobre IP (VoIP) e Sistemas de Controle via Rede (NCS) são exemplos de novas aplicações que buscase implantar em ambientes industriais. Para estes tipos de aplicações, o suporte à serviços de comunicação tempo real é uma das maiores necessidades.

Muitas atividades em ambientes industriais, tais como amostragem e atuação, devem ser periodicamente realizadas e executadas dentro de intervalos de tempo limitados. A Figura 1 ilustra uma aplicação de controle típica, onde dispositivos sensores amostram informações que servem como entrada para um controlador, o qual gera um sinal de controle para um ou mais atuadores. Em abordagens tradicionais, muitos desses controladores, assumem que nenhuma amostra será perdida durante o ciclo de controle. Contudo, alguns estudos realizados nos últimos anos (WILLIG, 2008; HESPANHA et al., 2007; LEMMON et al., 2003; SCHENATO et al., 2007) indicam que os efeitos dos descartes de mensagens de controle frente ao desempenho dos sistemas de controle podem ser significativamente compensados. Entretanto, estes trabalhos propõem o uso de técnicas onde o descarte de algumas ativações (ou perdas de *deadlines*) são suportados, contanto que estes descartes sejam precisamente espaçados. Sendo possível a utilização de técnicas de interpolação ou estimação, como filtros de Kalman (NAKAMURA et al., 2007) para compensar o não recebimento de uma mensagem periódica.

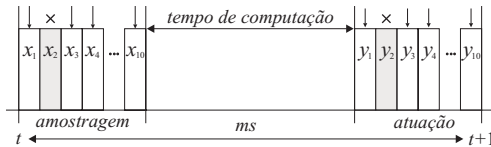


Figura 1: Laço de controle de uma aplicação industrial.

O grupo de trabalho IEEE 802.15.4 (802.15.4, 2006) em conjunto com a Aliança ZigBee (ZIGBEE-ALLIANCE, 2007) têm desenvolvido a

especificação para um padrão de Rede Pessoal de Área Privada de Curto Alcance (LR-WPAN) denominada IEEE 802.15.4. O interesse da Aliança ZigBee neste tipo de tecnologia advém do fato da ampla diversidade de aplicações que podem tirar proveito das características oferecidas pelas LR-WPANs. Estas redes apresentam a vantagem de serem flexíveis o bastante para se adequarem às necessidades de diversos conjuntos de aplicações em diferentes campos.

A tecnologia IEEE 802.15.4 vem sendo adotada em RSSFs. Características como baixo consumo energético, baixa taxa de transmissão e baixo custo tipicamente se encaixam com às necessidades das RSSFs. Adicionalmente, o protocolo IEEE 802.15.4 prevê garantias para aplicações de tempo real através do mecanismo de Compartimento de Tempo Garantido (GTS). Quando a rede opera no modo com *beacon*, uma largura de banda pode ser dedicada exclusivamente para transmissão de um dispositivo que necessite de garantias temporais. Este período de tempo no qual há compartimentos garantidos é denominado *Contention Free Period* (CFP). Desta forma, o mecanismo de GTS oferece garantias mínimas para os dispositivos que alocam GTSs e permite, assim que estes dispositivos tenham uma estimativa de pior caso dentro da rede, uma vez que estes não precisarão disputar compartimentos com outros dispositivos que também desejam se comunicar.

No entanto, a principal desvantagem do mecanismo de GTS é o número de compartimentos garantidos que podem ser alocados simultaneamente entre os nodos da rede. De acordo com a especificação IEEE 802.15.4 (802.15.4, 2006), durante o CFP no máximo sete compartimentos podem ser alocados. Este recurso escasso, pode rapidamente exaurir-se quando considerada uma rede com uma grande quantidade de nodos. Como consequência, os dispositivos que não possuem compartimentos alocados devem tentar realizar suas transmissões durante o período com contenção (CAP), usando o mecanismo de acesso ao meio *Carrier Sense Multiple Access with Collision Avoidance* (CSMA/CA). Contudo, o CSMA/CA não prioriza tráfego e não fornece qualquer tipo de garantia temporal ou de entrega de mensagens para as aplicações.

Essas limitações não seriam muito estritas caso as aplicações que executam nessas redes tivessem um número reduzido de nodos e um número limitado de fluxos de mensagens periódicas, pois estas poderiam ser alocadas constantemente nos compartimentos garantidos. Porém, uma simples aplicação de controle, composta por alguns nodos sensores e atuadores distribuídos, já esgotaria esses recursos. Executar mais de uma malha de controle sobre essas redes, torna-se desafiador, e um complexo trabalho de engenharia, pois envolve a confi-

guração dos períodos e *offsets* das tarefas periódicas, buscando um rodízio na alocação das mensagens existentes nos compartimentos garantidos. Além disso, mesmo que fosse aplicada alguma técnica de otimização para a configuração desses períodos e *offsets*, a solução resultante tornar-se-ia demasiadamente estática e inflexível.

Uma das principais motivações por trás deste trabalho é o fato que, em algumas situações, aplicações podem tolerar perdas de *deadlines* e descartes de mensagens sem que ocorra uma falha no sistema (e.g. NCSs). A limitação do número máximo de GTS supracitadas já seria um forte motivo para se propor abordagens e modelos de tarefas mais flexíveis para as aplicações que executam sobre redes IEEE 802.15.4. No entanto, além disso, apesar de alguns trabalhos recentes (ex. (ANASTASI et al., 2011; BELLO; TOSCANO, 2009; ANDERSSON et al., 2008; TOSCANO; BELLO, 2012; CACCAMO; ZHANG, 2002; KOUBAA et al., 2006b)), o uso adequado de abordagens de tempo real nessas redes não têm sido totalmente investigado. Principalmente no que tange à exploração da capacidade das aplicações para lidar com as perdas de *deadlines* e de ativações periódicas. Oferecer QoS provendo algum tipo de garantia temporal exigida por algumas aplicações que utilizam redes IEEE 802.15.4 ainda é um desafio em aberto.

## 1.2 OBJETIVOS DA TESE

Esta tese investiga as seguintes questões interligadas ao escopo de comunicação de tempo real em redes IEEE 802.15.4:

1. É possível desenvolver novas abordagens que estendam o Controle de Acesso ao Meio (MAC) de redes IEEE 802.15.4 de forma a explorar a capacidade de tarefas periódicas lidarem, em tempo de execução, com eventuais perdas de *deadlines* sem nenhuma alteração ao padrão IEEE 802.15.4?
2. É possível que estas abordagens atuem tanto nos mecanismos de envio e recepção de mensagens que usam tanto os compartimentos garantidos (CFP – *Contention Free Period*), quanto nas que usam o protocolo CSMA/CA (CAP – *Contention Access Period*)?

Sendo assim, esta tese tem por objetivo principal propor abordagens de provimento de QoS em redes IEEE 802.15.4 para aplicações de tempo real que tolerem perdas de *deadline*. Primeiramente, um mecanismo de priorização de tráfego para dispositivos de tempo real, durante períodos de acesso com contenção (CAP) foi proposto. Este

mecanismo, denominado Distributed DBP (DDBP), oferece de forma descentralizada aos dispositivos que desejam realizar suas transmissões durante o CAP, priorização de acesso ao meio de comunicação de acordo com a distância para falha dos dispositivos. Posteriormente, duas abordagens de escalonamento durante períodos de acesso sem contenção (CFP) foram propostas: (i) abordagem Slotted DBP (SDBP), a qual oferece garantias de atendimento de *deadline* às aplicações, através de um teste de escalonabilidade determinístico e uma heurística *on-line* de priorização de mensagens; (ii) abordagem (m,k)-*spin*, apresenta uma análise de escalonabilidade baseada no conceito de período de nível de ocupação, onde um teste de admissão *on-line* é executado em tempo polinomial. Esta última abordagem assume padrões de classificação de tarefas pré-definidos, possibilitando assim que um maior número de tarefas seja admitido no sistema.

As abordagens propostas voltadas para o escalonamento de CFP oferecem garantias temporais para as aplicações com restrições de QoS, modeladas de acordo com o modelo de tarefas (m,k)-*firm*. Dispositivos que não conseguem realizar suas transmissões durante períodos livres de contenção, poderão realizar suas transmissões utilizando a abordagem de priorização de tráfego tempo real desenvolvida para o CAP, entretanto importante ressaltar que o uso do CAP não oferece garantias temporais para as aplicações. Para atingir o objetivo principal desta tese, os seguintes objetivos específicos foram delineados:

- Especificação formal do modelo de sistema para fins de especificação de comportamento e para facilitar a verificação de algumas propriedades de tempo real, importantes para a proposta;
- Criação de um algoritmo de escalonamento de tarefas com restrições de QoS especificadas em conformidade com modelo (m,k)-*firm*, durante períodos de acesso com contenção, em redes IEEE 802.15.4;
- Proposição de um mecanismo de escalonamento de compartimentos de tempo garantido em redes IEEE 802.15.4;
- Definição de testes de escalonabilidade exato para tarefas (m,k)-*firm* em redes IEEE 802.15.4;
- Avaliação das abordagens através de método analítico, simulações e de cenário experimental.

### 1.3 CONTRIBUIÇÕES DA TESE

A principal contribuição deste trabalho de doutorado é a proposição de novas abordagens para provimento de QoS à aplicações de tempo real baseadas no modelo de tarefas  $(m,k)$ -firm. As propostas apresentam total compatibilidade com o padrão IEEE 802.15.4. Desta forma, não existe necessidade de alterações no padrão, sendo as soluções propostas compatíveis com dispositivos de prateleira (COTS) que já implementam o protocolo IEEE 802.15.4. Dentre as contribuições alcançadas com o desenvolvimento deste trabalho, destacam-se:

- Proposição de um modelo de sistema para tarefas com restrições temporais especificadas na forma de *deadlines*  $(m,k)$ -firm em redes IEEE 802.15.4;
- Estudo das limitações físicas impostas pelo padrão IEEE 802.15.4 no que tange o provimento de suporte de tempo real às aplicações. O padrão oferece pouca flexibilidade no oferecimento de valores de períodos e duração dos compartimentos de tempo para aplicações;
- Extensão da abordagem de escalonamento DBP (*Distance Based Priority*) aplicado de forma descentralizada em redes IEEE 802.15.4. Abordagem oferece priorização para tarefas em proximidade de uma falha dinâmica;
- Proposição de um teste de escalonabilidade exato para tarefas com restrições  $(m,k)$ -firm. A solução *on-line* proposta melhora o mecanismo de alocação de GTSs atualmente empregado em redes IEEE 802.15.4. A abordagem proposta toma decisões de escalonabilidade em tempo polinomial;
- Avaliação experimental das abordagens propostas através de cenários reais, considerando os sobrecustos (*overheads*) impostos pelo ambiente, as limitações físicas impostas pelo padrão e pelos dispositivos sensores. A avaliação experimental demonstra a factibilidade de implementação das soluções propostas em dispositivos sensores compatíveis com o padrão IEEE 802.15.4.

### 1.4 ORGANIZAÇÃO DO TEXTO

Esta tese é dividida em sete capítulos, incluindo esta introdução. O restante deste documento está organizado como segue:



O Capítulo 2 apresenta a fundamentação teórica sobre sistemas de tempo real e descreve-se de forma sucinta os aspectos mais relevantes para compreensão deste trabalho. Este capítulo também discorre as principais características sobre a especificação IEEE 802.15.4. As limitações estabelecidas pela especificação relacionadas à periodicidade e capacidade de transmissão de informações são discutidas.

O Capítulo 3 traz um estudo detalhado dos principais trabalhos relacionados a esta tese que foram encontrados na literatura. São abordados trabalhos referentes ao escalonamento de tarefas que toleram descartes de ativações, escalonamento de compartimentos de tempo garantido e testes de escalonabilidade.

O modelo de sistema delineado neste trabalho é apresentado no Capítulo 4.

O Capítulo 5 apresenta a abordagem de priorização de tarefas descentralizada aplicada durante períodos de acesso com contenção. Apresenta-se o algoritmo de escalonamento e a avaliação da proposta frente a outras proposta encontradas na literatura e discutidas no Capítulo 3.

No Capítulo 6 são apresentadas duas propostas de escalonamento de GTSs durante períodos de acesso sem contenção. Primeiramente, uma abordagem mais restritiva é apresentada e avaliada. A segunda proposta, relaxa as restrições impostas pela primeira abordagem oferecendo um teste de escalonabilidade mais eficiente, onde um maior conjunto de tarefas pode ser escalonado.

Finalmente, o Capítulo 7 apresenta as conclusões e introduz os possíveis desdobramentos sobre esta tese na forma de trabalhos relacionados.



## 2 FUNDAMENTAÇÃO TEÓRICA

O presente capítulo se destina a apresentar a fundamentação teórica necessária para a compreensão desta tese. Este capítulo está dividido em três partes: Primeiramente, são apresentados os principais conceitos e taxonomias relacionados aos sistemas de tempo real, posteriormente é apresentado uma introdução sobre o modelo de tarefas  $(m,k)$ -firm e por fim, uma descrição geral sobre a especificação IEEE 802.15.4 é apresentada.

### 2.1 SISTEMAS DE TEMPO REAL

Sistemas de Tempo Real (STR) são aqueles nos quais a correção do sistema não depende apenas do resultado lógico da computação, mas também do tempo o qual os resultados são produzidos (RAMAMRITHAM; STANKOVIC, 1994). Ou seja, os resultados devem ser produzidos dentro de intervalos de tempo bem definidos para que sejam considerados úteis. Esses sistemas abrangem um amplo escopo de aplicações envolvendo complexidades que vão desde sistemas embarcados em microcontroladores, até amplos e sofisticados sistemas distribuídos. Exemplos de utilização incluem controladores de voo, controle de tráfego aéreo, telecomunicações, controle de processos de plantas industriais e aplicações de multimídia distribuída.

Uma rede de sensores sem fio, por exemplo, pode ser concebida como um sistema computacional de tempo real, cujas restrições temporais surgem pela interação próxima com o ambiente. Este tipo de sistema precisa reagir a estímulos gerados externamente pelo ambiente dentro de prazos de tempo estipulados e finitos (BURNS; WELLINGS, 2001) (Figura 2). O tempo é um importante recurso gerido por sistemas de tempo real. Esses sistemas são constituídos por um conjunto de tarefas cooperantes com restrições temporais (usualmente, períodos e *deadlines*<sup>1</sup>) (MONTEZ, 1999).

Apesar do desempenho no processamento ser uma característica importante para a maioria dos sistemas de tempo real, uma questão importante diz respeito à previsibilidade temporal (SHIN; RAMANATHAN, 1994a), ou seja, em sistemas de tempo real, a previsibilidade indica se em tempo de projeto é possível obter algum tipo de antecipação do

---

<sup>1</sup>O termo *deadline* neste trabalho é considerado como prazo máximo no qual uma tarefa deve completar sua execução.

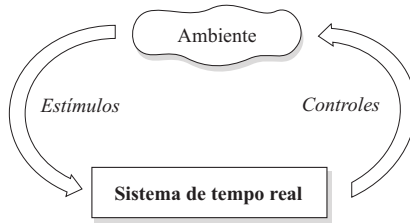


Figura 2: Exemplo de sistema de tempo real e seu ambiente.

comportamento temporal (STANKOVIC; RAMAMRITHAM, 1990). Para tanto, é necessário assumir algumas hipóteses de carga e faltas do sistema. Isto é, assumir no sistema quais serão os tempos máximos de utilização de recursos, a carga máxima em que o sistema é submetido e os tipos e frequência das faltas que podem ocorrer no sistema.

A previsibilidade determinista de um sistema pode ser obtida apenas em ambientes deterministas, onde é possível uma completa antecipação dos comportamentos do ambiente. Garantias de *deadlines* são possíveis apenas se existe o conhecimento prévio de algumas características das tarefas, como tempos máximos de computação e taxas de chegada. Além da dificuldade em se obter esses valores, ainda existe o problema de grande subutilização do sistema.

Quando não é factível a antecipação da carga e faltas que podem ocorrer no sistema, a única previsibilidade que pode ser obtida é a probabilista. O conhecimento do comportamento de um sistema pode ser adquirido através de simulações. No entanto, valores estimados a partir de simulações ou testes podem, em algumas situações, não corresponder aos valores reais obtidos durante uma execução do sistema.

As garantias no atendimento dos requisitos temporais fornecidos pelos STR podem ser classificadas como: garantia em tempo de projeto e garantia dinâmica. A primeira, relacionada com a obtenção de previsibilidade determinista do sistema, normalmente, é obtida através de análises *off-line* do comportamento do sistema e com adoção de esquemas de reservas de recursos. Alguns autores (SHIN; RAMANATHAN, 1994b) associam o conceito de previsibilidade a uma antecipação probabilista do comportamento do sistema (baseada em simulações ou estimativas) que estipulam probabilidades dos *deadlines* serem atendidos. A previsibilidade probabilista é utilizada em sistemas onde a carga computacional não pode ser previamente conhecida, logo não se consegue obter garantias em tempo de projeto sobre os atendimentos de todos os *deadlines*. Em alguns sistemas com previsibilidade probabi-

lista, o atendimento as restrições temporais segue o modelo de melhor esforço. Neste tipo de abordagem a dificuldade em prever a carga computacional, implica em eventuais perdas de *deadlines*. Porém garantias dinâmicas podem ser alcançadas em abordagens de melhor esforço através da adoção de mecanismos de teste de aceitação dinâmicos, assegurando o atendimento de algumas tarefas admitidas no sistema (MONTEZ, 1999).

Tradicionalmente os sistemas de tempo real são classificados conforme a rigidez (*strictness*) de seus *deadlines* (STANKOVIC; RAMAMRITHAM, 1990). Dependendo das consequências advindas da perda de um *deadline*, os sistemas de tempo real são usualmente classificados em duas categorias: *soft* e *hard* (BUTTAZZO, 1997). Um STR é dito ser *soft* se o cumprimento de seus *deadlines* é desejável por motivos de desempenho do sistema, porém a perda de alguns *deadlines* não causa danos sérios ao ambiente e não prejudica o comportamento correto do sistema (BUTTAZZO, 1997). Um sistema de tempo real é dito ser *hard* se uma falha no atendimento de um *deadline* pode ocasionar danos catastróficos e irreversíveis no ambiente controlado (perdas de vidas humanas, danos em equipamentos e perdas financeiras) (BUTTAZZO, 1997). Esta classe de sistemas de tempo real engloba sistemas aviônicos, robóticos, nucleares, plantas químicas, etc.

(LIU, 2000) define um STR *hard* como aquele onde são exigidas garantias no cumprimento dos *deadlines* das tarefas. De forma análoga, se não são exigidas garantias, apenas demonstrações de que a tarefa tenha algumas restrições atendidas de forma estatística o sistema é dito ser *soft*.

### 2.1.1 Modelo de tarefas

Uma aplicação de tempo real é constituída tipicamente de várias tarefas. Além da correção lógica, uma tarefa de tempo real deve satisfazer restrições temporais. Diferentes abordagens de tempo real se apoiam em diferentes modelos de tarefas. Porém, de uma forma geral, os modelos encontrados na literatura apresentam os seguintes parâmetros (BUTTAZZO, 1997):

- **Tempo de chegada**  $a_i$ : é o tempo no qual a tarefa torna-se pronta para execução<sup>2</sup>.

---

<sup>2</sup> Alguns autores consideram que o tempo de chegada de uma tarefa (*arrival time*) é diferente do tempo de liberação (*release time*) onde a tarefa torna-se pronta para

- **Tempo de computação**  $C_i$ : é o tempo necessário para o processador executar a tarefa sem interrupções. Sistemas *hard* costumam trabalhar com piores casos de tempo de execução (*worst case execution time* - WCET).
- **Deadline**  $d_i$ : é o instante de tempo no qual uma tarefa deve ser completada (*deadline* absoluto).
- **Deadline**  $D_i$ : é o intervalo de tempo no qual uma tarefa deve ser completada (*deadline* relativo).
- **Tempo de início**  $s_i$ : é o tempo no qual uma tarefa inicia sua execução.
- **Tempo de término**  $f_i$ : é o tempo no qual uma tarefa finaliza sua execução.
- **Criticalidade**: parâmetro relacionado com as consequências de perda de um *deadline* (tipicamente, um valor numérico que define se a tarefa é *hard* ou *soft*).
- **Importância**  $v_i$ : representa a importância relativa da tarefa com respeito as outras tarefas do sistema.
- **Folga**  $L_i$ :  $L_i = d_i - f_i$ , representa o tempo máximo que uma tarefa pode ser postergada para completar sua ativação dentro de seu *deadline* (considerando que C representa o pior caso de tempo de execução).

Alguns dos parâmetros apresentados acima são graficamente ilustrados na Figura 3.

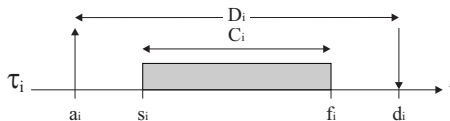


Figura 3: Parâmetros típicos de uma tarefa de tempo real.

Outra característica, relacionada com o comportamento temporal das tarefas de tempo real diz respeito à regularidade de suas

---

execução (XU; PARNAS, 1993). Esses autores geralmente trabalham com análises de escalonamento que consideram mais um parâmetro denominado *jitter* de liberação (*release jitter*).

ativações. Tarefas podem ser definidas como periódicas ou aperiódicas. Tarefas periódicas consistem em uma sequência infinita de tarefas idênticas, chamadas de instâncias ou *jobs*, com tempos de chegadas com taxas constantes. Tarefas aperiódicas (ou assíncronas) assim como as tarefas periódicas, consistem em uma sequência infinita de *jobs*, entretanto, seus tempos de chegada não são regulares.

O período de uma tarefa  $\tau_i$  é comumente denotado por  $T_i$ . Em alguns modelos de tarefas mais simples, considera-se que o *deadline* relativo  $D_i$  da tarefa coincide com o final do período ( $T_i = D_i$ ). Essa suposição geralmente simplifica e facilita a análise de escalonabilidade desses sistemas.

### 2.1.2 *Deadlines* e funções benefício

A reserva de recursos em STR *hard*, considerando os piores casos de execução em tempo de projeto, geralmente leva à subutilização dos recursos do sistema, este é o preço a ser pago por sistemas onde existe a necessidade de garantias. Porém, em STR *soft*, onde não existe o oferecimento de garantias em tempo de projeto, tarefas podem receber atendimento baseado nos tempos médios de computação ou ainda, permitir que algumas ativações de suas tarefas sejam descartadas em virtude de alguma tarefa executar mais do que o tempo estimado. Este tipo de sistema consegue uma melhor utilização dos recursos do sistema, durante a ocorrência de eventuais situações de sobrecarga transientes ou falhas temporais (MONTEZ, 1999).

Objetivando a adoção de modelos mais flexíveis, abordagens de escalonamento adaptativo assumem que as condições de carga do sistema são monitoradas e as decisões de escalonamento são baseadas na observação destas condições (LIU, 2000). As abordagens de escalonamento adaptativo, são geralmente empregadas com o objetivo de obter-se uma degradação suave da qualidade oferecida pelo sistema durante situações onde a carga do sistema não pode ser previamente identificada.

A qualidade recebida por uma tarefa de tempo real pode ser avaliada através de um valor de qualidade (representado na Figura 4 por  $V(t_i)$ ). Este valor depende do tempo o qual a tarefa é completada. Assim, a qualidade recebida pelas tarefas de tempo real pode ser modelada através de uma função benefício (ilustrada na Figura 4).

Uma tarefa não tempo real adiciona valor de qualidade constante ao sistema, uma vez que esta classe de sistema não apresenta valores

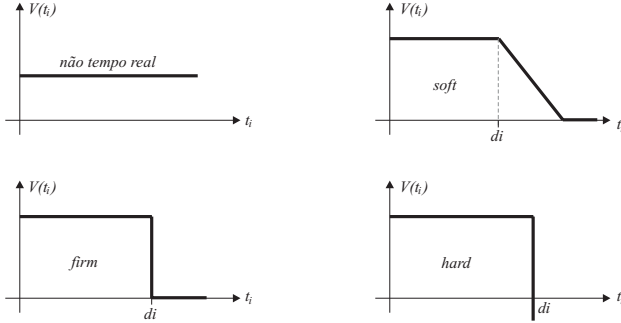


Figura 4: Funções benefício associadas a uma tarefa tempo real.

de *deadlines*. Já uma tarefa *hard*, o não cumprimento de um *deadline* adiciona valor de benefício negativo ao sistema, prejudicando todo o sistema. No caso de uma tarefa *soft*, um valor de benefício é adicionado ao sistema, mesmo após o seu *deadline*. Este valor de benefício diminui com o passar do tempo. Finalmente, uma tarefa *firm* não prejudica o sistema caso perca seu *deadline*, porém o valor de benefício adicionado passa a ser zero, caso a tarefa não seja completada dentro de seu *deadline*.

É importante ressaltar que caso os *deadlines* das tarefas *hard* não sejam atendidos o valor benefício passa a ser infinitamente negativo, desta forma é necessário adoção de mecanismos de escalonamento garantidos para as tarefas *hard*. Se todas as tarefas *hard* têm atendimento garantido, o algoritmo de escalonamento deve garantir o escalonamento das tarefas em condições de carga moderada e maximizar o atendimento de tarefas *firm* e *soft* durante sobrecargas transientes (BUTTAZZO, 1997).

Alguns trabalhos da literatura classificam como *firm* os sistemas de tempo real formados exclusivamente por tarefas com *deadlines firm*. Estes sistemas permitem que alguns *deadlines* ocasionalmente sejam violados, resultando em degradação na qualidade do sistema (MARCHAND; S.-CHETTO, 2006). Exemplo desta classe de sistemas de tempo real são aplicações multimídia, onde perdas na qualidade de áudio e vídeo ocorrem quando o sistema não é capaz de processar todos os quadros a tempo. Dentro deste contexto, algumas pesquisas têm considerado outras informações para melhorar a utilização do sistema, incorporando técnicas de escalonamento que consideram não apenas atributos temporais das tarefas, mas também alguns aspectos de qualidade.



### 2.1.3 Escalonamento tempo real

Uma das áreas de pesquisas em sistemas de tempo real que tem recebido grande atenção dos pesquisadores tem sido o escalonamento de tarefas. Sistemas de tempo real possuem restrições temporais (*deadlines*) que devem ser atendidas de forma previsível e robusta (FARINES et al., 2000). Estas necessidades demandam o projeto de algoritmos de escalonamento que atendam os requisitos do sistema de tempo real de maneira satisfatória.

O processo de escalonamento de um conjunto de tarefas pode, muitas vezes, ser dividido em duas etapas. Inicialmente um teste de escalonabilidade determina se é possível atender os *deadlines* de todas as tarefas do conjunto. Caso o conjunto de tarefas possa ser escalonado cumprindo suas restrições temporais (ex. *deadlines*), o conjunto de tarefas é dito ser factível (*feasible*). Os algoritmos que realizam testes de escalonabilidade são classificados em (OLIVEIRA, 1997):

- Suficiente mas não necessário: todo o conjunto de tarefas admitido pelo teste é escalonável, porém nada pode ser afirmado a respeito dos conjuntos de tarefas reprovados pelo teste.
- Necessário mas não suficiente: todo o conjunto de tarefas reprovado não é escalonável, porém nada pode ser dito sobre os conjuntos aprovados.
- Exato: todo o conjunto de tarefas aprovado pelo teste é escalonável e todo conjunto reprovado não é escalonável.

A segunda etapa corresponde em calcular a escala de execução das tarefas, ou seja, determinar qual tarefa utiliza qual recurso a cada momento. Um escalonador (*scheduler*) provê um algoritmo ou política para ordenar a execução de um conjunto de processos (ou tarefas) no processador de acordo com algum critério pré-estabelecido. Dentre as taxonomias propostas para classificar algoritmos de escalonamento tempo real encontrados na literatura, destacam-se as seguintes (BUTTAZZO, 1997):

- Quanto à capacidade de preempção:
  - Preemptivo. Nesta classe, uma tarefa em execução pode ser interrompida a qualquer momento para que o processador seja atribuído para outra tarefa, de acordo com uma política de escalonamento predefinida.

- Não-preemptivo. Um algoritmo é dito ser não-preemptivo quando, uma vez executada, uma tarefa utiliza o processador até completar sua execução. Assim, todas as decisões de escalonamento são tomadas após o término de execução de uma tarefa.
- Quanto ao tempo de decisão sobre parâmetros de escalonamento:
  - Estático. Algoritmos estáticos são aqueles onde as decisões de escalonamento são baseadas em parâmetros fixos (prioridades), atribuídos para as tarefas antes de suas ativações.
  - Dinâmico. Algoritmos dinâmicos são aqueles onde as decisões de escalonamento são baseadas em parâmetros dinâmicos que podem mudar durante a evolução do sistema.
- Quanto ao tempo de execução do escalonamento:
  - Off-line. Um algoritmo de escalonamento é dito ser *off-line* se a execução de todo o conjunto de tarefas é realizada antes da ativação efetiva das tarefas. A escala gerada é armazenada em uma tabela e posteriormente é executada.
  - On-line. Um algoritmo de escalonamento é considerado *on-line* se as decisões de escalonamento são tomadas em tempo de execução, sempre que uma tarefa é admitida no sistema ou quando uma tarefa em execução é finalizada.
- Quanto à optimalidade do escalonamento:
  - Ótimo. Um algoritmo de escalonamento é considerado ótimo se, caso ele falhe no atendimento de um *deadline*, então nenhum outro algoritmo de escalonamento da mesma classe é capaz de atender o *deadline*.
  - Heurístico. Um algoritmo heurístico tenta oferecer uma escala ótima para o conjunto de tarefas, porém não oferece garantias quanto a encontrar esta escala ótima.

Uma classe de algoritmos de escalonamento também considerada na literatura são os algoritmos clarividentes (*clairvoyant*). Um algoritmo é dito ser clarividente se ele conhece o futuro *a priori* (BUTTAZZO, 1997), isto é, se ele conhece antecipadamente os tempos de chegada de todas as tarefas. Como tal algoritmo na realidade não existe, eles são utilizados apenas para comparação de desempenho frente outros algoritmos.

## 2.1.4 Escalonamento de tarefas periódicas

Sistemas de tempo real são geralmente formados por aplicações periódicas e não periódicas. Tradicionalmente, aplicações de controle ou até mesmo aplicações multimídia são caracterizadas pelo comportamento periódico de suas atividades. Em muitas aplicações industriais (ex. controle de processos), as tarefas periódicas representam a maior demanda computacional do sistema (BUTTAZZO, 1997).

As características cíclicas das tarefas periódicas, permitem que os tempos de chegada e a carga computacional sejam conhecidos *a priori*, favorecendo a obtenção de garantias em tempo de projeto sobre a escalonabilidade do conjunto de tarefas periódico.

Dentro deste contexto, uma tarefa periódica  $\tau_i$  é dita ser factível (*feasible*) se todas as suas instâncias são atendidas dentro de seus *deadlines*. Um conjunto de tarefas  $\Gamma$  é dito ser escalonável se todas as tarefas em  $\Gamma$  são factíveis (BUTTAZZO, 1997).

Dado um conjunto  $\Gamma$  composto por  $n$  tarefas periódicas, o fator de utilização do processador  $U$  é a fração de tempo de processador gasto na execução do conjunto de tarefas. Considerando  $C_i/T_i$  a fração de tempo de processador consumida pela tarefa  $\tau_i$ , o fator de utilização para  $n$  tarefas é dada pela equação 2.1. O fator de utilização do processador provê uma indicação da carga computacional imposta ao processador pelo conjunto de tarefas periódicas.

$$U = \sum_{i=1}^N \frac{C_i}{T_i} \quad (2.1)$$

Neste capítulo, três algoritmos clássicos para lidar com tarefas periódicas são brevemente discutidos: Taxa Monotônica (RM, *Rate Monotonic*), EDF (*Earliest Deadline First*) e *Deadline* Monotônico (DM, *Deadline Monotonic*). Testes de escalonabilidade são apresentados em cada algoritmo com a finalidade de prover garantias para um conjunto de tarefas.

### *Escalonamento Taxa Monotônica*

Em algoritmos de escalonamento baseados em prioridade fixa, todas as instâncias de uma tarefa possuem a mesma prioridade. Usualmente, as tarefas são numeradas de tal forma que  $\tau_i$  possui prioridade  $i$ , onde os valores menores denotam as maiores prioridades e valores

elevados denotam prioridades baixas.

O algoritmo de escalonamento Taxa Monotônica (RM) atribui prioridades de acordo com as taxas de chegada das tarefas. Tarefas com altas taxas de chegada (menores períodos  $T_i$ ) recebem as maiores prioridades do sistema. Como os períodos das tarefas são constantes, as prioridades atribuídas às tarefas não mudam e o algoritmo RM é considerado um algoritmo de prioridade fixa, pois as prioridades são atribuídas de forma estática antes da execução das tarefas. O RM pertence à classe de algoritmos preemptivos, a execução de uma tarefa é suspensa com a chegada de uma nova tarefa com menor período.

Em um trabalho clássico, datado de 1973, os autores (LIU; LAYLAND, 1973) apresentam o conceito de instante crítico. O instante crítico para uma tarefa é o instante de tempo o qual o tempo de resposta da tarefa é maximizado – ou excede o *deadline*. No referido trabalho os autores apresentam um teorema demonstrando que, para um conjunto de tarefas periódicas com prioridades fixas, o instante crítico para uma tarefa  $\tau_i$  ocorre quando esta é liberada juntamente com todas as tarefas  $\tau_j$  ( $j < i$ ) de maior prioridade. O intervalo de 0 até  $D_i$  é então um intervalo sobre o qual a demanda das tarefas mais prioritárias  $\tau_1, \dots, \tau_{i-1}$  é máxima, sendo esta a pior situação possível para que  $\tau_i$  tenha seu *deadline* atendido.

Baseado no conceito de instante crítico (LIU; LAYLAND, 1973) mostram que o RM é um algoritmo de escalonamento ótimo considerando todos os algoritmos de prioridades fixas. Os autores apresentam um teste de escalonabilidade suficiente, mas não necessário, derivado do fator de utilização do processador. O teste proposto é resumido pela Equação 2.2.

$$U = \sum_{i=1}^N \frac{C_i}{T_i} \leq n \left( 2^{\frac{1}{n}} - 1 \right) \quad (2.2)$$

O algoritmo RM garante que um conjunto de tarefas periódicas é escalonável se a utilização total do processador  $U$  não ultrapassar 0.69. Este teste se torna exato nos casos onde as tarefas do conjunto a ser escalonado apresentam períodos  $T_i$  múltiplos de suas tarefas mais prioritárias (LIU, 2000). Neste caso, o teste passa a ser necessário e suficiente podendo alcançar o fator de utilização máximo ( $U = 100\%$ ), conforme definido na Equação 2.3.

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

Importante ressaltar que os testes 2.2 e 2.3 são válidos apenas para um modelo de tarefas bastante restrito, onde as tarefas não possuem relações de precedência e de exclusão mútua. Além disso, considera-se que os *deadlines* relativos coincidem com os períodos. Existem outros trabalhos mais recentes que relaxam essas restrições ao modelo de tarefas e propõem testes de escalonabilidade exatos, porém mais complexos que os apresentados pelas equações 2.2 e 2.3 (AUDSLEY et al., 1993; TINDELL et al., 1993).

#### *Escalonamento Earliest Deadline First*

O EDF é um algoritmo de escalonamento preemptivo e dinâmico, que seleciona as tarefas para serem escalonadas conforme seus valores de *deadlines* absolutos. Segundo o EDF, a tarefa com o menor *deadline* recebe a maior prioridade do sistema. O EDF é considerado ótimo dentre a classe dos algoritmos de prioridades dinâmicas. Assim como no RM, as premissas do trabalho original de (LIU; LAYLAND, 1973) que determinavam o modelo de tarefas no EDF são que os *deadlines* relativos coincidem com o período e que não há relações de precedência e exclusão mútua.

Assumindo as premissas acima, a escalonabilidade de um conjunto de tarefas segundo o EDF pode ser verificado através do fator de utilização do processador. Desta forma, um conjunto de tarefas periódicas é escalonado pelo EDF se:

$$U = \sum_{i=1}^N \frac{C_i}{T_i} \leq 1 \quad (2.4)$$

O teste de escalonabilidade (Equação 2.4) é suficiente e necessário, porém caso uma das premissas acima sejam violadas, o teste passa a ser apenas necessário e deixa de ser suficiente. Existem outros trabalhos que relaxam o modelo de tarefas e que apresentam análise de escalonabilidade mais complexas, que não são baseadas no conceito de utilização (LEHOCZKY, 1990).

### Escalonamento Deadline Monotônico

Em meados dos anos 80, testes de escalonabilidade mais complexos começaram a ser desenvolvidos. O algoritmo *Deadline Monotônico*, proposto por (LEUNG; WHITEHEAD, 1982), é um algoritmo de escalonamento por prioridades fixas que relaxa a premissa imposta pelo RM, onde os *deadlines* relativos das tarefas coincidem com o final dos períodos. Desta forma, tem-se *deadlines* relativos menores ou iguais aos valores dos períodos das tarefas  $D_i \leq T_i$  (Figura 5).

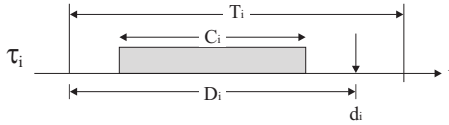


Figura 5: Parâmetros típicos para escalonamento DM.

Segundo o DM, cada tarefa recebe uma prioridade com valor inversamente proporcional a seu *deadline* relativo. Consequentemente, em um dado instante a tarefa pronta com o menor *deadline* relativo é executada. Similar ao RM, o DM é intrinsecamente preemptivo, uma tarefa em execução é suspensa pela chegada de uma tarefa com *deadline* relativo menor. O DM é também considerado ótimo em sua classe de algoritmos com *deadlines* menores que os períodos das tarefas. Na verdade, pode-se considerar o RM como um caso particular do DM, onde os *deadlines* relativos coincidem com os períodos das tarefas.

O trabalho apresentado por (LEHOCZKY et al., 1989) assume que o instante crítico para um conjunto de tarefas ocorre no instante zero – todas as tarefas são liberadas simultaneamente em  $t = 0$ . Objetivando determinar quando uma tarefa pode cumprir seus *deadlines* o teste calcula a demanda de processador de uma tarefa  $\tau_i$  juntamente com as demandas de processador das tarefas  $\tau_j$  de maior prioridade, como uma função de tempo a partir do instante crítico. Desta forma, considerando um conjunto de tarefas liberadas simultaneamente no instante zero, a tarefa  $\tau_i$  terá seu *deadline* atendido se existe um instante de tempo  $0 < t \leq T_i$  tal que a demanda de processador,  $W_i(t)$  é menor ou igual a  $t$ . A demanda de tempo  $W_i(t)$  da tarefa  $\tau_i$  liberada em  $[0,t)$  é dado pela Equação 2.5.

$$W_i(t) = C_i + \sum_{j=1}^{i-1} \left\lceil \frac{t}{T_j} \right\rceil C_j, \quad \text{para } 0 < t \leq T_i \quad (2.5)$$

Como  $(t/T_j)$  é um valor crescente, exceto nos instantes onde as tarefas são liberadas, os únicos valores de  $t$  que precisam ser testados são os múltiplos dos períodos das tarefas entre zero e  $T_i$ .

Em (LEHOCZKY, 1990) é introduzido o conceito de período de ocupação (*busy period*). Considerando que o instante crítico de uma tarefa  $\tau_i$  ocorre em  $t_0$ , o período de ocupação de nível  $i$  ocorre no primeiro instante  $t$ , após  $t_0$ , quando todas as ativações da tarefa  $\tau_i$  que foram liberadas em  $t_0$  são completadas. Em outras palavras, no intervalo  $[0,t)$ , o processador está ocupado executando ativações de tarefas  $i$  ou de maior prioridade. Todas as instâncias executadas dentro do intervalo  $[0,t)$  são liberadas no intervalo e ao término do intervalo não há carga adicional de instâncias a serem executadas posteriormente.

Desta forma, o pior caso de tempo de resposta para  $\tau_i$  está associado com sua primeira instância, assim apenas o primeiro *deadline* da tarefa  $\tau_i$  precisa ser verificado. Este resultado permanece válido quando  $D_i \leq T_i$ . Entretanto, uma análise mais criteriosa deve ser realizada quando considera-se o *deadline* de uma tarefa maior do que seu período ( $D_i > T_i$ ). Neste caso, o pior caso de tempo de resposta da tarefa não ocorre necessariamente na primeira ativação da tarefa. Desta forma, como o tempo de resposta da tarefa pode ser maior do que seu período, pode haver mais de uma instância pronta para execução em um instante de tempo. Instâncias de tarefas prontas para execução são usualmente escalonadas de acordo com a política de escalonamento FIFO.

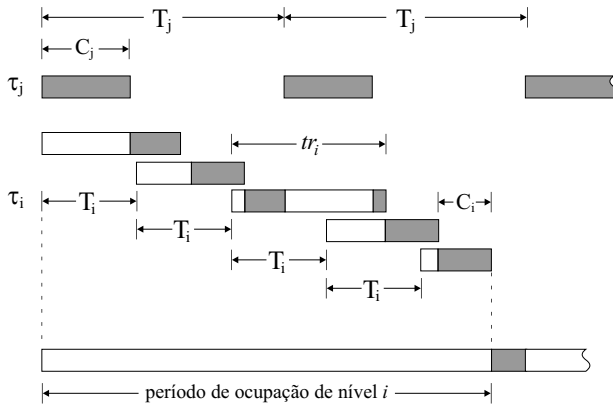


Figura 6: Cálculo do período de ocupação de nível  $i$  ( $D_i > T_i$ ).

Lehoczky (LEHOCZKY, 1990) demonstra como o pior caso de

tempo de resposta de uma tarefa  $\tau_i$  pode ser encontrado através da análise de um número de janelas, cada uma definida como sendo a duração do período de ocupação iniciando na janela, e cada janela a partir de uma chegada de tarefa  $\tau_i$  (a partir deste momento em algum múltiplo de  $T_i$  antes da ativação atual da tarefa  $\tau_i$ ). A Figura 6 apresenta parte da execução de duas tarefas: a tarefa  $\tau_j$  é a tarefa de mais alta prioridade e a tarefa  $\tau_i$  de prioridade mais baixa (novas ativações podem ocorrer antes mesmo de suas ativações anteriores terem sido atendidas). Para encontrar o pior caso de tempo de resposta, todas as ativações da tarefa  $\tau_i$  dentro do período de ocupação de nível  $i$  devem ser examinadas (na Figura 6 a terceira ativação de  $\tau_i$  apresenta o pior tempo de resposta).

Outro grupo de pesquisadores analisou o problema de determinar o pior caso de tempo de resposta de uma tarefa; isto é, encontrar o maior tempo entre a chegada de uma tarefa e seu término de execução, de uma forma mais genérica. Uma vez conhecido o pior caso de tempo de resposta de uma tarefa, a factibilidade de execução da tarefa pode ser verificada comparando-se o pior caso de tempo de resposta com seu respectivo *deadline*.

Em (JOSEPH; PANDYA, 1986; AUDSLEY et al., 1991) é apresentado um teste de escalonabilidade necessário e suficiente para o DM. O teste é baseado no conceito de tempo de resposta das tarefas. O cálculo do tempo de resposta considera o intervalo de uma janela de tempo  $R_i$  que é o tempo transcorrido desde a liberação da tarefa  $\tau_i$  até o término de sua execução. O cálculo do tempo de resposta considera ainda a interferência das tarefas de maior prioridade (considerando o instante crítico das tarefas). A fórmula do tempo de resposta  $R_i$  é descrita na equação 2.6.

$$R_i = C_i + \sum_{j=1}^{i-1} \left\lceil \frac{R_i}{T_j} \right\rceil C_j \quad (2.6)$$

Uma tarefa  $\tau_i$  tem suas restrições atendidas sempre que  $R_i \leq D_i$ . A verificação do tempo de resposta ocorre através de um teste iterativo. O teste é finalizado quando os valores de  $R_i^{n+1} = R_i^n$ . Este teste de escalonabilidade apesar de ser mais complexo que os apresentados nas equações 2.2, 2.3 e 2.4, possui como vantagem a capacidade de facilmente incorporar extensões ao modelo de tarefas, permitindo modelos de tarefas com *jitter* de liberação, relações de precedência, de exclusão mútua, etc.



### 2.1.5 Escalonamento de tarefas aperiódicas

Nem todas as tarefas das aplicações de tempo real chegam em intervalos de tempo regulares. Algumas são ativadas pela ocorrência de determinados eventos. Esta classe de tarefas é referenciada na literatura de tempo real como tarefas aperiódicas (SPRUNT, 1990). Um exemplo é uma tarefa que dispara um alarme. Este evento não ocorre em intervalos de tempo conhecidos e a taxa de chegada da tarefa não é conhecida.

Geralmente, tarefas periódicas são consideradas críticas, necessitando testes de escalonamento em tempo de projeto, para que o sistema possa prover garantias temporais. Algumas tarefas aperiódicas podem necessitar garantias em seus *deadlines* (caso as tenha). Tarefas aperiódicas que apresentam intervalo mínimo entre ativações e possuem um *deadline* associado são denominadas tarefas esporádicas (FARINES et al., 2000).

### 2.1.6 Escalonamento na sobrecarga

O problema de escalonamento tempo real em condições de sobrecarga tem sido objeto de intensa pesquisa durante os últimos anos. Uma situação de sobrecarga pode ser definida como uma situação crítica no qual a demanda pelos recursos computacionais pelas tarefas excede a capacidade de processamento do sistema. Desta forma, nessas situações não é possível que todos os *deadlines* das tarefas sejam atendidos (BUTTAZZO et al., 1995).

Mesmo quando um sistema de tempo real é adequadamente projetado, eventuais situações de sobrecarga podem ser experimentadas por estes sistemas. Uma situação de sobrecarga transiente pode ocorrer em virtude de uma série de fatores, dentre eles destacam-se: mudanças no ambiente controlado, chegadas simultâneas de eventos assíncronos, falhas de periféricos ou exceções do sistema. A ocorrência destes eventos podem, eventualmente comprometer o atendimento de alguma tarefa crítica, prejudicando o comportamento do sistema como um todo.

Quando um sistema está sobrecarregado, não é possível que todos os *deadlines* sejam atendidos. Desta forma, o objetivo do algoritmo de escalonamento passa a ser garantir o atendimento ao menos das requisições mais importantes. Conforme discutido no início desta Seção, objetivando estabelecer um nível de importância as tarefas, podem ser atribuídos valores para parâmetros que reflitam a importância das ta-

refas, este parâmetro concebido como prioridade ordena o atendimento das tarefas em uma eventual condição de sobrecarga do sistema.

Infelizmente, em condições de sobrecarga não existem algoritmos de escalonamento *on-line* ótimos, desta forma, as decisões de escalonamento devem ser dirigidas por algoritmos dinâmicos e adaptativos, os quais podem, por exemplo, atender os *deadlines* das tarefas mais importantes.

Alguns trabalhos encontrados na literatura procuram, através de algoritmos de escalonamento heurísticos melhorar o comportamento do sistema em condições de sobrecarga (KOREN; SHASHA, 1992; BUTTAZZO; STANKOVIC, 1993; KOREN; SHASHA, 1995; RAMANATHAN, 1999). Uma possível classificação dos algoritmos de escalonamento encontrados na literatura por ser realizada considerando-se três grupos (BUTTAZZO, 1997):

1. **Melhor esforço:** Esta classe de algoritmo não adota nenhum mecanismo de controle de admissão. Desta forma, mesmo que o sistema encontre-se sobrecarregado, a chegada de uma nova tarefa ao sistema é admitida. O escalonador utiliza apenas mecanismos de atribuição de prioridades para escalonar as tarefas, conforme ilustrado na Figura 7.

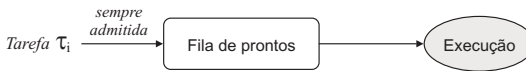


Figura 7: Escalonamento melhor esforço.

2. **Garantidos:** Esta classe de algoritmos possui mecanismo de controle de admissão, onde uma tarefa só é admitida no sistema após a realização de um teste de escalonabilidade (considerando o pior caso de execução das tarefas), caso uma tarefa não seja aceita pelo teste de aceitação a tarefa é descartada (Figura 8).



Figura 8: Escalonamento garantido.

3. **Robustos:** Esta classe de algoritmos considera duas políticas de escalonamento: a primeira para aceitação e a segunda para rejeição das tarefas. Com a chegada de uma requisição ao sistema,

um teste de aceitação verifica a escalonabilidade do sistemas (novamente, considerando o pior caso de execução das tarefas). Se o conjunto ainda é escalonável com a aceitação da nova tarefa, então a tarefa é admitida e inserida na fila de pronto; caso contrário, uma ou mais tarefas são rejeitadas, baseadas em alguma política. Uma possível política de descarte a ser adotada seria efetuar os descartes baseado nas prioridades das tarefas. A Figura 9 ilustra este modelo de escalonamento.

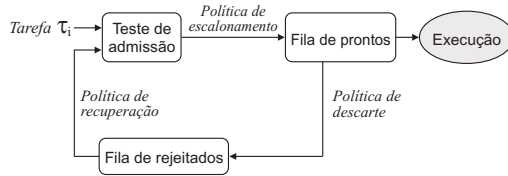


Figura 9: Escalonamento robusto.

O modelo garantido evita a ocorrência do efeito dominó<sup>3</sup> através de descartes de requisições. O teste de admissão funciona como um filtro quando o sistema encontra-se sobrecarregado, evitando que a admissão de uma nova requisição prejudique as requisições já admitidas pelo sistema. No entanto, é importante ressaltar que este tipo de garantia não é eficiente durante condições de sobrecarga transientes, pois não consideram a importância das tarefas, simplesmente descartam as tarefas recém chegadas no sistema. No modelo robusto, um mecanismo de recuperação é adicionado, assim uma tarefa que foi descartada pelo teste de admissão pode ser temporariamente mantida em uma fila de rejeitados, podendo, eventualmente ser atendida quando uma requisição conclui sua execução antes do seu *deadline* (antes do pior caso).

### 2.1.7 Abordagens adaptativas

É possível encontrar na literatura diferentes propostas de escalonamento para os sistemas de tempo real. A Figura 10 apresenta uma taxonomia com cinco abordagens básicas para o escalonamento de sistemas de tempo real, divididas em dois grupos, de acordo com o tipo

<sup>3</sup>Este fenômeno ocorre quando a perda de *deadline* de uma tarefa causa a perda de *deadline* de todas as tarefas subsequentes. O EDF é um algoritmo suscetível a ocorrência do efeito dominó em condições de sobrecarga, devido ao fato do EDF atribuir prioridades altas para as tarefas que estão mais próximas de perderem seus *deadlines*.

de garantia oferecida pela abordagem (OLIVEIRA, 1997). O primeiro grupo de abordagens (com garantia em tempo de projeto) apresenta abordagem com previsibilidade determinista, empregada em sistemas onde é necessário garantir em tempo de projeto, que todas as tarefas terão seus *deadlines* atendidos. Nas abordagens contidas no segundo grupo, não existe garantia fornecida em tempo de projeto.

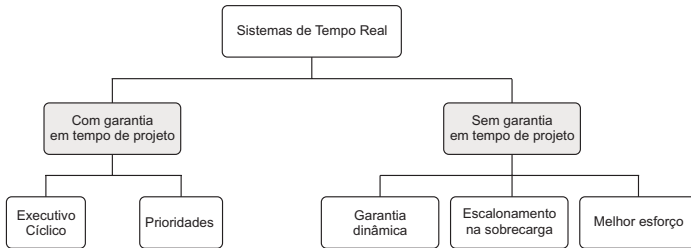


Figura 10: Abordagens de escalonamento tempo real (adaptado de (OLIVEIRA, 1997)).

Abordagens que fornecem garantias em tempo de projeto implicam na necessidade de recursos suficientes para a execução do pior caso de carga computacional. Isso pode representar uma enorme subutilização de recursos. Em abordagens estáticas como o executivo cíclico, as escalas são testadas e construídas considerando a carga computacional máxima. Estas escalas são montadas considerando os tempos máximos das tarefas e a ocorrência das tarefas esporádicas com intervalos mínimos entre ativações definidos. Essa abordagem pode ser extremamente pessimista se o pior caso for distante do caso médio de execução.

Uma outra abordagem para prover garantias em tempo de projeto é a que executa o teste de escalonabilidade em tempo de projeto, e produz a escala em tempo de execução. Nesta abordagem valores de prioridades são atribuídos às tarefas, e em tempo de execução o escalonador preemptivo é responsável por montar a escala conforme as prioridades das tarefas. A atribuição de prioridades às tarefas pode ser feita de forma estática em tempo de projeto ou dinâmica em tempo de execução.

Na atribuição estática de prioridades as tarefas vão recebendo os valores de prioridades durante o teste de escalonabilidade, obedecendo a critérios pré-definidos. Um exemplo de atribuição estática de prioridade são as políticas Taxa Monotônica e *Deadline* Monotônico citados na Subseção 2.1.4.

Na atribuição dinâmica de prioridades, em determinados momentos o escalonador verifica a condição temporal das tarefas, e atribui prioridades às tarefas, escolhendo a tarefa mais prioritária para ser executada naquele momento. Um exemplo clássico é o algoritmo EDF apresentado na Subseção 2.1.4.

Dentro do segundo grupo de abordagens (sem garantia em tempo de projeto), existe a classe de abordagens com garantia dinâmica. Nessas abordagens, testes de escalonabilidade (conhecidos também como testes de aceitação) e a produção de escalas são realizadas em tempo de execução, uma vez admitida no sistema, a tarefa admitida e todas as outras possuem garantia de que terão recursos disponíveis para sua execução. O trabalho apresentado por (STANKOVIC; RAMAMRITHAM, 1991) é um exemplo de uso deste tipo de abordagem. Ainda no conjunto de abordagens com garantias dinâmicas, temos as abordagens com garantia estatística (ou probabilista) (ATLAS; BESTAVROS, 1998), onde o comportamento temporal do sistema pode ser estimado, informando quais *deadlines* serão ou não atendidos.

Durante condições de sobrecarga, o escalonador dinâmico deve sacrificar algum tipo de parâmetro do conjunto de tarefas. Uma abordagem para tratar sobrecarga é o sacrifício do prazo de execução, onde a tarefa é aceita no conjunto, ainda que algumas tarefas percam seus *deadlines*. Esse tipo de abordagem considera um modelo de tarefas onde existe benefício mesmo que a tarefa complete sua execução após seus *deadlines*, conforme apresentado na Seção 2.1.2. O trabalho (BUTTAZZO et al., 1995) apresenta um exemplo de uso desta abordagem.

Outra abordagem possível durante condições de sobrecarga é o sacrifício da tarefa. Onde quando uma nova tarefa chega no sistema, um teste de admissão é executado verificando se o novo conjunto de tarefas é escalonável. Caso o novo conjunto não seja escalonável, alguma tarefa do conjunto é descartada<sup>4</sup>. Existem alguns trabalhos na literatura que propõem o uso de abordagens de sacrifício de tarefas: *Skip-over* (KOREN; SHASHA, 1995), *(m,k)-firm* (HAMDAOUI; RAMANATHAN, 1995a) e *(n/m)-hard* (BERNAT; BURNS, 1997).

Na abordagem de sacrifício de tempo de execução, cada tarefa do sistema possui uma parte obrigatória e uma parte opcional. A parte mandatória da tarefa é capaz de gerar resultados com a qualidade mínima exigida pelo sistema. A parte opcional, refina esse resultado até que o sistema obtenha a qualidade desejada. O trabalho clássico

---

<sup>4</sup>Geralmente a tarefa recém chegada é descartada devido ao custo computacional necessário para a realização de um novo teste de escalonabilidade, tendo em vista que o conjunto já era escalonável antes da chegada da nova tarefa.

apresentado por (LIU et al., 1991) introduz o conceito de computação imprecisa.

Algumas tarefas periódicas em sistemas de controle de processos, podem operar de maneira satisfatória quando realizam amostragem dos sinais de controle dentro de faixas de períodos específicos. Um melhor desempenho é oferecido ao sistema quando as taxas de amostragem destas tarefas são maiores, em contrapartida um limite inferior sobre a frequência dessas operações de amostragem é estabelecido, abaixo do qual, o sistema passa a executar de maneira insatisfatória. O modelo elástico proposto por (BUTTAZZO et al., 1998) propõem alteração nas frequências das tarefas visando manter a escalonabilidade do sistema. Desta forma, um conjunto de tarefas poderia passar a ser escalonável, se as frequências de algumas tarefas fossem reduzidas.

## 2.2 MODELO DE TAREFAS (M,K)-FIRM

Aplicações de tempo real periódicas assumem, usualmente, que todas as suas ativações serão executadas, ou seja, que todos os seus *deadlines* serão atendidos a cada período. No caso de aplicações tempo real que envolvem envio de mensagens via rede, isso implica que todas as mensagens enviadas precisam alcançar seus destinos antes de seus *deadlines*. Por outro lado, existem aplicações em que os descartes de algumas mensagens podem ser tolerados ou compensados, desde que estes descartes ocorram de forma esparsa. Diversos trabalhos exploram estas características (SCHENATO et al., 2007; LEMMON et al., 2003; HESPANHA et al., 2007; SEMPREGOM et al., 2010; WILLIG, 2008).

A tolerância de uma aplicação em perder *deadlines* pode ser representada através de um valor de percentagem de perda permitida. Por exemplo, um fluxo de vídeo pode especificar uma taxa tolerável de perda de *deadlines* de 10%. Entretanto, esta especificação implicitamente assume que as perdas de *deadlines* são adequadamente espaçadas. Ou seja, as perdas de *deadline* não deveriam ocorrer de forma consecutiva. Como exemplo, a perda de 25 *deadlines* em um horizonte de 300 ativações, respeita o limite máximo de 10% de perdas de *deadline*. No entanto, caso estas perdas sejam consecutivas, as consequências destas para a aplicação podem ser intoleráveis.

Objetivando especificar precisamente valores de intervalos entre *deadlines* descartados suportados pelas aplicações, em (HAMDAOUI; RAMANATHAN, 1995a) foi proposto o uso de duas constantes  $m$  e  $k$  ( $0 < m \leq k$ ), nas quais uma aplicação encontra-se em um estado sem

falhas quando ao menos  $m$  *deadlines*, em qualquer janela de  $k$  *deadlines* consecutivos, são atendidos. Contrastantemente, uma tarefa encontrase em um estado de **falha dinâmica** caso menos que  $m$  *deadlines* tenham sido atendidos em uma janela de  $k$  *deadlines* consecutivos. Por exemplo, embora ambas restrições (9,10)-*firm* e (90,100)-*firm* impliquem estatisticamente uma taxa de perdas de *deadline* de 10%, a distribuição destas perdas de *deadline* são particularmente diferentes.

No trabalho seminal de (HAMDAOUI; RAMANATHAN, 1995a) o conceito de **distância para falha** foi aplicado em uma heurística de priorização de tarefas denominado *Distance Based Priority* (DBP). Posteriormente, diversos trabalhos propuseram soluções de escalonamento adotando este mesmo modelo. Por ser tema de interesse direto desta tese, estes algoritmos de escalonamento serão discutidos no Capítulo 3.

## 2.3 ESPECIFICAÇÃO IEEE 802.15.4

As especificações do IEEE 802.15.4 definem a Camada de Acesso ao Meio (MAC) e a Camada Física (PHY) para Redes sem Fio de Área Privada com Baixa Taxa de Transmissão. O baixo consumo de energia, baixo custo e baixa taxa de transmissão são as principais características das redes IEEE 802.15.4, as quais tipicamente se enquadram nas necessidades das RSSFs. Ainda que uma rede IEEE 802.15.4 não tenha sido projetada especificamente para as RSSFs, esta tem se mostrado adequada e vem sendo amplamente adotada. Este fato vem possibilitando o surgimento e o desenvolvimento de novos produtos baseados em IEEE 802.15.4 facilitando ainda mais a disseminação desta tecnologia.

Esta seção descreve essencialmente as características do padrão IEEE 802.15.4 relacionados à camada Física e principalmente à camada MAC. Serão abordados aspectos relacionados as limitações de período e tamanhos de compartimentos (*slots*) de transmissão, impostas pelo padrão. As tecnologias de rede ZigBee e Wireless Hart que consideram o IEEE 802.15.4 como padrão da camada física e de acesso ao meio, também são descritas neste texto.

### 2.3.1 Descrição geral do IEEE 802.15.4

As LR-WPANs são redes de comunicação simples e de baixo custo que permitem a conexão sem fio para aplicações com capacidades

energéticas limitadas e que suportam baixa taxa de vazão de dados. O principal objetivo de uma LR-WPAN é sua fácil instalação, transmissão em curtas distâncias, baixo custo e baixo consumo energético.

Dois tipos de dispositivos podem participar de uma rede LR-WPAN:

- **Dispositivo de Função Completa (FDD):** é um dispositivo que pode suportar três modos de operação:
  1. **Coordenador PAN:** é o controlador principal da *Personal Area Network* (PAN), este dispositivo identifica a rede, na qual outros dispositivos devem ser associados.
  2. **Coordenador:** oferece serviços de sincronização através da transmissão de *beacons*. O coordenador deve estar associado a um coordenador PAN e ele não cria sua própria rede.
  3. **Dispositivo simples:** um dispositivo que não implementa as funcionalidades supracitadas.
- **Dispositivo de Função Reduzida (RFD):** dispositivo que opera com implementação mínima do protocolo IEEE 802.15.4. Este tipo de dispositivo é destinado para aplicações extremamente simples, que não necessitam transmitir grandes quantidades de dados e que estão associados com um único FDD.

Uma LR-WPAN deve incluir ao menos um FDD atuando como provedor global de serviços de sincronização para a rede, gerenciando FDD e RFD.

### 2.3.2 Topologias de rede

O padrão IEEE 82.15.4 define dois tipos de topologias de rede, dependendo das necessidades das aplicações: A topologia estrela e a topologia ponto a ponto. Uma terceira topologia – topologia com agrupamento em árvore – pode ser considerada como um caso especial da topologia ponto a ponto.

#### *Topologia estrela*

Na topologia estrela (Figura 11), um único nodo opera como coordenador PAN. Se um FFD é ativado, este estabelece sua própria rede, tornando-se o coordenador PAN. O Coordenador PAN escolhe



então um identificador que não esteja sendo usado por outra rede na mesma área de cobertura.

O paradigma de comunicação na topologia estrela é centralizado, ou seja, cada dispositivo (FFD ou RFD) da rede que deseja se comunicar com outro dispositivo deve primeiramente enviar seus dados para o coordenador PAN, o qual encaminha os dados para o dispositivos adequados.

Tendo em vista que tarefas realizadas pelo coordenador PAN apresentam alto gasto energético, o padrão recomenda que o coordenador PAN seja alimentado por alguma fonte de energia (ex. energia solar), enquanto os demais dispositivos comumente são alimentados por baterias.

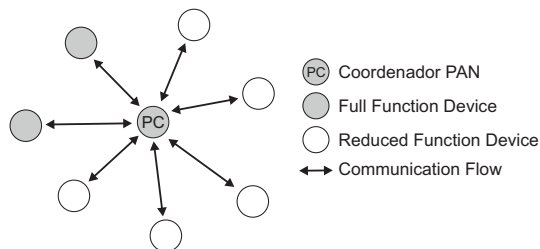


Figura 11: Topologia estrela.

Um Coordenador PAN que não esteja ligado a uma fonte externa de energia pode ter seus recursos exauridos rapidamente. Como solução potencial para esse problema, sugere-se que a eleição do coordenador PAN na rede seja realizada periodicamente, baseada na capacidade de energia dos nodos da rede (ex. LEACH (HEINZELMAN; BALAKRISHNAN, 2000)). Desta forma, o padrão IEEE 802.15.4 recomenda a topologia estrela para aplicações onde seja possível a substituição da fonte de energia do coordenador, como automação residencial, predial e industrial.

### *Topologia ponto a ponto*

A topologia ponto a ponto (*peer-to-peer*) também inclui um coordenador PAN, o qual é selecionado, por exemplo, por ser o primeiro dispositivo a se comunicar no canal. A topologia ponto a ponto é descentralizada, onde cada dispositivo pode se comunicar diretamente com outro dispositivo na sua área de cobertura (Figura 12).

Esta topologia possibilita flexibilidade, mas introduz complexi-

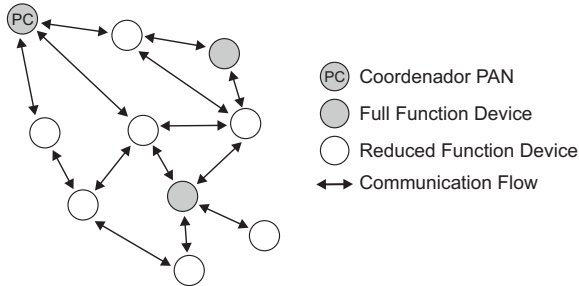


Figura 12: Topologia ponto a ponto.

dade adicional na rede, em virtude dos nodos poderem se comunicar com múltiplos saltos entre os dispositivos. No entanto, estas funções devem ser definidas na camada de rede e não são consideradas na especificação do padrão IEEE 802.15.4.

### *Agrupamento em árvore*

A topologia com agrupamento em árvore é um caso especial da topologia ponto a ponto, no qual alguns dispositivos são FFDs (Figura 13).

- Um (e apenas um) coordenador é nomeado coordenador PAN, o qual identifica toda a rede.
- Qualquer FFD pode atuar como coordenador e prover serviços de sincronização para outros dispositivos ou outros coordenadores.
- Um RFD se conecta ao agrupamento como um nodo livre ao final da árvore e se associa com apenas um FFD.

Atualmente, o padrão IEEE 802.15.4 (802.15.4, 2006) não define especificamente como construir um agrupamento em árvore. Ele apenas indica que é possível e que deve ser iniciado pelas camadas superiores. A formação de um agrupamento pode ser realizado da seguinte maneira:

#### 1. O coordenador PAN:

- forma o primeiro agrupamento, nomeando a si mesmo como líder do agrupamento (*Cluster Head* (CH)) com o identificador do agrupamento (CID) igual a zero;

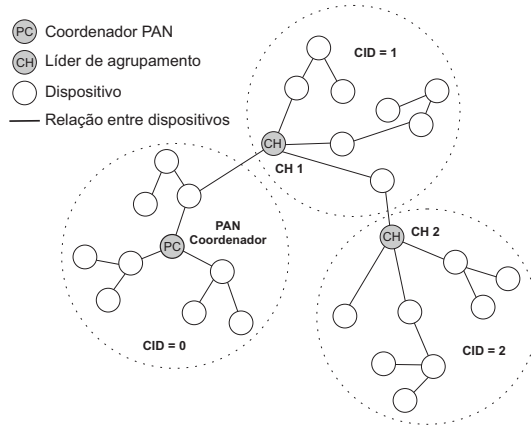


Figura 13: Agrupamento em árvore.

- escolhe um identificador PAN ainda não utilizado;
  - difunde *beacons* para seus dispositivos vizinhos (também conhecido como vizinhança).
2. Um dispositivo que recebe um quadro de *beacon* deve solicitar sua junção ao agrupamento ao seu CH:
- se um PAN aceita a requisição para se juntar a rede, ele adiciona este dispositivo candidato como um dispositivo filho em sua lista de vizinhos. Em seguida, o novo dispositivo adiciona o CH como seu nodo pai em sua lista de vizinhos e passa a transmitir *beacons* periodicamente. Outros dispositivos ouvindo estes *beacons* devem se juntar à rede através deste dispositivo.
  - Se por alguma razão o dispositivo candidato não puder se juntar à rede através do CH, este irá procurar por outro dispositivo vizinho.

Em uma rede de grande escala, é possível formar uma rede de múltiplos agrupamentos vizinhos. Neste caso, o coordenador PAN pode atualizar um dispositivo para tornar-se um novo CH de um novo agrupamento próximo ao dispositivo em questão. Outros dispositivos gradualmente se conectam aos agrupamentos formando uma rede de agrupamento, como ilustrado na Figura 13. A especificação permite a implementação de agrupamentos em árvore com um único agrupamento ou uma rede com vários agrupamentos.

### 2.3.3 Camada física IEEE 802.15.4

A camada física é responsável pela transmissão e recepção de dados utilizando um determinado canal de rádio de acordo com alguma modulação ou técnica de difusão. O padrão IEEE 802.15.4 oferece três frequências de operação ISM (Industrial, Scientific and Medical): 2.4 GHz, 915 MHz e 868 MHz. Existe apenas um único canal de entre 868 e 868.6 MHz, 10 canais entre 902 e 928 MHz, e 16 canais entre 2.4 e 2.4835 GHz.

As taxas de dados nominais máximas alcançadas são de 250 kbps em 2.4 GHz, 40 kbps em 915 MHz e 20 kbps em 868 MHz (Tabela 1). Baixas frequências são adequadas para transmissões de longas distâncias devido à menor perda por atenuação do sinal. Baixas taxas de transmissão proporcionam melhor sensibilidade à área de cobertura. Altas taxas de transmissão significam alta vazão e baixa latência (A IEEE possui um padrão específico para PANs com altas taxas de transmissão – o padrão IEEE 802.15.3).

Frequência	Modulação	Taxa de dados máxima
868 MHz	BPSK	20 kbps
915 MHz	BPSK	40 kbps
2400 MHz	O-QPSK	250 kbps

Tabela 1: Frequências, modulações e taxas de dados IEEE 802.15.4.

A camada física é definida pelo padrão IEEE 802.15.4 (802.15.4, 2006) como responsável pelas seguintes tarefas:

1. Ativação e desativação do transceptor (*transceiver*) de rádio: o transceptor pode operar em um dos três modos: transmitindo, recebendo ou adormecido. Dependendo da solicitação da camada MAC o rádio pode ser ligado ou desligado.
2. Detecção de energia no canal (*Energy Detection (ED)*): este serviço é utilizado pela camada de rede como parte do mecanismo de seleção de canal ou pelo *Clear Channel Assessment (CCA)* para determinar se o canal está ocioso ou ocupado.
3. *Link Quality Indication (LQI)*: o LQI caracteriza a qualidade de um sinal recebido em um enlace. Pode ser utilizado pelas camadas superiores (rede e aplicação) para implementação de suas políticas.

4. *Clear Channel Assessment*: esta tarefa essencial para implementação do *Carrier Sense Multiple Access* (CSMA) é responsável por reportar o estado do meio: ocioso ou ocupado. O CCA pode ser realizado em três modos de operação:
- Modo de detecção de energia: o CCA reporta que o canal está ocupado quando detecta energia acima de um limiar de energia (ED) no canal.
  - Modo de detecção de portadora: o CCA reporta que o canal está ocupado apenas se ele detecta um sinal com as mesmas características de modulação e difusão do IEEE 802.15.4, e que possua um valor de energia (ED) acima ou abaixo do limiar.
  - Modo de detecção de portadora com modo de detecção: é a combinação das duas técnicas citadas anteriormente.
5. Seleção de frequência do canal: o padrão IEEE 802.15.4 define até 27 canais diferentes. Assim, a camada física deve estar apta para configurar seus transceptores em um canal específico, escolhido pela camada superior.

Comercialmente, alguns sensores disponíveis no mercado implementam o padrão IEEE 802.15.4. Por exemplo, o MICAz, cujo fabricante é a Crossbow Tech<sup>5</sup>, implementa parcialmente o padrão, operando apenas em 2.4 GHz ISM a 250 kbps. Este tipo de sensor utiliza 5 MHz de espaçamento de canal, conforme sugerido pela especificação.

### 2.3.4 Controle de Acesso ao Meio do IEEE 802.15.4

A Subcamada MAC do padrão IEEE 802.15.4 (802.15.4, 2006) provê uma interface entre a camada Física e as camadas superiores das LR-WPANs. Várias características são compatíveis com as do protocolo IEEE 802.11 (802.11, 2007), como o uso do protocolo de acesso ao meio *Carrier Sense Multiple Access with Collision Avoidance* (CSMA/CA). No entanto o protocolo IEEE 802.15.4 é adaptado para as necessidades das LR-WPANs, eliminando as mensagens de *Request to Send* (RTS) / *Clear to Send* (CTS), usadas no IEEE 802.11 para redução da probabilidade de colisões, mas que consomem muita energia a cada envio de mensagem.

---

<sup>5</sup><http://www.xbow.com>

O protocolo MAC suporta dois modos de operação que devem ser selecionados pelo coordenador e são apresentados graficamente na Figura 14:

1. **Modo com *beacon***: os *beacons* são periodicamente gerados pelo coordenador com a finalidade de sincronizar dispositivos e identificar a PAN. Um quadro de *beacon* é a primeira parte de um superquadro, o qual embute todos os quadros trocados entre os nodos e o coordenador PAN. Trocas de dados entre nodos também são possíveis durante a duração do superquadro.
2. **Modo sem *beacon***: neste modo de operação os nodos podem simplesmente enviar seus dados usando CSMA/CA sem compartimentos. Neste modo não se utiliza a estrutura do superquadro.

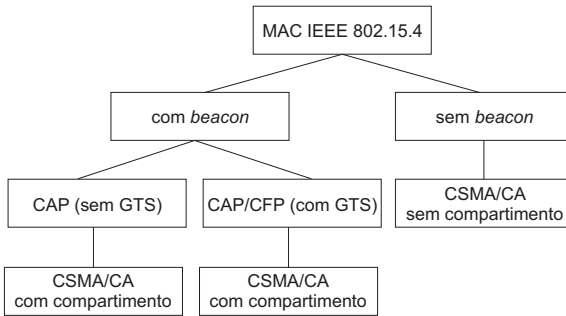


Figura 14: Modos de operação IEEE 802.15.4.

Esses dois modos de operação são melhor descritos a seguir.

#### 2.3.4.1 Modos de operação do IEEE 802.15.4

##### *Modo de operação com beacon*

Quando o coordenador habilita o modo com *beacon*, ele força a utilização da estrutura do superquadro para gerenciar a comunicação entre os dispositivos (que são associados com a PAN). O formato do superquadro é definido pelo coordenador PAN e transmitido para os outros dispositivos contido dentro de cada quadro de *beacon*, o qual é enviado periodicamente pelo coordenador PAN. O superquadro é dividido em 16 compartimentos de mesmo tamanho, seguido por um

período inativo previamente definido. Conforme ilustrado na Figura 15, o superquadro é contido em um intervalo de *beacons* (*Beacon Interval* - BI) e limitado por dois quadros de *beacon*. Internamente este pode conter o período de acesso com contenção (CAP) e o período de acesso sem contenção (CFP). Se o mecanismo de comunicação utiliza exclusivamente o CAP, um dispositivo que deseja se comunicar deve competir com outros dispositivos usando o CSMA/CA com compartimentos. Todas as transmissões devem ser finalizadas antes do fim do CAP, ou antes do início do período inativo (se houver).

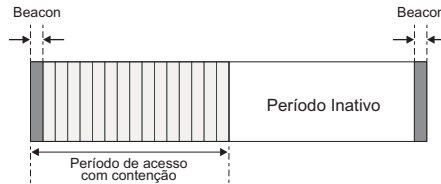


Figura 15: Estrutura do superquadro sem GTSs.

Se existe a necessidade de provimento de alguma garantia com respeito à ausência de colisões, então o *Contention Free Period* (CFP) é utilizado. O CFP consiste na alocação de compartimentos garantidos (GTS) pelo coordenador PAN para aplicações que necessitam de largura de banda definidas. O CFP é uma parte do superquadro iniciada imediatamente após o término do período de acesso com contenção. O coordenador PAN pode alocar até sete compartimentos garantidos e cada GTS pode ocupar mais do que um compartimento. De acordo com a especificação IEEE 802.15.4, o GTS deve ser utilizado apenas para comunicação ponto a ponto envolvendo o coordenador PAN e um dispositivo da rede.

Em ambas as configurações (com e sem CFP), o superquadro pode possuir um período inativo, durante o qual todos os dispositivos, inclusive o coordenador PAN, não interagem com a PAN, e devem entrar em um modo de economia de energia (*sleep mode*). Esta troca entre os modos de operação da rede é bastante apropriado para RSSFs, onde a economia de energia e prolongamento do tempo de vida da rede é de fundamental relevância. Esse modo de economia de energia é compatível com o modelo de funcionamento de aplicações de manufatura que trabalham baseadas em ciclo de trabalho (*duty cycles*). Um ciclo de trabalho é utilizado para descrever a fração de tempo em que um sistema está em um estado ativo.

### Modo de operação sem beacon

Quando o coordenador PAN seleciona o modo de operação sem *beacon*, não existe a estrutura do superquadro nem o envio de quadros de *beacon*. O acesso ao meio é realizado através do tradicional protocolo de acesso ao meio CSMA/CA sem compartimentos. Todas as mensagens a serem transmitidas, com exceção das mensagens de reconhecimento (ACK) e quadros que seguem imediatamente um ACK, devem ser transmitidos segundo este mecanismo, o qual não oferece nenhuma garantia temporal às aplicações. O protocolo CSMA/CA será visto adiante na Sub-seção 2.3.4.3.

#### 2.3.4.2 Estrutura do superquadro

A estrutura do superquadro é descrita por dois parâmetros. O primeiro, *Beacon Order* (BO) determina o intervalo no qual o coordenador transmite quadros de *beacon*. O Intervalo de *Beacon* (BI) é definido pela Equação 2.7.

$$BI = aBaseSuperframeDuration \cdot 2^{BO}, \quad (2.7)$$

para  $0 \leq BO \leq 14$

O segundo atributo, *Superframe Order* (SO) define o comprimento da porção ativa do superquadro. A porção ativa do superquadro, *Superframe Duration* (SD) é definida a seguir pela Equação 2.8.

$$SD = aBaseSuperframeDuration \cdot 2^{SO}, \quad (2.8)$$

para  $0 \leq SO \leq BO \leq 14$

Se  $SO = BO$  então  $SD = BI$ , portanto o superquadro é sempre ativo. De acordo com o padrão, se  $SO = 15$ , o superquadro não terá período ativo após o *beacon*. Além disso, se o  $BO = 15$ , então o superquadro não existirá, e a rede operará em modo sem *beacon*, neste caso o valor de  $SO$  é ignorado.

Nas Equações 2.7 e 2.8, o parâmetro *aBaseSuperframeDuration* denota a duração mínima do superquadro, correspondendo a  $SO = 0$ . O padrão IEEE 802.15.4 (802.15.4, 2006) fixa esta duração em 960 símbolos (um símbolo corresponde a 4 bits). Este valor corresponde



a 15.36 ms, assumindo uma taxa de dados de 250 kbps, em uma frequência de 2.4 GHz, a qual será considerada no resto deste trabalho sempre que forem fornecidos para mapeamentos de símbolos para valores de tempo ou bytes.

A porção ativa de cada superquadro é composta de três partes: *beacon*, CAP e CFP.

- *Beacon*: é transmitido sem uso do período de contenção (sem CSMA/CA) no início do compartimento 0 (zero) do superquadro. O quadro de *beacon* contém informações sobre o endereçamento, especificações do superquadro, os campos de GTS, endereços pendentes, etc.
- *Contention Access Period* (CAP): inicia-se imediatamente após o quadro de *beacon* e termina antes do CFP (se existir). De outra forma, o CAP é finalizado ao término do período ativo do superquadro. O comprimento mínimo recomendado para o CAP é  $aMinCAPlength = 440$  símbolos. Este tamanho mínimo garante que os comandos MAC ainda possam ser trocados entre os dispositivos quando o GTS estiver ativado. Todas as transmissões realizadas durante o CAP são realizadas usando o protocolo de acesso ao meio CSMA/CA. No entanto, ACKs e qualquer dado transmitido imediatamente após um ACK são transmitidos imediatamente, sem contenção. Um dispositivo que não consegue completar sua transmissão dentro de um *Interframe Space* (IFS) antes do término do CAP, deve postergar sua transmissão para o próximo superquadro.
- *Contention Free Period* (CFP): inicia-se imediatamente após CAP, e deve ser completado antes do início do próximo quadro de *beacon*. Todos os GTSs que podem ser alocados pelo coordenador PAN estão localizados no CFP e devem ocupar compartimentos contíguos. O tamanho do CFP pode aumentar ou diminuir de acordo com o tamanho total dos GTSs. As transmissões dentro do CFP são efetuadas sem contenção. Um quadro só é transmitido se sua transmissão puder ser concluída um IFS antes do final do GTS correspondente.

A Figura 16 ilustra a estrutura do superquadro, onde o Intervalo de *Beacon* (BI) é duas vezes maior do que a porção ativa (SD) do superquadro. Existem também, dois GTSs alocados durante o CFP.

A Subcamada MAC precisa de tempo para processar dados recebidos da camada física. Portanto, dois quadros sucessivos transmiti-

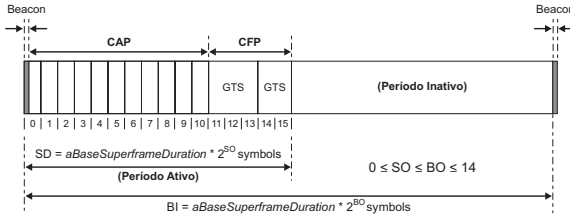


Figura 16: Exemplo de estrutura do superquadro com CAP e CFP.

dos por um dispositivo devem ser separados por ao menos um período de IFS. Se a primeira transmissão necessita de uma mensagem de reconhecimento ACK, a separação entre o quadro de ACK e a segunda transmissão deve ser de pelo menos um período de IFS.

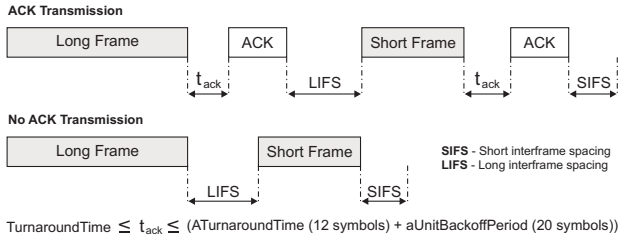


Figura 17: Espaçamento entre quadros no IEEE 802.15.4.

A duração do período de IFS é dependente do tamanho do quadro transmitido. Quadros com tamanho de até  $aMaxSIFSFrameSize$  são seguidos por um período *Short Interframe Space* (SIFS), com duração de no mínimo  $macMinSIFSFramePeriod$ . Quadros com tamanho maior que  $aMaxSIFSFrameSize$  são seguidos por um período de *Long Interframe Space* (LIFS), de no mínimo  $macMinLIFSFramePeriod$ . A Figura 17 ilustra a relação entre tamanho de quadros e períodos de IFS, considerando transmissões com e sem mensagens de reconhecimento.

A Tabela 2 mostra os valores de IFS definidos pelo padrão IEEE 802.15.4 (802.15.4, 2006). Estes valores consideram que são exigidas mensagens de reconhecimento depois das transmissões dos quadros.

aMaxSIFSFrameSize	36 symbols	18 bytes
macMinSIFSPeriod	12 symbols	6 bytes
macMinLIFSPeriod	40 symbols	20 bytes
Ack Time $t_{ack}$	32 symbols	16 bytes

Tabela 2: Valores de IFS.

### 2.3.4.3 Protocolo de acesso ao meio CSMA/CA

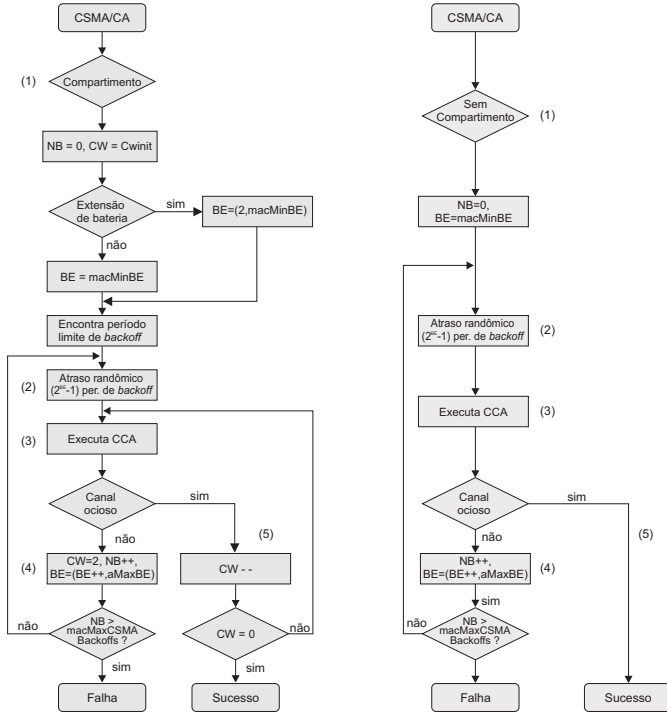
O problema central nos enlaces de múltiplo acesso é determinar quem deve transmitir e quando. Como vários nodos podem transmitir quadros ao mesmo tempo, estes poderão colidir e serão perdidos. Os protocolos de acesso múltiplo ao meio permitem coordenar o acesso ao meio, evitando ou detectando estas colisões. Uma técnica tradicionalmente utilizada em redes de computadores são os protocolos de acesso aleatório. O protocolo utilizado em redes locais Ethernet são baseados na técnica de escuta de portadora antes de enviar uma informação *Carrier Sense Multiple Access* (CSMA).

O protocolo CSMA foi desenvolvido para funcionar com computadores conectados em barramento. Desta forma, antes de enviar um quadro, o nó escuta o canal. Caso o canal esteja livre, o nodo transmite o quadro imediatamente; caso o canal ocupado, volta a escutá-lo depois de decorrido um tempo aleatório para tentar novamente transmitir. Se mais de um nodo começa a transmitir ao mesmo tempo, a transmissão é abortada. Este procedimento é chamado de detecção de colisões (*Collision Detection*), onde os nodos continuam ouvindo o canal enquanto transmitem: caso detectem uma sobreposição de transmissões (colisões), param imediatamente a transmissão. Estes dois procedimentos são realizados pelo protocolo de acesso ao meio utilizado em redes locais Ethernet.

Em redes sem fios, os dispositivos desligam seus tranceptores durante suas transmissões, impossibilitando que os transmissores escutem o meio de comunicação enquanto realizam suas transmissões; e consequentemente detectem a ocorrência de colisões. Sendo assim, as redes de sensores sem fio tentam evitar a ocorrência de colisões (*collision avoidance*) através de períodos de *backoff* aleatórios antes de verificar a disponibilidade do canal, ao invés de permanecer escutando o meio (e consumindo energia) até que o mesmo esteja livre.

O padrão IEEE 802.15.4 define duas versões para o CSMA/CA: com e sem compartimentos. O CSMA/CA com compartimentos é

usado no modo com *beacon*, e o CSMA/CA sem compartimentos, utilizado no modo sem *beacon*. Em ambos os casos, o protocolo CSMA/CA é baseado em períodos de *backoff*, onde o período é igual  $aUnitBackoffPeriod = 20$  símbolos.



(a) CSMA/CA com compartimentos. (b) CSMA/CA sem compartimentos.

Figura 18: Algoritmos CSMA/CA com e sem compartimentos.

Apesar do protocolo CSMA/CA com compartimentos se assemelhar ao protocolo Aloha (ABRAMSON, 1970), pois ambos são baseados no conceito da divisão do tempo em compartimentos de tamanhos iguais, a operação do CSMA/CA com compartimentos é totalmente diferente do Aloha. Em contraste com o Aloha, a versão do CSMA/CA com compartimentos não precisa aguardar o início de um compartimento para iniciar suas transmissões e nem concluir suas transmissões antes do término de um compartimento. Esta abordagem permite melhor aproveitamento do canal de comunicação, tornando possível mais

do que 15 (número máximo de compartimentos) transmissões dentro de um mesmo superquadro.

O protocolo CSMA/CA utiliza três variáveis para escalonar o acesso ao meio (Figura 18).

1. **Number of Backoff (NB)**: corresponde ao número de vezes que o algoritmo CSMA/CA foi solicitado para entrar em *backoff* enquanto tenta acessar o canal. Valor inicialmente em zero antes de cada nova transmissão. Caso o número de tentativas de acesso ao canal seja maior que  $macMaxCSMABackoffs$  ( $NB > macMaxCSMABackoffs$ ) uma mensagem de falha no canal é enviada para as camadas superiores na pilha do protocolo de comunicação.
2. **Contention Window (CW)**: tamanho da janela de contenção, define o número de vezes necessárias para verificar a disponibilidade de acesso ao canal, antes do início de uma transmissão. O parâmetro CW só é utilizado na versão do CSMA/CA com compartimento, seu valor inicial é dois ( $CW = 2$ ), e a cada tentativa de transmissão com canal ocupado, o valor desta variável é novamente reiniciado para 2.
3. **Backoff Exponent (BE)**: é o expoente de *backoff*, valor relacionado com quantos períodos de *backoff* um dispositivo deve aguardar para tentar avaliar a atividade do canal. O valor  $BE = 0$  indica que não haverá backoff na primeira verificação de condição do canal (ocupado ou ocioso). Este valor relacionado com a tentativa de evitar a colisão (CA do protocolo, *collision avoidance*) obedece uma função de período aleatório onde  $[0, 2^{BE} - 1]$ .

A Figura 18 apresenta o fluxograma para as duas versões do CSMA/CA. A versão com compartimentos pode ser resumida em cinco etapas (802.15.4, 2006), ilustrado graficamente na Figura 18(a).

1. Inicialização das variáveis NB, CW e BE: NB é inicializado com 0 e a Janela de Contenção CW é inicializada com 2. A seguir o protocolo MAC verifica se o atributo *macBattLifExt* está ativo. Neste caso, BE é iniciado com o valor mínimo de 2 ou  $macMinBE$ , de outra forma,  $BE = macMinBE$  (iniciado em 3, por padrão). Se o valor de  $macMinBE = 0$ , o mecanismo que evita colisões é desativado na primeira interação. Depois da inicialização o algoritmo busca o limite para o próximo período de *backoff*.

2. Espera aleatória para evitar colisões: o algoritmo busca evitar colisões aguardando um tempo de espera aleatório, durante um dado intervalo aleatório gerado na faixa de  $[0, 2^{BE} - 1]$  períodos de *backoff*.
3. CCA: é utilizado para verificar se o canal está ocupado, executado logo após o tempo de *backoff*.
4. Canal ocupado: se o canal é verificado ocupado, CW é reiniciado com valor 2 e os valores de NB e CW são incrementados. No entanto, BE não pode exceder o valor de *aMaxBE*. Se o número de tentativas exceder o valor de *macMaxCSMABackoffs*, o algoritmo termina como um erro de acesso ao canal. Caso contrário, o algoritmo volta ao passo 2.
5. Canal ocioso: se o canal é verificado como ocioso, o valor de CW é decrementado. Se o valor de CW expira ( $CW = 0$ ), o protocolo MAC deve iniciar sua transmissão com sucesso, de outra maneira, o algoritmo volta ao passo 3. É importante ressaltar que a transmissão do quadro é iniciada apenas se o número restante de *backoffs* no superquadro for suficiente para manipular o quadro e seu ACK subsequente. Caso contrário, a transmissão do quadro é postergada para o próximo superquadro.

A versão do CSMA/CA sem compartimentos é similar a versão com compartimentos, com a particularidade de a variável *CW* não ser utilizada (ver Figura 18(b)).

#### 2.3.4.4 Alocação e gerência de GTS

O Compartimento de Tempo Garantido (GTS) é a porção do superquadro que pode ser alocada para acesso exclusivo de alguns dispositivos da PAN. O GTS permite ao dispositivo acesso ao meio sem períodos de contenção dentro do CFP. Um GTS só pode ser alocado por um coordenador PAN e deve ser usado apenas para comunicação ponto a ponto entre o coordenador PAN e um dispositivo. Um único GTS pode possuir mais de um compartimento contíguo do superquadro. O PAN Coordinator pode alocar até sete GTSs simultaneamente, considerando que exista capacidade suficiente no superquadro. Um GTS precisa ser reservado antes de seu uso e cada GTS deve ser desalocado quando não é mais utilizado, podendo ser desalocado a qualquer momento pelo PAN Coordinator ou pelo dispositivo que originalmente o

reservou para uso. Todas as transmissões dentro do CFP devem ser completadas antes do término se seus períodos de GTS.

A Figura 19 mostra o campo de características de GTS, enviado dentro de um quadro de alocação de GTS de um nodo para o Coordenador PAN (802.15.4, 2006). O dispositivo expressa o número de GTS a serem alocados no campo **comprimento**. A especificação de direção indica a direção do fluxo de dados (em cada GTS), dados são transmitidos do Coordenador PAN para o nodo solicitante, ou dados são transmitidos do dispositivo solicitante para o Coordenador PAN. Cada dispositivo seleciona um GTS para transmissão ou recepção. O campo **características** refere-se à característica do GTS sendo: alocação (valor = 1) ou desalocação (valor = 0).

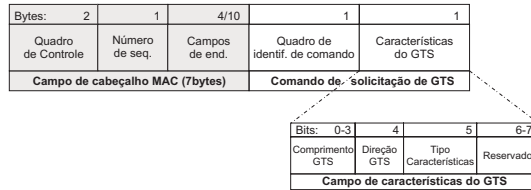


Figura 19: Estrutura do quadro de solicitação de GTS.

Após o recebimento da solicitação, o coordenador PAN envia uma mensagem de reconhecimento – para confirmar o recebimento da solicitação de GTS – e verifica se há compartimentos suficientes disponíveis no superquadro para atender esta solicitação. Se o número de compartimentos de tempo no superquadro for menor que o número de GTSs solicitados, a solicitação é rejeitada. O PAN Coordinator deve garantir que o comprimento CAP não seja menor que  $aMinCAPLength = 440$  símbolos (equivalente a 220 bytes com meio físico usando a faixa de frequência de 2.4 GHz).

Usualmente, a alocação de GTS é realizada segundo ordem FIFO (*First-in, First-out*), e todos os GTS devem ser alocados contiguamente no final do superquadro. O coordenador PAN toma essa decisão em  $aGTSDescPersistenceTime$  (onde  $aGTSDescPersistenceTime = 4$  superquadros). Sendo assim, o dispositivo que solicitou GTS permanece monitorando o recebimento de *beacons* até  $aGTSDescPersistenceTime$  superquadros. Caso nenhum *beacon* com descritores relacionados a sua requisição chegue, a alocação do GTS é considerada sem sucesso.

Caso a solicitação de alocação de GTS seja aceita pelo coordenador PAN, este gera um descritor de GTS com a especificação solicitada. Esta informação está localizada no campo de descritores de

GTS, inclusa em cada quadro de *beacon*, e enviada pelo coordenador PAN. Dispositivos que solicitaram alocação de GTSs devem monitorar os quadros de *beacon* e verificar qual compartimento de tempo foi alocado no superquadro atual.

A desalocação de um GTS pode resultar em fragmentação do superquadro. A figura 20, apresenta uma possível fragmentação, após a desalocação do GTS2 (802.15.4, 2006).

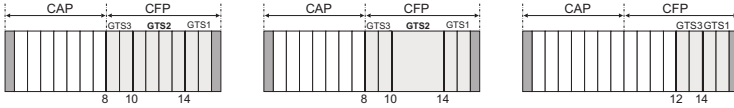


Figura 20: Fragmentação e realocação de GTSs.

O coordenador PAN deve garantir, que lacunas existentes no CFP, sejam removidas para maximizar a utilização do superquadro (realinhamento), aumentando o comprimento do CAP.

### 2.3.5 Estudo dos limites físicos do padrão IEEE 802.15.4

Apesar de buscar desenvolver um padrão genérico e flexível, considerando uma ampla gama de aplicações, o padrão IEEE 802.15.4 apresenta uma série de restrições para suas aplicações. Por exemplo, existe pouca flexibilidade para estabelecimento de valores de períodos e tamanhos de compartimentos.

Os dados apresentados a seguir consideram uma PAN configurada a 2.4 GHz. A Figura 21 ilustra o caso para  $SO = 1$  e  $BO = 2$ . Neste caso, existe um superquadro limitado por um intervalo de *beacon* (BI) de 61.44 ms, formado por um período ativo SD e um período inativo com 30.72 ms. Cada um dos dezesseis compartimentos que formam o período ativo possuem 60 bytes (1.92 ms). Como apenas um dispositivo está alocando um compartimento durante o CFP, este dispositivo tem acesso garantido apenas no 15º compartimento. Em outras palavras: 28.8 ms depois do início do superquadro. A carga útil dentro do GTS alocado, considerando um quadro com mais que 18 bytes que necessita de uma mensagem de reconhecimento, é de apenas 24 bytes dos 60 bytes que formam o compartimento.

A periodicidade das tarefas em aplicações IEEE 802.15.4 pode variar entre valores que vão de 15.36 ms até 4.2 minutos, com tempo de compartimento de 960  $\mu$ s a 15.7 segundos e o número máximo de



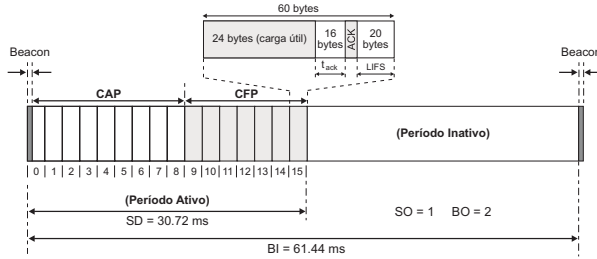


Figura 21: Utilização real na alocação de compartimentos de GTS.

bytes que podem ser transmitidos nestes compartimentos de tempo varia de 30 bytes até 491 kbytes. Esta relação entre tamanho de compartimentos e periodicidade das tarefas pode ser um fator negativo para algumas aplicações, desde que a necessidade de um compartimento com capacidade de transmissão maior implica na imposição de um período mais longo para as aplicações. A Tabela 3 relaciona os valores de períodos possíveis juntamente com os comprimentos dos compartimentos de tempo, de acordo com o padrão IEEE 802.15.4 (802.15.4, 2006). Adicionalmente, apresenta-se o tamanho dos compartimentos durante o período ativo.

SO=BO	Período (ms)	Compartimento (ms)	Tamanho (bytes)
0	15.36	0.96	30
1	30,72	1.92	60
2	61.44	3.84	120
3	122.88	7.68	240
4	245.76	15.36	480
5	491.52	30.72	960
6	983.04	61.44	1920
7	1966.08	122.88	3840
8	3932.16	245.76	7680
9	7864.32	491.52	15360
10	15728.64	983.04	30720
11	31457.28	1966.08	61440
12	62914.56	3932.16	122880
13	125829.12	7864.32	245760
14	251658.24	15728.64	491520

Tabela 3: Períodos do superquadro no IEEE 802.15.4 (sem período inativo).

A Tabela 4 apresenta os valores de carga útil (*payload*) disponíveis para as aplicações em cada compartimento durante o SD, dentro de um superquadro. Considera-se o tempo de IFS e ACK dentro do mesmo compartimento de tempo, conforme ilustrado na Figura 21.

SO	Com reconhecimento		Sem reconhecimento	
	LIFS	SIFS	LIFS	SIFS
0	0	8	10	24
1	24	38	40	54
2	84	98	100	114
3	204	218	220	234
4	444	458	460	474
5	924	938	940	954
6	1884	1898	1900	1914
7	3804	3818	3820	3834
8	7644	7658	7660	7674
9	15324	15338	15340	15354
10	30684	30700	30698	30714
11	61404	61418	61420	61434
12	122844	122858	122860	122874
13	245724	245738	245740	245754
14	491484	491498	491500	491514

Tabela 4: Carga útil no IEEE 802.15.4 (bytes).

Para o estudo do escalonamento do CFP feito nesta tese e mostrado no Capítulo 6 foi necessário compreender os limites existentes no IEEE 802.15.4 no modo sem contenção. Através do estudo da especificação e de experimentos práticos com nodos MicaZ, constatou-se diversas restrições existentes neste protocolo. Algumas dessas restrições justificam fortemente as abordagens propostas nesta tese.

### 2.3.6 ZigBee

As especificações do IEEE 802.15.4 (802.15.4, 2006) estão amplamente associadas com as especificações ZigBee. A Aliança ZigBee (ZIGBEE-ALLIANCE, 2007) é uma organização com mais de 150 membros, trabalhando em conjunto com o Instituto de Engenheiros Eletricistas e Eletrônicos (IEEE) com o intuito de especificar protocolos completos para redes com restrições (baixo consumo energético, baixa taxa de transmissão de dados, baixa complexidade e baixo custo).

O padrão ZigBee especifica as camadas de protocolo acima do IEEE 802.15.4 (camadas de rede e de aplicação). A Figura 22 apresenta a arquitetura da pilha de protocolos ZigBee/IEEE 802.15.4.

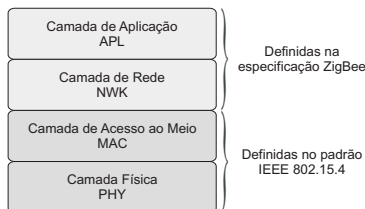


Figura 22: Pilha de protocolos ZigBee/IEEE 802.15.4.

O custo total no desenvolvimento de um sistema é fator fundamental para aplicações industriais e residenciais, o desenvolvimento de chips otimizados e altamente integrados em um único circuito são as soluções adotadas pela IEEE 802.15.4 para o desenvolvimento de seus transceptores de rádios. Devido às soluções sem fios de curto alcance desenvolvidas pela IEEE, acima das camadas Física de MAC, uma camada importante neste padrão é a camada de roteamento. A Aliança ZigBee está em processo de definição as características para esta camada para as topologias estrela, ponto a ponto e agrupamento em árvore. Desta forma o desempenho destas redes complementar o padrão IEEE 802.15.4 enquanto mantém os requisitos de baixa complexidade e baixa potência.

Redes ZigBee são arquiteturas auto-organizáveis e autogerenciáveis, permitindo que mensagens sejam repassadas de um nodo para outro via múltiplos caminhos. Esta característica estende o alcance da rede e aumenta sua confiabilidade. As redes ZigBee oferecem também ferramentas de segurança para garantir redes seguras e confiáveis contendo listas de controle de acesso e criptografia de 128 bits.

### 2.3.7 WirelessHart

O WirelessHart (HART, ) é uma rede de comunicação para redes sem fio projetada para atender as necessidades de aplicações de automação e controle de processos. Estas aplicações incluem monitoramento, diagnósticos, alarmes, detecção de eventos, controle, etc. Para dar suporte a essas aplicações várias formas de comunicação são necessárias.

O WirelessHart é uma especialização da rede *Highway Addressable Remote Transducer* (HART) que há cerca de 20 anos vem sendo utilizada por indústrias de automação de processos. Uma característica do WirelessHart é sua compatibilidade com aplicações que utilizam o HART. Desta forma, aplicações HART como sistemas de controle, PLCs, ferramentas de manutenção e aplicações de gerenciamento, podem utilizar o WirelessHart sem a necessidade de atualizações de software.

Tradicionalmente, o protocolo HART utiliza um mecanismo de passagem de permissão (*token-passing*) para comunicação dos dados. Com a inclusão do WirelessHart, duas camadas adicionais foram adicionadas: Camada física IEEE 802.15.4 e camada de enlace TDMA. Atualmente uma especificação completa da camada de rede também está disponível.

### 2.3.7.1 Visão geral

WirelessHart é uma tecnologia sem fios que opera em 2.4 GHz ISM, com rádio compatível com o padrão IEEE 802.15.4. A arquitetura do WirelessHart é projetada para ser uma rede de sensores de fácil utilização, confiável e interoperável. O WirelessHart especifica requisitos mandatórios que equipamento compatíveis devem suportar para atender requisitos de interoperabilidade de tal forma que dispositivos sem fios de diferentes fabricantes possam trocar informações sem comprometer a operação do sistema.

A comunicação entre os dispositivos WirelessHart utiliza *Time Division Multiple Access* (TDMA) para arbitrar e coordenar a comunicação entre os dispositivos da rede. A camada de enlace define compartimentos de tempo e frequências específicas para serem utilizadas na comunicação entre dispositivos. Estes compartimentos são organizados em superquadros, que são periodicamente repetidos com o objetivo de oferecer suporte para tráfego cíclico e acíclico de dados.

Um compartimento pode ser dedicado exclusivamente para um dispositivo (ex. garantindo que os dados serão entregues com a mínima latência possível) ou compartilhado entre os nodos da rede (como no CAP em redes IEEE 802.15.4), buscando maior utilização da largura de banda disponível. Os modos de alocação disponíveis oferecidos pelo WirelessHart são semelhantes ao modo de operação com *beacon* em redes IEEE 802.15.4. O primeiro modo de alocação de compartimentos WirelessHart é semelhante à alocação de GTS durante o CFP em redes

IEEE 802.15.4 e o segundo modo de alocação pode ser visto como o CAP.

A *Instrumentation Systems and Automation Society (ISA)*<sup>6</sup> lançou uma versão preliminar (*draft*) da especificação ISA 100.11a, onde são especificadas as camadas física e de enlace sobre o protocolo IEEE 802.15.4. Portanto, as aplicações para controle de processos industriais que seguem esta especificação, como é o caso do WirelessHART, serão especificados em conformidade pela norma ISA 100.11a.

### 2.3.7.2 Arquitetura de rede

Uma rede WirelessHart é formada por três elementos principais (Figura 23):

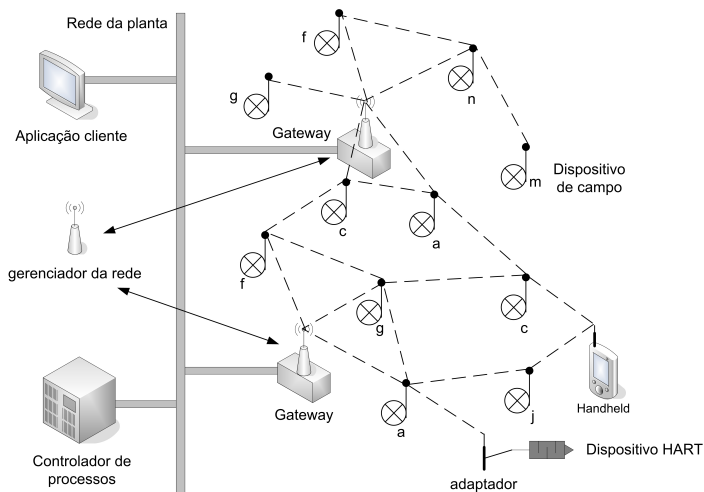


Figura 23: Elementos de uma rede WirelessHart.

- **Dispositivos de campo (WFD):** são conectados ao processo ou a algum dispositivo da planta.
- **Gateways:** dispositivos que habilitam a comunicação entre aplicações cliente e os dispositivos de campo em uma rede WirelessHart. Um *Gateway* suporta um ou mais pontos de acesso. Um *Gate-*

<sup>6</sup><http://www.isa.org/>

*way* e seus pontos de acesso devem ser incluídos em todas as redes WirelessHart.

- **Gerenciador da rede:** dispositivo responsável pela configuração da rede, escalona a comunicação entre os dispositivos WirelessHart (ex. configuração do superquadro), gerencia as tabelas de roteamento, monitora e reporta o estado da rede. Podem existir gerenciadores redundantes na rede, porém deve haver apenas um gerenciador ativo por rede em um dado instante.

Adicionalmente, o WirelessHart suporta dois tipos de dispositivos:

- **Adaptadores WirelessHart:** adaptadores que permitem dispositivos de campo já existentes integrar-se na rede WirelessHart.
- **Portáteis WirelessHart:** portáteis ou *handhelds* suportam acesso direto a um dispositivo de campo, utilizando antenas WirelessHart embutidas no equipamento.

## 2.4 CONCLUSÕES DO CAPÍTULO

STR vêm sendo empregado atualmente em uma vasta gama de aplicações. Em virtude da ampla disseminação desses sistemas, surge a necessidade de encontrar abordagens que permitam sua execução em ambientes distribuídos e heterogêneos.

Análises de escalonabilidade podem ser utilizadas para prever o comportamento temporal das aplicações através de testes, os quais determinam quando as restrições temporais das tarefas serão atendidas em tempo de execução. Tais análises podem ser caracterizadas por uma série de fatores incluindo as restrições do modelo computacional adotado (ex. processador, memória, relações de precedência, etc) e a cobertura do teste de escalonabilidade adotado. Testes suficientes e necessários são ideais, porém para muitos modelos computacionais a adoção de tais testes não podem ser considerados. Testes suficientes são geralmente menos complexos, porém são pessimistas. É importante notar que fatores de utilização elevados podem ser assegurados através da escolha apropriada dos períodos das tarefas do conjunto. Em particular, se os períodos das tarefas são harmônicos (os períodos das tarefas são múltiplos entre si) então a escalonabilidade do sistema é garantida para fatores de utilização de até 100%.

Os modelos de escalonamento tradicionalmente empregados em sistemas de tempo real apresentam-se pouco flexíveis, tornando seu uso muitas vezes impraticável devido ao volume de recursos reservados necessários para o pior caso de tempo de execução das tarefas. Por outro lado, existem propostas para o uso de abordagens adaptativas onde a utilização dos recursos do sistema é realizada de maneira mais eficiente, possibilitando ainda que os sistemas sejam executados em ambientes não deterministas.

O uso de técnicas adaptativas e de reserva de recursos podem ser utilizadas como abordagens complementares, no sentido de oferecer garantias para as tarefas aproveitando a melhor utilização dos recursos do sistema factível através do uso de abordagens adaptativas.

Em razão do recente sucesso do IEEE 802.15.4, o padrão ZigBee assumiu seu uso na camada física e MAC enquanto que o WirelessHart assumiu o seu uso como camada física. Esta adesão tem reforçado mais ainda o interesse pelo IEEE 802.15.4 para executar aplicações de automação, em especial, aplicações industriais e principalmente, demandado suportes tempo real para os protocolos de rede que suportam essas aplicações.

Apesar das especificações do IEEE 802.15.4 citarem a possibilidade de três diferentes topologias, em virtude das topologias ponto a ponto e em árvore terem a maior parte de seus mecanismos definidos fora da especificação IEEE 802.15.4 (ex. definidas na especificação ZigBee).

Esta tese investiga o uso de técnicas de escalonamento deterministas em cenários e aplicações não deterministas – como é o caso das RSSFs. Este trabalho possui interesse em investigar o emprego de mecanismos de provimento de QoS para aplicações compatíveis com o padrão IEEE 802.15.4.





### 3 TRABALHOS RELACIONADOS

Este capítulo versa sobre os principais trabalhos encontrados na literatura relacionados a esta tese. Como estamos enfocando neste trabalho o problema de escalonamento de requisições de compartimentos em coordenadores de uma PAN IEEE 802.15.4, e consideramos que a carga submetida ao sistema pode ultrapassar a sua capacidade, basicamente o problema tratado neste trabalho pode ser considerado como um problema de escalonamento de recursos na sobrecarga (*overload scheduling*). Este problema tem sido tratado por alguns trabalhos na literatura, porém em contextos e suposições completamente diferentes dos propostos para este trabalho de doutoramento. Neste capítulo serão apresentados trabalhos que tratam diretamente o problema de escalonamento de tarefas sobre um conjunto reduzido de recursos, considerando que algumas tarefas podem perder seus *deadlines* sem que o sistema entre em um estado de falha. Buscando facilitar a discussão e análise sobre esses trabalhos, estes foram divididos em três grandes grupos, apresentados a seguir.

#### 3.1 ESCALONAMENTO DE TAREFAS QUE TOLERAM DESCARTES DE ATIVAÇÕES

Conforme aludido no Capítulo 2, aplicações de tempo real são caracterizadas por um conjunto de tarefas com restrições temporais que cooperam e interagem entre si, almejando o funcionamento adequado do sistema. As interações entre estas tarefas são, muitas vezes, realizadas através de trocas de mensagens periódicas. Desta forma, uma tarefa e sua respectiva mensagem são periodicamente ativadas, e cada instância da tarefa, por sua vez, possui usualmente um *deadline* associado a sua execução.

Em muitas aplicações, o não cumprimento de um *deadline* por parte de uma tarefa implica em graves consequências para o sistema controlado. Porém, existem muitas aplicações de tempo real onde não é necessário que todas as ativações periódicas de uma tarefa ou de uma mensagem tenham seus *deadlines* atendidos. Essas tarefas são denominadas neste texto como tarefas que toleram descartes (apesar do fato que em algumas abordagens apresentadas neste capítulo não ocorrer, de fato, o descarte da ativação, mas sim, a perda do *deadline* desta).

### 3.1.1 Escalonamento DBP

Em meados da década de 90, Hamdaoui e Ramanathan (HAM-DAOUI; RAMANATHAN, 1995a) introduziram a noção de *deadlines* (m,k)-firm para escalonamento de tarefas periódicas no contexto de aplicações de tempo real com *deadlines* firmes. Usando este modelo, um sistema pode ser projetado para tolerar perdas de *deadline*, contanto que o número de perdas seja limitado e precisamente espaçada.

De acordo com este modelo clássico, garantias de serviço são oferecidas às tarefas periódicas através de uma heurística dinâmica de priorização de tarefas denominado *Distance Based Priority* (DBP). Segundo a política DBP, a tarefa mais próxima de perder o  $m + 1^{th}$  *deadline* em uma janela de  $k$  *deadlines* consecutivos recebe a maior prioridade do sistema. O modelo tolera até  $k - m$  descartes por tarefa, e caso este valor seja ultrapassado, diz-se que a tarefa encontra-se em um estado de falha dinâmica. Para aplicar este conceito, foi proposto que o estado de execução de cada tarefa seja mantido em um histórico. O histórico de execução de uma tarefa é uma  $k$ -tupla que armazena as  $k$  recentes execuções de uma tarefa. Uma forma factível de implementar o histórico é como segue: seja 0 e 1 a representação de perda e cumprimento de *deadlines*, respectivamente. Para cada novo estado produzido, o histórico é movido da direita para a esquerda e um novo estado é adicionado ao lado mais a direita do histórico. Por exemplo, uma tarefa com  $k = 3$  e histórico “110” denota que a última ativação da tarefa perdeu seu *deadline*, enquanto que as duas ativações anteriores foram atendidas.

A Figura 24 apresenta o diagrama de transição de estados possíveis para uma tarefa com restrições (2,3)-firm. Desta forma, a tarefa  $j$  pode estar localizada em um dos  $2^{k_j}$  estados possíveis. Os estados hachurados possuem menos que dois *deadlines* atendidos (menos que dois 1's) e portanto encontra-se em um estado de falha dinâmica.

A principal ideia por trás da heurística DBP é que as tarefas se tornem mais prioritárias a medida em que suas perdas de *deadline* se aproximem do número máximo de perdas suportado pela tarefa. Quando uma tarefa encontra-se próxima de um estado de falha dinâmica, sua próxima ativação recebe alta prioridade, aumentando assim suas chances de ter seu *deadline* cumprido em sua próxima execução.

Os autores conduziram experimentos onde foi possível observar o comportamento do DBP frente a abordagens sem priorização de tarefas e em sistemas heterogêneos. Os resultados indicaram que o esquema de atribuição dinâmica de prioridades proposto pela abordagem DBP re-

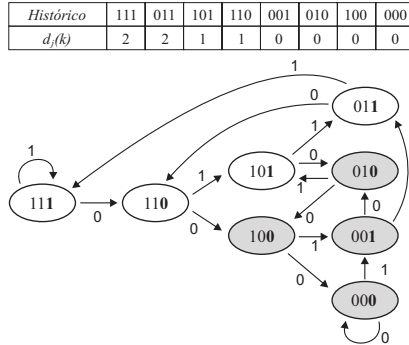


Figura 24: Distância para falha para uma tarefa  $(2,3)$ -firm (baseado em (HAMDAOUI; RAMANATHAN, 1995a)).

sultou em uma diminuição substancial na probabilidade de ocorrência de falhas dinâmicas, sem que as perdas de *deadline* das outras tarefas do sistema fossem demasiadamente afetadas. Porém, importante destacar que os resultados foram obtidos apenas em condições de carga moderada, onde os recursos disponíveis no sistema eram levemente superiores ou iguais à demanda do sistema.

### 3.1.2 Escalonamento Skip-Over

Quase ao mesmo tempo que o trabalho anteriormente descrito, Koren e Shasha (KOREN; SHASHA, 1995) apresentaram o trabalho intitulado *Skip-Over*. Este trabalho considera um conjunto de tarefas periódicas que toleram descartes (*skips*). Estes descartes são caracterizados por um parâmetro de descarte  $s$ , o qual representa a tolerância desta tarefa em perder *deadlines*.

Uma tarefa é dividida em duas instâncias: vermelha e azul. Uma instância de tarefa vermelha deve ser executada antes de seu *deadline*; uma instância de uma tarefa azul pode ser abortada a qualquer momento. No entanto, a distância entre dois descartes consecutivos deve estar no intervalo  $2 \leq s \leq \infty$ , o qual denota a tolerância de uma tarefa em suportar perdas de *deadlines*. Quando uma tarefa possui seu parâmetro  $s = \infty$  não são permitidas perdas de *deadline*<sup>1</sup>.

Este trabalho introduziu também um teste de escalonabilidade

<sup>1</sup>Equivalente a uma tarefa com restrição  $(1,1)$ -firm proposto por (HAMDAOUI; RAMANATHAN, 1995a).

suficiente, mas não necessário, no qual, em alguns casos, tarefas são descartadas desnecessariamente. O teste é apresentado nas equações 3.1 e 3.2, e é baseado no teste por demanda de processador do EDF. Este teste aponta que um conjunto de tarefas periódicas é escalonável se e apenas se  $\forall L \geq 0$ .

Dado um conjunto  $\Gamma = T_i(p_i, c_i, s_i)$  de tarefas periódicas que toleram descartes de algumas ativações, então:

$$L \geq \sum_{i=1}^n D(i[0, L]) \quad \forall L \geq 0 \quad (3.1)$$

Onde,

$$D(i[0, L]) = \left( \left\lfloor \frac{L}{p_i} \right\rfloor - \left\lfloor \frac{L}{p_i s_i} \right\rfloor \right) c_i \quad (3.2)$$

é uma condição suficiente para verificar a escalonabilidade do conjunto  $\Gamma$ .

Os autores propõem ainda alguns algoritmos de escalonamento *on-line* para sistemas que toleram descartes. Dentre eles destacam-se os algoritmos:

- *Red Task Only (RTO)*: Este algoritmo nunca tenta escalonar tarefas azuis. As tarefas vermelhas são escalonadas segundo o EDF. No caso especial onde existam apenas tarefas vermelhas, o algoritmo RTO é ótimo (KOREN; SHASHA, 1995).
- *Blue When Possible (BWP)*: Este algoritmo, mais flexível que o RTO, escalona as tarefas azuis quando sua execução não prejudica a execução de nenhuma tarefa vermelha. O algoritmo escalona as tarefas vermelhas segundo o EDF e caso não existam tarefas vermelhas, então executa uma tarefa azul.
- *Rate Monotonic - RTO (RM-RTO)*: Similar ao RTO, executa as tarefas vermelhas, porém o algoritmo de suporte utilizado nessa abordagem é um algoritmo de prioridade fixa, onde as prioridades são atribuídas de acordo com o esquema de prioridades do RM.

### 3.1.3 Modelo de tarefas com *deadlines* (n,m)-hard

Em (BERNAT; BURNS, 1997), os autores introduziram o modelo de tarefas com restrições (n,m)-hard. O modelo é semelhante ao (m,k)-firm com a adição de uma abordagem de escalonamento baseada em prioridades duplas (*Dual Priority*) proposto inicialmente por (DAVIS; WELLINGS, 1995). O esquema de prioridade duplas melhora a responsividade na execução de tarefas *soft*, executando todas as tarefas *hard* o mais tarde possível.

Na abordagem de prioridades duplas existem três níveis diferentes de prioridades: baixa, média e alta. Tarefas de tempo real *hard* começam a executar na faixa de prioridades baixas, e só são promovidas para faixas de prioridades altas quando a capacidade do sistema é suficiente apenas para atender seus *deadlines*. Desta forma, enquanto tarefas com *deadline hard* não são promovidas, as tarefas *soft* executam na faixa de prioridades médias.

O foco do trabalho é melhorar a responsividade das tarefas de melhor esforço, na presença de tarefas com restrições (n,m)-hard. Os tempos de processador provenientes dos descartes de ativações de tarefas *hard* é utilizado para retardar a promoção das próprias tarefas *hard* para as faixas de prioridades altas, aproveitando para aumentar a escalonabilidade das tarefas *soft*. A abordagem apresentada necessita que um teste de escalonabilidade *off-line* seja realizado para garantir a escalonabilidade das tarefas com restrições *hard*.

Um fato relevante a se destacar neste trabalho, e praticamente inédito na literatura, é o fato de ser uma proposta de um modelo de tarefas no qual tarefas *hard* podem perder *deadlines* sem consequências catastróficas (sem falhar).

### 3.1.4 Escalonamento baseado em janelas

West e Zhang (WEST; ZHANG, 2004) descreveram um algoritmo para escalonamento de pacotes intitulado *Dynamic Window-Constrained Scheduling* (DWCS). O algoritmo leva em consideração a tolerância desses pacotes com relação à violação de *deadlines* de algumas ativações. Os autores introduzem o conceito de *window-constrained* em grupos de *deadlines* consecutivos de pacotes. A janela de restrições ( $W_i = x_i/y_i$ ) define o número de *deadlines* que podem ser perdidos (ou, similarmente, que devem ser atendidos) em uma janela de *deadlines* para pacotes consecutivos em um *stream*. O algoritmo tenta garantir que

não mais que  $x$  *deadlines* em uma janela de tamanho  $y$  sejam perdidos.

<i>Earliest Deadline First</i> (EDF)
<i>Deadlines</i> iguais: ordena pela menor janela $W_i$
<i>Deadlines</i> iguais e $W_i = 0$ : ordena pelo maior $y_i$
<i>Deadlines</i> iguais e $W_i \neq 0$ : ordena pela menor $x_i$
Outros casos: FIFO

Tabela 5: Precedência de execução das tarefas segundo DWCS.

O algoritmo DWCS ordena as tarefas para atendimento com base nos valores da janela de restrição atual e dos *deadlines* relativos das tarefas. A precedência de execução das tarefas é efetuada conforme explicitado na Tabela 5.

---

**Algoritmo 1**  $W_i$  de uma tarefa atualizada antes do *deadline*.

---

```

if ( $y'_i > x'_i$ ) then
   $y'_i = y'_i - 1$ ;
else if ( $(y'_i = x'_i)$  and  $(x'_i > 0)$ ) then
   $x'_i = x'_i - 1$ ;
   $y'_i = y'_i - 1$ ;
  if ( $x'_i = y'_i = 0$ ) then
     $x'_i = x_i$ ;
     $y'_i = y_i$ ;
  end if
end if

```

---

Durante a execução do algoritmo, quando uma tarefa perde um *deadline*, o valor da janela de restrição  $W_i$  das próximas tarefas é decrementada, visando refletir o aumento de prioridade da tarefa. Analogamente, uma tarefa atendida antes de seu *deadline*, ocasiona acréscimo do valor de  $W_i$  das próximas tarefas, refletindo assim a redução de sua prioridade. Os valores de  $W_i$  das tarefas são constantemente atualizados durante a execução do algoritmo. O Algoritmo 1 apresenta a atualização da janela de restrição  $W_i$  de uma tarefa executada antes de seus *deadline*.

Desta forma, o valor da janela de restrição  $W_j$  de qualquer outro *stream*  $S_j \mid j \neq i$ , composta por pacotes que perderam seus *deadlines* é ajustada, conforme ilustrada no Algoritmo 2. Considerando que a abordagem não realiza um teste de admissão, é possível que ocorram violações nas janelas de restrição das tarefas. Uma violação de janela

ocorre quando  $W'_j = x'_j/y'_j \mid x'_j = 0$  e então outro pacote do *stream*  $S_j$  perde seu *deadline*.

---

**Algoritmo 2** Atualização  $W_i$  de uma tarefa que perdeu *deadline*.

---

```

if ( $x'_j > 0$ ) then
   $x'_j = x'_j - 1$ ;
   $y'_j = y'_j - 1$ ;
  if ( $x'_j = y'_j = 0$ ) then
     $x'_j = x_j$ ;
     $y'_j = y_j$ ;
  else if ( $(x'_j = 0)$  and ( $y_j > 0$ )) then
     $y'_j = y'_j + 1$ ;
  end if
end if

```

---

Os autores apresentam um teste de escalonabilidade para qualquer janela de restrições utilizada pela aplicação<sup>2</sup>, porém o teste não é exato. O algoritmo e a análise realizada em (WEST; ZHANG, 2004) apresentam problemas<sup>3</sup> e possibilitam que as restrições informadas pela aplicação sejam violadas. A prova apresentada no artigo, trata exclusivamente do caso onde  $x_i = y_i - 1$ , podendo ocorrer falhas dinâmicas para todos os outros valores de janela de restrição onde:  $(x_i \neq y_i - 1)$ .

Na avaliação da proposta, os autores comparam o DWCS com o DBP analisando as perdas de *deadlines* e violações de janela de restrições. No entanto, a comparação feita é injusta. Os autores mostram que o DWCS é superior ao DBP em situações com cargas baixas, porém o fato de o DWCS utilizar o EDF como primeiro critério de desempate no escalonamento, leva o DWCS a se comportar como o EDF quando a carga do sistema é menor que 100% ( $U \leq 1.0$ ), onde o EDF por si só já é ótimo.

### 3.1.5 Escalonamento baseado em *deadlines* virtuais

West e Zhang reconhecem as limitações da abordagem baseada em janelas descrita na seção anterior e propuseram o *Virtual Deadline Scheduling* (VDS) (ZHANG et al., 2004).

O trabalho apresenta uma abordagem de escalonamento de ta-

---

<sup>2</sup>Considerando que todas as tarefas possuem os mesmos períodos.

<sup>3</sup>Os próprios autores descrevem este problema em artigo publicado a posteriori (WEST; ZHANG, 2007)).

refas com restrições parecidas com  $(m,k)$ -firm. O algoritmo proposto no trabalho, tenta oferecer garantias para tarefas com períodos distintos. O valor de *deadline* virtual de uma tarefa é calculado em pontos específicos dentro da janela de execução das tarefas, com o intuito de assegurar oferecimento proporcional de serviços às tarefas. Os autores consideram no trabalho que as tarefas podem ter atendimentos consecutivos dentro de um mesmo período; em outras palavras,  $m_i$  *deadlines* são atendidos dentro de uma janela de atendimento de  $k_i.T_i$ . No entanto, esta concepção no modelo de tarefas é inadequada para uma ampla variedade de tarefas, considerando que tarefas podem ter seus atendimentos descartados em vários períodos consecutivos, sendo assim seus atendimentos postergados até os últimos períodos compreendidos dentro da janela de  $k_i.T_i$ . A Figura 25 ilustra a limitação imposta pela abordagem.

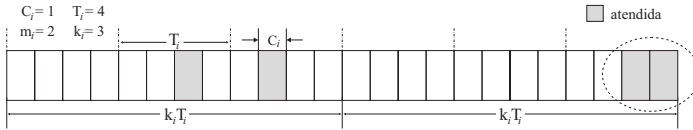


Figura 25: Janela de atendimento de *deadlines*  $k_i.P_i$ .

### 3.2 ESCALONAMENTO TDMA

Um problema fundamental em sistemas distribuídos de tempo real é o compartilhamento de recursos de comunicação entre os diferentes nodos que compõem a rede, de tal forma que suas restrições de tempo real sejam respeitadas. Segundo (CACCAMO; ZHANG, 2002), os atuais protocolos de controle de acesso ao meio (MAC) não são adequados para redes de sensores, pois as redes de sensores se diferem essencialmente das redes *ad hoc* em três principais aspectos:

1. As mensagens trocadas dentro da rede são em sua grande maioria periódicas e precisam de garantias temporais;
2. Diferentes sensores podem coletar as mesmas informações criando grande redundância de dados;
3. Nodos são tipicamente fixos em redes de sensores, enquanto os objetos a serem controlados encontram-se no interior da rede.



Estes aspectos levam em direção a desafios difíceis de tratar pelos protocolos MAC em redes *ad hoc*, como o conhecimento de qual nodo da rede necessita do meio de comunicação em um dado instante de tempo. Alguns protocolos de comunicação consideram que o tráfego da rede é intrinsecamente aleatório, esta hipótese não se aplica diretamente em redes de sensores sem fios (CACCAMO; ZHANG, 2002), onde os dados amostrados são transmitidos periodicamente pelos nodos da rede.

Protocolos de comunicação baseados no modelo *Time Division Multiple Access* (TDMA) buscam resolver este problema de compartilhamento de recursos atribuindo compartimentos de tempo para a transmissão de mensagens pelos nodos. Tipicamente, estes protocolos de comunicação operam baseados em ciclos TDMA, onde um nodo recebe um ou mais compartimentos de tempo para efetuar suas transmissões. Usualmente, cada compartimento possui tamanho e o número de compartimentos disponíveis por ciclos fixos. Desta forma, um ciclo TDMA possui duração fixa e conhecida onde os limites de atraso nas transmissões de mensagens são conhecidos.

A maioria das pesquisas em comunicação TDMA está relacionada ao problema de encontrar uma escala apropriada (algoritmo de escalonamento) para oferecer garantias temporais para as mensagens de tempo real. Infelizmente, a flexibilidade em atribuir compartimentos de tempo para os nodos segundo estas abordagens de escalonamento tem um preço. Um compartimento não utilizado é desperdiçado não podendo ser alocado para nenhum outro dispositivo de tempo real da rede. Outro ponto interessante a ser ressaltado é que alguns dispositivos com mensagens periódicas necessitam de compartimentos de tempo em apenas alguns ciclos TDMA, enquanto que na maioria dos ciclos o compartimento de tempo acaba desperdiçado.

Na área de escalonamento de sistemas de tempo real, escalonamento baseado em TDMA é comumente conhecido como abordagem por executivo cíclico.

### **3.2.1 Protocolo de comunicação com compressão de compartimentos**

O trabalho introduzido por (ANDERSSON et al., 2005, 2008) considera protocolos TDMA que toleram compressão de compartimentos (SS - *Slot Skipping*). Um compartimento de tempo é comprimido quando não é utilizado. Desta forma, o próximo compartimento de tempo pode começar sua transmissão antes do tempo previamente reservado,

em benefício de tráfego de tempo real. A abordagem proposta apresenta uma análise de escalonabilidade considerando uma compressão de compartimentos em redes TDMA/SS. Os autores assumem que todas as mensagens do sistema são conhecidas, e que cada nodo escalona suas mensagens na fila de saída em conformidade com a política Taxa Monotônica, em contraste com algumas outras abordagens que utilizam política FIFO nas filas de mensagens em cada nodo (TOVAR et al., 2002). A análise é apresentada em duas etapas. Na primeira etapa considera-se o caso onde apenas uma mensagem pode ser transmitida por um nodo dentro de um mesmo ciclo TDMA. Posteriormente, uma generalização da abordagem foi proposta para o caso onde um nodo pode transmitir um número fixo de mensagens dentro de um mesmo ciclo TDMA, os autores apresentam também um algoritmo que indica como os nodos devem transmitir suas mensagens neste caso, pois caso os nodos gerem muitas mensagens para serem transmitidas, a fila de saída pode ficar sobrecarregada, aumentando assim os tempos de transmissão das mensagens.

Os autores apresentam algumas suposições sobre a análise proposta. Assumem que o *deadline* das tarefas é menor ou igual ao período ( $D_i \leq P_i$ ), que o tempo de transmissão de uma mensagem em um nodo é de exatamente um compartimento de tempo e que uma mensagem  $S_i$  de um nodo  $k$  termina sua transmissão antes da chegada de uma nova mensagem  $S_j$  na fila de saída. Segundo a proposta, quando um nodo termina sua execução ele envia uma mensagem de conclusão de transmissão avisando que não possui mais mensagens para transmitir, desta forma um nodo que possui mensagens de tempo real para serem transmitidas pode iniciar suas transmissões imediatamente. Em um protocolo TDMA tradicional, um nodo que não deseja transmitir mensagens no tempo reservado simplesmente permanece mudo durante o tempo do compartimento.

### 3.2.2 Escalonamento de mensagens em redes de sensores/atuadores

Escalonamento de mensagens, segundo o protocolo TDMA, tem sido objeto de pesquisa durante os últimos anos (DONG et al., 2000; HAN et al., 1996; KOPETZ; GRUNSTEIDL, 1994), porém esses trabalhos assumem que os compartimentos de tempo e, conseqüentemente os ciclos TDMA, devem ser suficientemente grandes para suprir as necessidades de transmissão de todos os nodos da rede. Usualmente, as escalas de

transmissão desses nodos são realizadas em tempo de projeto, considerando o conhecimento dos tempos de transmissão de todos os dispositivos da rede. No entanto, o trabalho proposto por (CARLEY et al., 2003), cria a escala em tempo de execução de maneira distribuída.

Em (CARLEY et al., 2003) um escalonador de mensagens é utilizado para determinar qual mensagem terá acesso ao meio em um determinado instante de tempo. Prioridades são atribuídas as mensagens com a finalidade de ditar qual mensagem poderá transmitir em um dado instante. Um escalonador, executado em cada nodo, deve gerenciar todas as mensagens de forma a resolver os períodos de contenção. Em suma, a proposta busca aperfeiçoar a transmissão das mensagens em redes densas, onde muitas vezes se torna impraticável realizar todas as trocas de mensagens entre os milhares de nodos da rede. Devido às restrições de recursos, o protocolo realiza uma otimização em cada nodo da rede, permitindo apenas que algumas mensagens de interesse do nodo sejam transmitidas em cada ciclo.

Análise experimental realizada pelos autores demonstra a eficiência da abordagem no sentido de que compartimentos de tempo são utilizados sem desperdício. Entretanto, existe a necessidade de protocolos de sincronização entre os nodos da rede, para que o sistema tenha ciência de qual nodo irá transmitir mensagens em cada compartimento. A transmissão de mensagens esporádicas não pode ser efetuada de maneira eficiente, tornando-se também um ponto não tratado pela proposta.

### 3.2.3 Protocolo de acesso ao meio com priorização implícita

Caccamo e Zhang apresentam em (CACCAMO; ZHANG, 2002) uma arquitetura de rede adequada para redes de sensores juntamente com um protocolo de acesso ao meio baseado no algoritmo de escalonamento EDF. A rede de sensores é dividida em várias células (*celular structure*), onde mensagens transmitidas entre nodos de uma mesma célula utilizam transmissão *multicast* e mensagens transmitidas entre nodos de células adjacentes utilizam canais *Frequency Division Multiplexing* (FDM) diferentes com o objetivo de evitar conflitos nas transmissões.

A idéia principal proposta pelos autores consiste em explorar a natureza periódica do tráfego de dados em redes de sensores. Desta forma, o acesso ao meio pode ser alcançado utilizando um esquema de priorização implícita ao invés de contar com mecanismos de controle

de pacotes, além de diminuir a quantidade de mensagens de reserva de recursos, recorrente em muitos protocolos de controle de acesso ao meio tradicionais.

Os autores assumem que todos os nodos da rede têm ciência de todas as mensagens do sistema. Cada nodo calcula os *deadlines* de todas as mensagens da rede e assim define, de acordo com o algoritmo EDF, quais mensagens são transmitidas a cada execução do algoritmo. Cada nodo possui uma tabela que armazena as restrições temporais dos nodos que compõem a rede, a cada ciclo o algoritmo é executado priorizando as transmissões dos nodos segundo a política EDF.

O protocolo desenvolvido oferece atendimento rápido para os eventos urgentes: em virtude da política de escalonamento utilizada ser baseada no EDF, as mensagens com os menores valores de *deadline* absolutos são favorecidas. Porém, este tipo de estratégia apresenta algumas desvantagens.

A primeira, e mais evidente, advém do fato da abordagem precisar ter conhecimento de todas as mensagens a serem transmitidas na rede. Em se tratando de uma rede de sensores sem fios, onde a topologia da rede pode mudar constantemente, o uso de tal abordagem implica no processamento constante destas mudanças de topologia, este processamento pode vir a consumir alguns recursos escassos da rede mais rapidamente, por exemplo, a bateria dos nodos. Outro aspecto a ser considerado neste tipo de estratégia diz respeito à sincronização dos nodos, cada dispositivo deve estar com seu relógio sincronizado em relação aos outros dispositivos da rede, e esta, obviamente é uma tarefa não-trivial. Finalmente, a transmissão de mensagens esporádicas não pode ser escalonada de forma eficiente utilizando a abordagem proposta.

### 3.3 ESCALONAMENTO DE GTS EM REDES IEEE 802.15.4

A alocação de Compartimentos de Tempo Garantido (GTS) possibilita aos dispositivos acesso ao meio de comunicação sem períodos de contenção, dentro do CFP. O Coordenador PAN é responsável pelo gerenciamento da alocação de GTSS e determina a duração do CFP em um superquadro. O número máximo de GTSS que podem ser alocados pelo Coordenador PAN é sete. Por padrão, os GTSS são alocados segundo ordem FIFO (*First-in, First-out*), tal método de ordenação não é adequado quando se faz necessário a priorização de mensagens. Recentemente, uma quantidade considerável de trabalho vem sido desenvolvido objetivando melhorar o mecanismo de alocação de GTSS em

redes IEEE 802.15.4 (SEMPREBOM et al., 2009c; KOUBAA et al., 2008; CHENG et al., 2007; CHEN J. FERREIRA, 2011; ANDERSSON et al., 2008; TOVAR et al., 2002).

O trabalho apresentado em (KOUBAA et al., 2006b, 2008) melhora a utilização de largura de banda permitindo que múltiplos dispositivos compartilhem o mesmo conjunto de GTS através de um algoritmo de escalonamento *round-robin*. Basicamente a idéia por trás da abordagem consiste em compartilhar o mesmo GTS entre múltiplos nodos ao invés de tê-lo reservado exclusivamente para um único nodo. Se existe um escalonador que atenda as especificações de todos os nodos que solicitam GTSs, então compartilhar um GTS entre vários nodos significa que este é dinamicamente alocado para nodos diferentes em cada superquadro, de acordo com tal escalonador. O trabalho considera duas importantes restrições: (i) os candidatos a compartilharem o mesmo GTS possuem taxas de chegadas similares e (ii) as tarefas não utilizam os recursos previamente reservados em todos os períodos de superquadros. Adicionalmente, se os nodos frequentemente solicitam alocação de GTSs, os compartimentos garantidos serão rapidamente exauridos devido ao número reduzidos de GTS disponíveis durante o CFP em redes IEEE 802.15.4.

Em (CHENG et al., 2007) os autores propuseram uma abordagem simples onde o CFP é dividido e dezesseis mini compartimentos. Em (CHEN J. FERREIRA, 2011) os autores apresentam uma abordagem mais flexível e propõem a divisão do CFP em mais de dezesseis mini compartimentos de tempo, habilitando assim mais do que sete dispositivos realizarem suas transmissões durante o CFP. O trabalho oferece *deadlines* mais apertados para as aplicações, uma vez que os *deadlines* das tarefas não podem exceder a duração do superquadro ( $D_i < BI$ ). A abordagem apresentada pelos autores é realizada através de um algoritmo de alocação de largura de banda realizado *off-line*, onde as configurações de alocação de GTS são configuradas *a priori* nos dispositivos sensores. Ademais, a abordagem só mantém compatibilidade com o padrão para aplicações que utilizam os períodos sem contenção, não mantendo compatibilidade com o padrão IEEE 802.15.4 no que tange a utilização de períodos livres de contenção.

Um modelo de alocação dinâmico de compartimentos garantidos em redes IEEE 802.15.4 foi introduzido em (CHO et al., 2010). A abordagem proposta nomeada *Guaranteed Dynamic Priority Assignment* (GDPA) provê uma probabilidade limitada de ocorrências de falhas dinâmicas para tarefas com *deadlines* (m,k)-firm quando o sistema encontra-se sobrecarregado. O algoritmo proposto toma decisões de es-

calonamento baseado no algoritmo DBP em conjunto com o algoritmo EDF.

### 3.4 TESTES DE ESCALONABILIDADE

Testes de escalonabilidade são fundamentais para determinar a possibilidade de existência de uma escala viável. Nas primeiras pesquisas sobre escalonamento, alguns algoritmos de escalonamento clássicos executavam seus testes de escalonamento baseados no conceito de utilização de processador. O teste de utilização é um teste necessário na maioria dos algoritmos de escalonamento. Ele é um teste suficiente ou exato apenas quando assume-se modelos de tarefas bastante simplificados. Por exemplo, no trabalho clássico de Liu e Layland (LIU; LAYLAND, 1973), o teste de utilização do processador é exato no caso do EDF e suficiente no algoritmo Taxa Monotônica. Esses algoritmos de escalonamento exigiam originalmente que o conjunto de tarefas fosse formado por tarefas periódicas e independentes com o *deadline* relativo coincidindo com o período das tarefas.

Uma série de trabalhos estenderam o modelo de tarefas para esses algoritmos e propuseram testes de escalonabilidade mais complexos. Sha e outros (SHA et al., 1990) propuseram o protocolo de *ceiling* com forma de permitir que tarefas possuam relações de precedência entre si e possam se sincronizar. Em (TINDELL et al., 1993) é apresentado um teste de escalonabilidade que permite o escalonamento de tarefas periódicas com *deadlines* arbitrários e *jitter* de liberação, além de considerar tarefas esporádicas.

#### 3.4.1 Teste de escalonabilidade para o DBP

Em (RAMANATHAN, 1999) Ramanathan propôs uma técnica para gerência de processador na sobrecarga em aplicações de controle. A técnica apresentada descarta de forma seletiva instâncias de tarefas objetivando reduzir a utilização efetiva da tarefa. A abordagem é implementada com base em dois processos executando de forma concorrente: processo de atendimento e processo de atribuição de prioridades.

O primeiro, é um escalonador baseado em prioridade preemptivo, o qual atende instâncias de tarefas em uma fila de prontos. Esta política de escalonamento é implementada com base em duas regras. Primeiro, o servidor nunca fica ocioso se existe uma tarefa para ser executada.

Segundo, o servidor sempre executa a tarefa com a maior prioridade que está aguardando pelo atendimento.

O processo de atribuição de prioridades, por outro lado, atribui prioridades para as tarefas e as insere em uma fila de prontos. As tarefas são classificadas em mandatórias e opcionais. Tarefas mandatórias recebem prioridades de forma que o atendimento de seus *deadlines* são garantidos. As instâncias de tarefas classificadas como opcionais, recebem as menores prioridades e não possuem garantias no atendimento de seus *deadlines*. A classificação das instâncias das tarefas em mandatória ou opcional é baseada nos valores de  $m_i$  e  $k_i$  das tarefas. Instâncias das tarefas  $\tau_i$  são ativadas nos instantes  $aT_i$ , para  $a = 0, 1, 2, \dots$ . Uma instância ativada no instante  $aT_i$  é classificada como mandatória se:

$$a = \left\lceil \left\lfloor \frac{am_i}{k_i} \right\rfloor \cdot \frac{k_i}{m_i} \right\rceil \quad (3.3)$$

Note que a classificação (Equação 3.3) não é necessariamente ótima em termos de atendimento de garantias (m,k)-*firm* para cada tarefa  $\tau_i$  na aplicação. O teste de escalonabilidade é suficiente (suficiente, mas não necessário) bastante restritivo, onde um conjunto considerável de tarefas que poderiam ser atendidas são descartadas pelo teste. De acordo com (QUAN; HU, 2000), a classificação ótima de jobs em mandatório ou opcional para um algoritmo de escalonamento dinâmico é um problema que não pode ser tratado em tempo polinomial (NP-difícil) considerando períodos, tempo de computação e valores de  $k$  arbitrários.

### 3.4.2 Teste de escalonabilidade exato para tarefas (m,k)-*firm*

Desde o trabalho seminal proposto por Hamdaoui e Ramanathan (HAMDAOUI; RAMANATHAN, 1995a) a maioria dos trabalhos propostos na literatura abordam o problema assumindo que a sequência inicial da janela de restrições (*k-sequence*) é formada unicamente por 1's, ou seja, o histórico inicial de cada tarefa não considera estados faltosos no início da execução do algoritmo. Porém, esta hipótese não é verdadeira em todos os casos (GOOSSENS, 2008). Como trabalhos da literatura que assumem esta premissa temos o trabalho introduzido por (LI et al., 2004) onde é apresentado um teste de escalonabilidade suficiente, considerando o uso do escalonador segunda a política DBP relacionados com escalonamento de tarefas com restrições (m,k)-*firm* e o trabalho (HAMDAOUI; RAMANATHAN, 1995b) onde é avaliada a probabilidade de

falha dinâmica das tarefas.

O trabalho (GOOSSENS, 2008) apresenta um teste de escalabilidade exato para aplicações com restrições  $(m,k)$ -*firm*, comparando a cada superperíodo o estado do sistema com estados obtidos em superperíodos anteriores. O teste é terminado uma vez que a escala de execução das tarefas se repete. Em outras palavras, no pior caso, todas as combinações de padrões  $(m,k)$ -*firm* possíveis entre para todas as tarefas do sistema devem ser testadas. A complexidade computacional deste teste não é adequada para sistemas com cargas dinâmicas, onde a carga do sistema varia de acordo com o tempo ou em dispositivos com restrições de hardware (e.g. memória, energia, processamento, etc) como é o caso das redes de sensores sem fios. O teste pode ser tratado computacionalmente, porém apenas em abordagens *off-line*. Ademais, o trabalho utiliza o DBP como algoritmo de escalonamento o qual não é ótimo no sentido de que a atribuição de prioridades adotado pelo DBP baseia-se apenas no valor de distância para falha de cada stream sobre suas restrições  $(m,k)$ -*firm*. O algoritmo de escalonamento não considera outros parâmetros das tarefas como período, *deadline* ou tempo de computação.

O trabalho apresentado em (QUAN; HU, 2000) propõe uma heurística para geração de padrões de jobs em mandatório ou opcional utilizando um algoritmo genético. O trabalho também traz um teste de escalabilidade suficiente, onde manipulando os períodos das tarefas, um novo conjunto de tarefas com períodos harmônicos é gerado. A solução proposta é adequada apenas para abordagens *off-line*, onde o algoritmo genético probabilístico busca encontrar um particionamento ideal de classificação de tarefas em mandatório ou opcional, produzindo, desta forma uma escala de execução factível para o conjunto de tarefas.

Em (LI et al., 2006) um teste de escalabilidade suficiente, baseado em (JEFFAY et al., 1991) é proposto. O trabalho considera um escalonador não-preemptivo baseado no algoritmo de atribuição de prioridades DBP. A abordagem foi avaliada em um cenário de controle veicular.

### 3.5 CONSIDERAÇÕES SOBRE AS ABORDAGENS

As abordagens baseadas em TDMA necessitam de uma tabela contendo os tempos de chegada de todos os nodos que compõem a rede seja previamente conhecida. Tradicionalmente esta tabela indica



qual nodo deve acessar o meio em cada unidade de tempo. A vantagem destas abordagens advém do determinismo oferecido às aplicações. Entretanto técnicas baseadas em TDMA são tipicamente inflexíveis, impondo a periodicidade das tarefas em ciclos TDMA estabelecidos *a priori*, além de não conseguirem capturar a folga na utilização dos compartimentos por uma tarefa ou mesmo a não ocupação de um compartimento em um ciclo TDMA. Caso a topologia da rede mude com certa frequência, mensagens de sincronização deverão ser anunciadas para os nodos para que a tabela TDMA seja atualizada, adicionando ou removendo um novo dispositivo na escala. O trabalho apresentado por (ANDERSSON et al., 2005) busca resolver o problema de desperdício de compartimentos através do uso de técnicas de compressão de compartimentos.

A maioria dos trabalhos relacionados atualmente encontrados na literatura, consideram o caso onde no máximo sete dispositivos podem utilizar GTSs durante o período do superquadro. Porém, em (CHEN, 2008; CHEN et al., 2009) os autores adaptam a estrutura do superquadro, objetivando prover um esquema de alocação de GTS baseado em TDMA, no qual o CAP não existe e a estrutura dos quadros de dados são simplificados. Entretanto, tais características prejudicam a utilização do protocolo IEEE 802.15.4 devido a ausência de oportunidade de transmissão de aplicações não tempo real durante os períodos com contenção.

Técnicas baseadas no modelo de tarefas  $(m,k)$ -firm podem ser utilizadas para representar vários tipos de *deadlines*. Por exemplo, *deadlines hard* podem ser representados na forma de uma tarefa com restrições  $(1,1)$ -firm, enquanto *deadlines soft* podem ser representados por valores de  $m$  e  $k$  escolhidos como  $(1, \infty)$ -firm. Desta forma, a representação de tarefas na forma de restrições  $(m,k)$ -firm mostra-se vantajosa pois os *deadlines* das tarefas podem ser representados mais fielmente dentro de um mesmo sistema.

A adoção do modelo de tarefas  $(m,k)$ -firm juntamente com testes de escalonabilidade podem oferecer garantias deterministas para aplicações que executam em ambientes dinâmicos, onde eventuais ocorrências de sobrecargas transientes levam o sistema a se comportar de tal forma que as restrições  $(m,k)$ -firm das tarefas sejam respeitadas, buscando a não ocorrência de falhas dinâmicas no sistema. Algumas abordagens que utilizam o modelo  $(m,k)$ -firm podem ainda considerar algoritmos de escalonamento ditos robustos (vide seção 2.1.6), uma vez que em situações de sobrecarga transientes uma ativação classificada com opcional poderia ser armazenada em uma fila de ativações rejeitadas e

ter seu *deadline* atendido posteriormente, quando uma ativação mandatória conclui sua execução antes de seu *deadlines*.

Mok e Wang mostraram em (MOK; WANG, 2001) que o problema de escalonamento de tarefas  $(m,k)$ -*firm* é NP-difícil, para tarefas com períodos e tempo de computação arbitrários. Os autores propuseram uma abordagem para tarefas que utilizem tempo de computação constante (igual a uma unidade de tempo) e valores de períodos arbitrários. Porém, a abordagem só oferece garantias para as tarefas quando o fator de utilização do sistema é inferior a 50%.

O problema de escalonamento de mensagens em redes de sensores sem fios IEEE 802.15.4, não têm sido profundamente investigado pela comunidade científica, principalmente a modelagem do comportamento das tarefas frente um modelo mais flexível, como o modelo  $(m,k)$ -*firm*.

### 3.6 CONCLUSÕES DO CAPÍTULO

Enquanto é essencial que todas as restrições temporais sejam atendidas em sistemas de tempo real *hard*, sistemas de tempo real *firm* ou *soft* não possuem restrições tão estritas, permitindo algum nível de atraso (*soft*) ou permitindo ainda que alguns *deadlines* sejam perdidos (*firm*). Desta forma, uma quantidade significativa de pesquisa vêm sendo realizada na área de sistemas de tempo real *soft* e *firm*, buscando minimizar os atrasos e quantificar de maneira precisa as perdas de *deadlines* dos sistemas.

STR atuais necessitam de mecanismos para especificar de forma clara e previsível quando alguns *deadlines* podem ser estrategicamente descartados (não atendidos) em prol da melhora do comportamento do sistema com um todo. A adoção destes novos mecanismos se faz necessária, pois podem suavizar o pessimismo presente em alguns parâmetros dos sistemas de tempo real (em especial o tempo de computação  $C_i$ ) em cenários de pior caso de execução (assumindo, por exemplo que todas as tarefas são iniciadas simultaneamente). Admitem ainda que a QoS do sistema seja degradada de forma equânime entre as tarefas. E principalmente, a inferência de quais tarefas precisam ser descartadas durante situações de sobrecarga transientes, para que as restrições das outras tarefas presentes no sistema possam ser corretamente atendidas. A adoção destes mecanismos possibilita que sistemas de tempo real sejam projetados com fatores de utilização  $U$  acima de 100%, porém com caso médio de utilização  $U \leq 100\%$ .

Este capítulo buscou apontar alguns trabalhos relevantes referen-

ciados na literatura, e também alguns trabalhos que apresentavam semelhança com o estudado neste trabalho. Diferentemente dos trabalhos citados anteriormente e considerando que o protocolo IEEE 802.15.4 não oferece mecanismos de provimento de QoS às aplicações, esta proposta busca atender requisições com restrições segundo o modelo de tarefas  $(m,k)$ -*firm*.



## 4 MODELO DO SISTEMA E MÉTRICAS DE DESEMPENHO

Este capítulo apresenta inicialmente uma breve introdução acerca das áreas de interesse do tema tratado nesse trabalho. O modelo de sistema adotado para o desenvolvimento das propostas apresentadas nos Capítulos 5 e 6 são discutidos em um segundo momento. As principais métricas de desempenho utilizadas na avaliação das abordagens propostas nesta tese também são apresentadas. Algumas considerações acerca do modelo e métricas adotadas finalizam o capítulo.

### 4.1 INTRODUÇÃO

O tema de interesse desta proposta pode ser descrito resumidamente como sendo resultante da confluência do interesse de três comunidades diferentes: *(i)* comunidade de sistemas de tempo real, no desenvolvimento de abordagens de escalonamento flexíveis; *(ii)* comunidade de controle, na investigação de técnicas que mantêm a estabilidade do sistema mediante descartes de mensagens (perdas de *deadline*); e *(iii)* comunidade de informática industrial, na adoção de redes sem fio.

Com relação à comunidade que investiga escalonamento tempo real, tradicionalmente, os sistemas eram projetados considerando cenários de pior caso de execução. Esta abordagem, no entanto, frequentemente resulta em grande subutilização de recursos. Uma possível solução seria adotar alguma abordagem *ad-hoc*, projetando sistemas considerando seus casos médios de utilização. No entanto, para algumas aplicações de tempo real, como as encontradas no domínio de controle industrial, se faz necessária uma especificação mais precisa de como as perdas de *deadlines* serão distribuídas no tempo (BERNAT; BURNS, 1997). Por este motivo, há algum tempo existem diversos trabalhos que tratam sobre escalonamento adaptativo (escalonamento em condições de sobrecarga). Há uma lista extensa de trabalhos na área, envolvendo abordagens que lidam, por exemplo, com: aplicações elásticas que podem variar suas ativações periódicas (ex. modelo elástico (BUTTAZZO et al., 1995)); aplicações que toleram descartes (ex. modelo *(m,k)-firm* (HAMDAOUI; RAMANATHAN, 1995a) e modelo *Skip-Over* (KOREN; SHA-SHA, 1995)); e aplicações com *deadlines* modelados por funções benefício. No entanto, além destes, há diversos outros trabalhos. Boa parte da comunidade tempo real atual está envolvida de alguma forma

com esse tipo de abordagem. O projeto europeu Artist (ARTIST, 2009), por exemplo, concentra boa parte dessas pesquisas.

Com relação à comunidade de controle, existe atualmente um grande interesse no estudo do comportamento de alguns sistemas de controle em redes (WILLIG, 2008; SCHENATO et al., 2007; HESPANHA et al., 2007; LEMMON et al., 2003). Esses trabalhos vão além de analisar o comportamento do sistema mediante variação do *jitter* de atraso. Há interesse direto no comportamento do sistema mediante presença de descartes de mensagens e perdas de *deadlines*. Esses trabalhos, de uma forma geral, estabelecem métricas (ex. *Quality of Control, Jitter Margin*, etc) para avaliar a qualidade do controle e estabelecem abordagens de escalonamento (estático ou dinâmico) para tratar do atendimento às restrições temporais. Isso vem sendo chamado de *control-scheduling co-design* (projeto do escalonador e projeto de controle feitos concomitantemente) (BUTTAZZO et al., 2007; MARTI et al., 2002, 2004; TRIO, 2009).

O interesse por parte da comunidade que lida com informática industrial no emprego de redes sem fio, não é recente. Contudo, devido aos problemas de confiabilidade e segurança dessas redes, a adoção vem sendo lenta. Recentemente a revista *IEEE Transaction Industrial Informatics* publicou uma edição especial para esta classe de aplicações (WILLIG, 2008; ANASTASI et al., 2011; TOSCANO; BELLO, 2012).

Com relação à rede IEEE 802.15.4, tema de interesse direto deste trabalho, seu surgimento é devido à demanda recente por redes sem fios que forneçam serviços de tempo real e qualidade de serviços às aplicações. Isto é consequência do desenvolvimento recente de tecnologias de acesso sem fios em ambientes domésticos e de escritório. Dentro deste contexto, a família de protocolos IEEE 802.15.4 (802.15.4, 2006) é um dos principais candidatos a se tornar o padrão *de facto* para comunicação sem fios em ambientes com redes LR-WPAN. Um grande número de aplicações podem ser beneficiadas pelas características oferecidas pelas LR-WPANs, tais como automação industrial, aplicações automotivas, aplicações de controle, agricultura, monitoramento, dentre outras. Uma característica comum entre tais aplicações é a existência de múltiplos tipos de tráfego de dados, incluindo dados periódicos, intermitentes e dados de tempo real (ANASTASI et al., 2011; LOW et al., 2005; RAMAMURTHY et al., 2007; RUIZ et al., 2011; TOSCANO; BELLO, 2012).

Aplicações com restrições temporais demandam serviços de comunicação previsíveis. No entanto, o provimento de qualidade de serviço em ambientes sem fios é bastante desafiador. A previsibilidade tempo-

ral implica em que a infraestrutura de comunicação possua certo grau de conhecimento sobre o estado da rede e sobre a demanda futura dos nodos por recursos.

Conforme discutido no Capítulo 2, o protocolo IEEE 802.15.4 oferece garantias temporais às aplicações através do mecanismo de compartimentos de tempo garantido (GTSs). Esta característica é bastante atrativa para aplicações com restrições temporais que executam em redes, pois possibilita a previsão do pior caso de tempo de execução para cada nodo que compõem a rede, uma vez que os dispositivos que alocam GTSs não precisam disputar uns com os outros pelas transmissões. Porém, uma das maiores desvantagens do uso do mecanismo de GTS é o número de dispositivos que podem utilizar o recurso simultaneamente. De acordo com a especificação IEEE 802.15.4 o número de dispositivos que podem alocar GTS dentro de uma mesma PAN é no máximo sete. Desta forma, este recurso escasso pode rapidamente exaurir-se em uma rede composta por vários nodos. Consequentemente, os dispositivos que não possuem GTS alocados devem tentar completar suas transmissões durante o CAP, utilizando o protocolo CSMA/CA.

O mecanismo de alocação de GTS, proposto pela especificação, necessita do envio prévio de uma mensagem de alocação de GTS. No entanto, o protocolo de acesso ao meio CSMA/CA originalmente proposto não oferece suporte de QoS para aplicações de tempo real, incluindo, por exemplo mensagens de alarme ou de controle da rede.

Apesar dos diversos trabalhos citados acima, infelizmente o uso adequado de políticas e mecanismos de tempo real em redes IEEE 802.15.4 não tem sido profundamente investigado, principalmente no que concerne modelos de tarefas que toleram perdas de ativações periódicas (ou perdas de *deadlines*). Oferecer a QoS exigida para essas aplicações, provendo garantias temporais ainda é um desafio em aberto. Modelos de comunicação tradicionalmente adotados em trabalhos relacionados a essas redes geralmente são inflexíveis. Não são capazes de explorar a capacidade de perder *deadlines* sem falhar que algumas aplicações tempo real possuem. Desta forma, no sentido de se adotar modelos mais flexíveis neste trabalho de doutorado, abordagens de escalonamento baseadas no modelo de tarefas  $(m,k)$ -firm são empregados, com o objetivo de evitar a ocorrência de falhas no sistema, principalmente em condições de sobrecarga.

## 4.2 MODELO DO SISTEMA

Considere uma aplicação de tempo real, onde tarefas são distribuídas entre nodos sensores. Considere também que tarefas comunicantes geram fluxos de mensagens que devem ser suportados pela rede de comunicação através da alocação de compartimentos de tempo garantido (GTSs). Desta forma, sempre que mais do que sete solicitações de alocação de GTSs chegarem ao Coordenador PAN, e cada requisição possui uma especificação de QoS na forma de restrições  $(m,k)$ -firm, o Coordenador PAN realizará uma operação de escalonamento considerando as restrições  $(m,k)$ -firm dos dispositivos, objetivando alocação destes fluxos de mensagens em compartimentos garantidos. Portanto, o problema tratado neste trabalho é: como o Coordenador PAN escala o meio de comunicação (CAP e CFP), entre  $N$  nodos sensores solicitantes, respeitando as restrições  $(m,k)$ -firm de cada fluxo de mensagem gerado pelo nodo  $i$ . Formalmente, este problema pode ser especificado como segue:

Considera-se um conjunto  $\Gamma = \{\tau_1, \tau_2, \dots, \tau_n\}$  de  $N$  tarefas periódicas independentes. Cada tarefa  $\tau_i$  gera um fluxo de mensagem  $S_i$  caracterizado por uma 5-tupla  $(T_i, D_i, C_i, m_i, k_i)$ , onde  $C_i$  representa o pior caso de duração da mensagem,  $T_i$  representa o período de duração da tarefa (fluxo de mensagem),  $D_i$  denota o *deadline* da mensagem,  $m_i$  e  $k_i$  representam as restrições  $(m,k)$ -firm da tarefa  $\tau_i$  (fluxo da mensagem  $S_i$ ). No caso do modelo proposto neste trabalho, uma tarefa periódica representa uma aplicação que é executada em um dispositivo sensor, o qual gera fluxos de mensagens periódicas. Adicionalmente, as seguintes suposições sobre as tarefas (fluxo de mensagens) são consideradas:

- A1.**  $1 \leq m_i \leq k_i$ ,  $m_i$  e  $k_i \in \mathbf{Z}^+$ ;
- A2.**  $D_i = T_i$ : o *deadline* da tarefa  $\tau_i$  (fluxo de mensagem  $S_i$ ) é igual ao período;
- A3.**  $C_i = l_i \cdot ut$ ,  $l_i \in \mathbf{Z}^+$ : tempo de execução (duração da mensagem) é múltiplo do slot ( $ut$ ).

A lógica pela suposição A3 é que, devido a natureza preemptiva do modelo proposto, um fluxo de mensagem pode transferir mais do que uma mensagem em cada período ( $C_i \geq 1$ ), mas não pode-se garantir que as mensagens serão transmitidas contiguamente. Por exemplo, um fluxo de mensagens pode transmitir duas mensagens com duração menor que a de um compartimento, mas não pode transmitir uma mensagem com duração de dois compartimentos de tempo. O tamanho dos



compartimentos de tempo, assim como os valores de BI, podem ser ajustados de forma flexível através dos parâmetros *Beacon Order* (BO) e *Superframe Order* (SO), controlados pelo Coordenador PAN.

Neste trabalho, considera-se um agrupamento (*cluster*) IEEE 802.15.4 composto por um conjunto de nodos sensores dentro da área de cobertura de um único nodo denominado Coordenador PAN (Figura 26). Redes organizadas segundo a topologia estrela possuem algumas vantagens em termos de robustez e latência, além da sua fácil organização e sincronização, quando comparadas com topologias ponto a ponto ou agrupamento em árvore (CHEN, 2008). Assume-se que o coordenador PAN configura a rede determinando a estrutura do superquadro através dos parâmetros BO e SO. Cada nodo sensor  $i$  que deseja alocar um GTS por um período de tempo específico, envia uma solicitação de alocação de GTS com suas restrições  $(m,k)$ -firm para o Coordenador PAN.

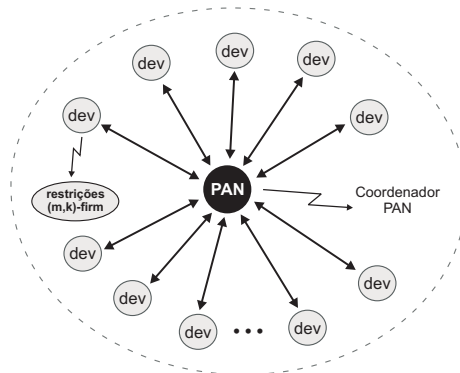


Figura 26: Modelo em topologia estrela.

No modelo de sistema apresentado, mensagens que não podem ser escalonadas durante períodos de acesso livres de contenção podem tentar realizar suas transmissões durante períodos com contenção (CAP).

A análise apresentada nesta tese assume um limite superior de utilização, considerando um canal de comunicação ideal, sem ocorrência de erros. Se um canal com erros for considerado (JONSONN; KUNERT, 2009), a utilização máxima do canal de comunicação será menor. Entretanto, esta análise está fora do escopo desta tese.

No restante deste trabalho, os termos tarefa e fluxo de mensagem (instância e mensagem) serão considerados de forma indiscriminada.

### 4.3 MÉTRICAS DE DESEMPENHO

Diferentes métricas de desempenho foram utilizadas na avaliação das abordagens propostas nesta tese. A primeira proposta de escalonamento em períodos de acesso com contenção (CAP) não oferece garantias às aplicações, apenas oferece mecanismos de priorização de mensagens com o objetivo de reduzir a ocorrência de falhas dinâmicas nos dispositivos sensores.

Esta primeira métrica – ocorrência de falhas dinâmicas (referenciada na Seção 2.2) – é utilizada para mensurar a ocorrência de violações de *deadline* suportados pelas aplicações. Segundo esta métrica, uma tarefa encontra-se em um estado de falha dinâmica caso menos que  $m$  *deadlines* tenham sido atendidos em uma janela de  $k$  *deadlines* consecutivos. Caso contrário, a aplicação encontra-se em um estado estável, sem falhas. É importante ressaltar que a especificação da quantidade de violações de *deadline* suportadas pelas aplicações é dependente dos valores das restrições  $(m,k)$ -*firm* adotados pelas aplicações.

Outra métrica considerada na avaliação da abordagem proposta durante o CAP, refere-se ao percentual de *deadlines* perdidos em cada cenário de carga submetido à rede. O percentual de perdas de *deadline* considera a quantidade de violações de *deadlines* experimentados pelos dispositivos da rede. Note que o número de *deadlines* perdidos é sempre superior ao número de falhas dinâmicas, pois uma perda de *deadline* não acarreta, necessariamente, na ocorrência de uma falha dinâmica.

A segunda proposta de escalonamento em períodos de acesso sem contenção (CFP) apresenta duas soluções deterministas, onde garantias são oferecidas às aplicações através de testes de escalonabilidade e algoritmos de escalonamento *on-line*. A primeira solução, intitulada Slotted DBP (SDBP) é comparada com o algoritmo DWCS (WEST; ZHANG, 2004) e a abordagem FIFO tradicionalmente oferecida pelo padrão IEEE 802.15.4 (802.15.4, 2006). Verificou-se o percentual de ocorrência de falhas dinâmicas na comparação entre as abordagens, além da percentagem de *deadlines* perdidos variando-se a quantidade de nodos e carga submetida ao sistema.

A segunda solução desenvolvida no CFP considera que o padrão de classificação  $(m,k)$ -*firm* das tarefas pode ser modificado objetivando a melhoria na escalonabilidade do sistema. A proposta  $(m,k)$ -*spin* assume um conjunto pré-definido de novos padrões de classificação  $(m,k)$ -*firm* para as tarefas do sistema, estas novas possibilidades de padrões de classificação são alcançados através da adoção de giros, no sentido anti-horário (*left-spin*), no padrão de classificação das tarefas. A primeira

métrica avaliada nesta proposta diz respeito a quantidade de giros adotada pelas soluções. Foram considerados giros apenas na última tarefa e posteriormente em todas as tarefas do conjunto, objetivando aumentar o percentual de tarefas admitidas no sistema.

O tempo de resposta para a tomada das decisões de escalonamento das novas tarefas submetidas ao teste de admissão também foi avaliado. Foram considerados o pior tempo de resposta, o tempo de resposta médio e o desvio padrão observado entre as amostras.

Por fim, avaliou-se o percentual de benefício alcançados pela solução proposta com relação ao trabalho apresentado por (RAMANATHAN, 1999). O benefício avaliado refere-se ao percentual de conjuntos de tarefas admitidos pela abordagem  $(m,k)$ -*spin* com relação aos conjuntos admitidos pela abordagem *Ramanathan* em dois diferentes cenários: conjuntos de tarefas não harmônicos e conjuntos de tarefas harmônicos.

#### 4.4 CONCLUSÕES DO CAPÍTULO

O modelo de sistema proposto para este trabalho foi baseado nas deficiências encontradas em trabalhos existentes na literatura e nas especificações de padrões para RSSF. Por exemplo, a topologia estrela adotada está em conformidade com a topologia estrela proposta nos padrões do ZigBee.

As soluções propostas nessa tese buscam oferecer garantias alocando-se compartimentos garantidos no CFP, considerando as restrições  $(m,k)$ -*firm* de cada dispositivos da rede. As outras  $k - m$  mensagens podem disputar o acesso ao meio no CAP. Nesse sentido no próximo capítulo apresentamos uma abordagem descentralizada que busca priorizar a transmissão de mensagens durante períodos com contenção em redes IEEE 802.15.4.

O modelo delineado nesse capítulo servirá para especificar e avaliar as soluções propostas no capítulos 5 e 6. Algumas restrições adicionais deverão ser consideradas para que as soluções propostas sejam válidas, estas restrições serão descritas, especificamente no capítulo 5.



## 5 ESCALONAMENTO NO PERÍODO DE ACESSO COM CONTENÇÃO

Os primeiros esforços deste trabalho de doutoramento foram na direção de investigar mecanismos de priorização de tráfego tempo real durante períodos de acesso com contenção (CAP) no padrão IEEE 802.15.4. O modelo descrito no Capítulo 4 foi delineado, e os seguintes artigos foram produzidos: (SEMPREBOM et al., 2009, 2009b). Os principais resultados alcançados foram: (i) a proposição de um algoritmo de priorização de mensagens tempo real com restrições  $(m,k)$ -firm intitulado Distributed DBP e (ii) a avaliação experimental em dispositivos sensores compatíveis com o padrão IEEE 802.15.4.

### 5.1 MOTIVAÇÃO

Uma das motivações para o desenvolvimento da abordagem que atua no CAP advém do fato que o protocolo CSMA/CA, oferecido pela especificação, não prevê mecanismos de provimento de QoS para eventos tempo real, incluindo solicitações de alocação de compartimentos, mensagens de alarme e comandos de controle da PAN, isso resulta na degradação do desempenho da rede. Além disso, nodos que possuem mensagens que não foram alocadas no CFP precisam recorrer ao CAP.

Considerando a flexibilidade de algumas aplicações tempo real em tolerar descartes ou perdas de *deadline* sem falhar, e a necessidade de provimento de mecanismos de QoS para aplicações com restrições temporais no protocolo CSMA/CA em redes IEEE 802.15.4, esta proposta investiga mecanismos para priorização de tráfego tempo real, considerando o modelo de tarefas  $(m,k)$ -firm, onde tarefas que estejam com suas execuções próximas a um estado faltoso, recebem as maiores prioridades do sistema.

Diferentemente das abordagens propostas no Capítulo 6, a abordagem de escalonamento no CAP proposta nesta seção não oferece garantias. O objetivo é tentar reduzir a quantidade de falhas dinâmicas, priorizando as tarefas mais próximas da falha dinâmica. Nesta abordagem não há testes de admissão ou alocação de recursos (compartimentos). Apenas tenta-se distribuir o recurso (no caso, a largura de banda da rede) de forma mais inteligente, priorizando-se as mensagens de tarefas mais próximas da falha.

Atualmente, alguns trabalhos têm contribuído com o intuito de

reduzir a latência em redes de sensores sem fio. Particularmente, no caso do padrão IEEE 802.15.4 algumas pesquisas têm procurado aprimorar o protocolo CSMA/CA com *slots*, proposto originalmente pela especificação, com a finalidade de oferecer garantias para as aplicações (KIM et al., 2005; KOUBAA et al., 2006a). No entanto, diferentemente desses trabalhos, esta proposta busca atender requisitos com restrições segundo o modelo (m,k)-*firm*.

## 5.2 ALGORITMO DISTRIBUTED DBP

O algoritmo proposto, denominado Distributed DBP (DDBP), é fruto do interesse deste trabalho de doutorado na investigação de algoritmos de escalonamento dinâmicos (*on-line*), onde o escalonamento é realizado considerando as condições locais de cada nodo. A Figura 27 apresenta o algoritmo Distributed DBP para provimento de QoS entre os dispositivos que compõem a PAN. O algoritmo, baseado no modelo de tarefas (m,k)-*firm*, busca minimizar a ocorrência de falhas dinâmicas no sistema, e foi construído sobre o conceito distância para falha proposto em (HAMDAOUI; RAMANATHAN, 1995b). Com objetivo de se detectar perdas de *deadline*, cada mensagem enviada por um nodo para o coordenador é sinalizada por uma mensagem de reconhecimento enviada no sentido contrário (do coordenador para o nodo).

Existem diversos outros trabalhos que propõem algoritmos de escalonamento considerando restrições (m,k)-*firm* das tarefas (ex. (HAMDAOUI; RAMANATHAN, 1995a; MONTEZ et al., 1999; ZHANG et al., 2004)). No entanto, todos esses trabalhos consideram uma entidade central – um escalonador – que conhece o histórico de todas as tarefas e toma decisões baseadas em um conhecimento global do sistema.

No algoritmo Distributed DBP, após o envio de uma mensagem endereçada ao Coordenador, a heurística verifica o recebimento da mensagem pelo Coordenador PAN através de uma mensagem de reconhecimento ACK, sinalizando o atendimento de uma *deadline* (linha 7 na Figura 27). Caso um dispositivo não consiga efetuar sua transmissão dentro do atual superquadro, uma ocorrência de não atendimento de *deadline* é sinalizada (linha 10). Finalmente, se uma mensagem de reconhecimento ACK não é recebida dentro de um intervalo de tempo determinado pelo dispositivo que transmitiu a mensagem, novamente um não atendimento de *deadline* é sinalizado (linha 11).

Posteriormente, a distância para falha dos dispositivos é atualizada baseada na situação de atendimento dos *deadlines* (variável “aten-

```

1  envia_mensagem;
2  //Evento sinalizado após transmissão da mensagem
3  evento(mensagem)
4  {
5      switch(mensagem)
6      {
7          case SUCESSO:
8              atendeu = 1;
9              break;
10         case FALHA_CSMA:
11         case SEM_ACK:
12             atendeu = 0;
13             break;
14         default:
15         }
16         distancia_falha = atualiza_distancia_falha(atendeu);
17         altera_prioridade(distancia_falha);
18     }

```

Figura 27: Algoritmo Distributed DBP (executado em cada nodo).

deu”) e a prioridade do dispositivo é atualizada. Caso um dispositivo se encontre próximo à ocorrência de uma falha dinâmica ( $d_j(k) = 1$ ) ou em uma falha dinâmica ( $d_j(k) = 0$ ), este recebe a maior prioridade do sistema. A Figura 24, apresentada no Capítulo 3, ilustra um diagrama de transição de estados para uma tarefa (2,3)-firm.

O *Point Coordinator Function* (PCF) é uma solução de acesso ao meio disponível no padrão IEEE 802.11 (802.11, 2007) para redução no número de colisões. Este mecanismo implementa uma solução centralizada para transmissões onde o *Point Coordinator* (PC) desempenha a função de um nodo mestre sincronizando a rede.

Uma emenda ao padrão (802.11, 2005) adicionou funções de coordenação chamada *Hybrid Coordination Function* (HFC), o qual é utilizada apenas para configuração de QoS da rede. O HFC oferece dois novos mecanismos: o *Enhanced Distributed Channel Access* (EDCA), o qual define níveis de prioridade entre usuários para diferenciação de tráfego e; o *HCF Controlled Channel Access* (HCCA), o qual permite a reserva de oportunidades de transmissões com o coordenador híbrido (HC).

Considerando que a especificação IEEE 802.15.4 não prevê qualquer tipo de priorização de mensagens. Um mecanismo de priorização de mensagens, baseado no padrão IEEE 802.11 (802.11, 2005) foi implementado através da alteração de parâmetros do protocolo de co-

municação CSMA/CA (Figura 18). Utilizou-se dois parâmetros para alterar dinamicamente a prioridade das mensagens:

1. **macMinBE**: valor relacionado com o expoente de *backoff* (BE). Este valor indica quantos períodos de *backoff* um dispositivo deve aguardar para tentar avaliar a atividade do canal de comunicação. Este valor está diretamente relacionado com o mecanismo de tentativa de evitar colisões (*Collision Avoidance* (CA) do protocolo). Por padrão este valor é iniciado com o valor mínimo de 3, caso exista extensão de bateria (*macBattLifExt*), BE é iniciado com 2. Se o valor  $\text{macMinBE} = 0$ , o mecanismo de detecção de ocupação do meio encontra-se desativado.
2. **macMaxCSMABackoffs**: corresponde ao número máximo de vezes que o algoritmo CSMA/CA entra em *backoff* enquanto tenta acessar o canal de comunicação. Este valor é inicialmente zero antes de cada nova transmissão. Caso o número máximo de tentativas de acessar o canal seja maior que *macMaxCSMABackoffs* uma mensagem de falha no canal é reportada às camadas superiores do protocolo de comunicação. Este valor é originalmente iniciado com 5.

Os valores utilizados para alterar esses parâmetros e os resultados obtidos pelos experimentos são apresentados na próxima seção.

### 5.3 RESULTADOS

O objetivo desta seção é avaliar o impacto do Distributed DBP frente à abordagem convencional oferecida pelo padrão IEEE 802.15.4, o qual não oferece nenhum mecanismo de diferenciação de serviços entre os dispositivos que formam a PAN. A abordagem convencional do IEEE 802.15.4 consiste simplesmente em envios periódicos de mensagens dos nodos para o coordenador, sem qualquer mecanismo de priorização de mensagens. Essa abordagem foi usada como uma abordagem *baseline*, a título de comparação.

Os estudos foram conduzidos através de análise experimental a partir de um cenário real. Utilizou-se o microkernel OpenZB<sup>1</sup> para avaliação da abordagem proposta neste trabalho. O OpenZB (CUNHA et al., 2007) implementa a pilha de protocolos especificada pelo padrão

---

<sup>1</sup><http://www.open-zb.net/>



IEEE 802.15.4, desenvolvido em NesC (GAY et al., 2003) sobre o sistema operacional TinyOS (HILL et al., 2000).

A configuração do cenário de experimentos foi composta por onze nodos MicaZ<sup>2</sup>. Um destes nodos foi escolhido para ser o coordenador e três outros foram configurados para enviar mensagens periodicamente para o coordenador, contendo restrições  $(m,k)$ -firm. Os sete outros nodos restantes foram programados para enviar mensagens continuamente, emulando tráfego de segundo plano, objetivando sobrecarregar o meio de acesso. Os três nodos com restrições  $(m,k)$ -firm receberam valores de  $m$  e  $k$  iguais a  $m = 2$  e  $k = 3$  (veja Figura 24). Estes nodos foram configurados para aguardar o recebimento de uma mensagem de confirmação (ACK) após o envio de uma mensagem.

Todos os nodos da rede enviam mensagens de 93 bytes – considerando um cabeçalho de 13 bytes. A estrutura do superquadro foi configurada utilizando os parâmetros  $BO = 5$  e  $SO = 4$  (então, com intervalos entre *beacons*  $BI = 491.5$  ms). Os nodos  $(m,k)$ -firm tiveram seus períodos fixados em 500 ms, o qual possuem quase o mesmo valor de BI. A carga externa imposta ao sistema foi implementada através do uso de nodos gerando mensagens com diferentes periodicidades.

A Tabela 6 e a Figura 28 apresentam a carga da rede em porcentagem, onde o valor de 100% representa o limite superior de utilização real da rede, assumindo uma taxa máxima de 130 kbps. Segundo (JENNIC, 2006) essa é a taxa real máxima que se consegue no IEEE 802.15.4 levando-se em consideração todos os *overheads* existentes. Nos experimentos, a carga mínima imposta ao sistema foi 3%. Esta carga foi obtida considerando-se apenas as mensagens provenientes dos nodos com restrições  $(m,k)$ -firm, sem interferência dos nodos de carga de fundo. As outras cargas submetidas ao sistema, 12%, 20%, 36%, 48%, 61%, 70%, 84% e 104% foram obtidas variando a periodicidade das mensagens geradas pelos nodos de carga de fundo 500ms, 250ms, 125ms, 90ms, 70ms, 60ms, 50ms e 40ms, respectivamente. Para cada ponto de carga da rede, ao menos 10 ensaios foram realizados, utilizando a abordagem Distributed DBP ou a abordagem oferecida pela especificação IEEE 802.15.4. Cada ensaio teve duração de cerca de 60 segundos.

A Tabela 6 apresenta o comportamento da abordagem padrão (sem prioridades, conforme especificado no padrão IEEE 802.15.4) frente a abordagem DDBP.

Na abordagem DDBP, para priorizar o tráfego de mensagens dos nodos com restrições  $(m,k)$ -firm, os parâmetros MacMaxCSMABackoffs e MacMinBE do protocolo CSMA/CA não são predeterminados

---

<sup>2</sup><http://www.xbow.com>

Carga da Rede	abordagem padrão IEEE		Distributed DBP	
	Perdas de <i>deadline</i>	Falhas	Perdas de <i>Deadline</i>	Falhas
3%	15.1%	5.9%	15.2%	4.2%
12%	32.0%	24.1%	29.5%	18.7%
20%	34.4%	26.4%	31.6%	19.9%
36%	36.5%	29.6%	31.9%	21.2%
48%	36.2%	29.2%	33.5%	24.3%
61%	37.6%	31.3%	35.3%	27.0%
70%	37.7%	31.9%	36.6%	28.6%
84%	39.6%	34.4%	38.6%	30.8%
104%	41.1%	35.9%	39.2%	32.8%

Tabela 6: Abordagem tradicional e abordagem DDBP.

estaticamente. Por padrão, estes parâmetros iniciam com os valores **MacMaxCSMABackoffs** = 5 e **MacMinBE** = 3, o qual nesse trabalho, representam um nodo com baixa prioridade. No entanto, quando a distância para falha em um nodo alcança o valor 1 (ou zero), este nodo está próximo (ou encontra-se) de uma falha dinâmica. Neste caso, estes parâmetros são dinamicamente alterados para: **MacMaxCSMABackoffs** = 6 permitindo que o dispositivo tenha mais chances de acesso ao canal de comunicação; e **MacMinBE** = 5 aumentando a probabilidade do dispositivo em encontrar o canal de comunicação livre, sem competir com os outros dispositivos da rede configurados com valores de MacMinBE padrão. Estes parâmetros são alterados dinamicamente, aumentando a taxa de sucesso na transmissão de uma mensagem, representando assim maiores prioridades para dispositivos em proximidade de ocorrência de falhas dinâmicas.

A Tabela 6 e a Figura 28 apresentam os resultados experimentais comparando a abordagem DDBP e abordagem padrão. Pode-se verificar que existe uma diminuição no número de ocorrências de falhas dinâmicas adotando-se a abordagem proposta. Durante situações de sobrecarga moderadas as ocorrências de falhas dinâmicas permaneceram próximas a 20%. Em condições de sobrecarga severa, as ocorrências de falhas dinâmicas permaneceram próximas a 30%.

Como conclusão dos experimentos, verifica-se que o funcionamento de uma rede IEEE 802.15.4 pode ser claramente melhorada com respeito a não ocorrência de falhas dinâmicas caso seja utilizado um algoritmo adequado para alocação de prioridades. O trabalho proposto mostra que algoritmos de escalonamento sofisticados podem ser im-

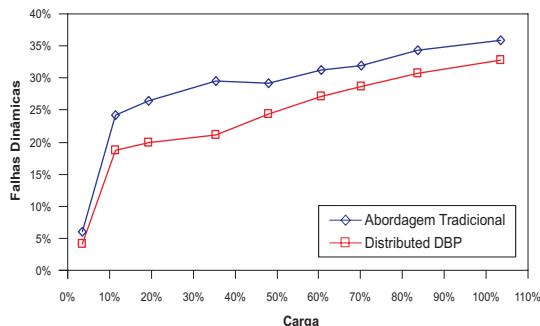


Figura 28: Falhas dinâmicas *vs.* utilização.

plementados de maneira eficaz sobre nodos com baixa capacidade de processamento, melhorando de forma significativa o funcionamento de uma rede típica no caso de sobrecarga e de não ocorrência de falhas dinâmicas.

Observou-se também que utilizando dispositivos sensores MicaZ executando o microkernel OpenZB, não é factível garantir níveis de desempenho e periodicidade especificados pelo padrão IEEE 802.15.4. Esta é uma limitação dos *clocks* do MicaZ conforme relatado em (CUNHA et al., 2008; SEVERINO, 2008).

Outro aspecto importante a ser ressaltado refere-se ao comportamento do coordenador quando submetido a condições severas de sobrecarga. Foi verificado que quando o número de mensagens endereçada ao coordenador aumenta, este leva muito tempo para gerar quadros de *beacon* para sincronização da rede, devido ao fato do coordenador precisar manipular todas as mensagens recebidas e criar mensagens de confirmação.

Durante a implementação do cenário experimental, algumas dificuldades foram observadas no que concerne o comportamento da plataforma de hardware e da implementação do protocolo CSMA/CA no OpenZB frente a priorização das mensagens. Sintonizar os parâmetros de priorização de mensagens no CSMA/CA mostrou-se ser uma tarefa bastante desafiadora. Conclui-se que com a atual implementação do OpenZB juntamente com a plataforma MicaZ a priorização das mensagens só pode ser realizada mediante uma calibração prévia nos parâmetros de comunicação (*macMinBE* e *macMaxCSMABackoffs*) levando em conta tanto o tamanho médio das mensagens da aplicação (e.g. mensagens curtas ou longas), como também o comportamento da

carga de fundo que gera interferências na comunicação da aplicação.

## 5.4 CONCLUSÕES DO CAPÍTULO

Este capítulo apresentou uma abordagem descentralizada de priorização de tráfego em RSSFs, objetivando reduzir as ocorrências de falhas dinâmicas. Resultados experimentais apresentam uma considerável redução no número de falhas dinâmicas nos dispositivos sensores usando o Distributed DBP em contraste com a abordagem tradicionalmente oferecida pela especificação IEEE 802.15.4. Ademais, a abordagem DDBP proposta necessita de alterações mínimas ao protocolo IEEE 802.15.4, garantindo compatibilidade com o padrão e com dispositivos COTS.

Observou-se que utilizando o microkernel OpenZB, executando sobre dispositivos MicaZ, não é possível garantir níveis de desempenho e tempo especificados no padrão IEEE 802.15.4. Outro importante aspecto que precisa ser ressaltado refere-se ao comportamento do Coordenador PAN quando submetido a condições de sobrecarga severas. Por exemplo, quando o número de mensagens endereçadas ao Coordenador PAN aumenta, este leva mais tempo que o esperado para gerar quadros de *beacons*, pois o Coordenador PAN precisa processar todas as mensagens recebidas e gerar as mensagens de reconhecimento (ACK) relativas à essas mensagens.

A abordagem proposta neste capítulo não é ótima, essencialmente devido ao fato de não haver uma visão global do sistema. Entretanto, esta abordagem descentralizada é mais escalável do que um abordagem centralizada. Adicionalmente, a utilização de uma abordagem descentralizada evita mensagens de sincronização entre o escalonador (provavelmente o Coordenador PAN) e outros dispositivos que compõem a PAN.

## 6 ESCALONAMENTO EM PERÍODOS DE ACESSO SEM CONTENÇÃO

O padrão IEEE 802.15.4 oferece uma solução de comunicação flexível para redes LR-WPAN. Quando ativo, o modo com *beacon* possibilita comunicação de tempo real através da alocação de GTSs. No entanto, este modo de transmissão suporta apenas sete dispositivos da rede utilizando este recurso em cada superquadro.

Este capítulo apresenta duas abordagens dinâmicas de escalonamento de mensagens periódicas durante períodos livres de contenção em redes IEEE 802.15.4. As abordagens propostas, baseadas no modelo de tarefas  $(m,k)$ -*firm*, garantem que ao menos  $m$  mensagens sejam alocadas em compartimentos de tempo garantido, para qualquer janela de  $k$  *deadlines* consecutivos. As soluções propostas oferecem garantias para as aplicações através da adoção de testes de escalonabilidade dinâmicos e algoritmos de priorização de mensagens *on-line*.

Este capítulo está organizado da seguinte forma: inicialmente o modelo do sistema é especificado. Posteriormente duas propostas de escalonamento de GTSs, durante períodos livres de contenção, são propostas. Por fim, algumas conclusões são apresentadas.

### 6.1 ABORDAGENS PROPOSTAS

Este trabalho, investiga características de tempo real do MAC no padrão IEEE 802.15.4. Uma solução promissora seria através da alocação de compartimentos garantidos (GTSs). Entretanto, esta abordagem mostra-se demasiadamente restritiva no que tange o número máximo de dispositivos que podem beneficiar-se deste mecanismo durante um superquadro (máximo de sete). Esta seção apresenta duas propostas de escalonamento de GTS durante períodos livres de contenção. A primeira abordagem, bastante restritiva com relação ao modelo de tarefas (ex. duração máxima das mensagens é de um compartimento de tempo,  $C_i=1$ ), é dirigida pelo algoritmo de escalonamento dinâmico Slotted DBP juntamente com um mecanismo de controle de admissão. A Segunda proposta, apresenta um teste de escalonabilidade dinâmico, onde tarefas admitem giros em seus padrões de classificação  $(m,k)$ -*firm*, em prol da melhoria na escalonabilidade do conjunto de tarefas.

### 6.1.1 SIGHT: explorando descartes de tarefas periódicas no escalonamento de GTSs

Nesta subseção, a política de alocação dinâmica de GTSs intitulada SIGHT (**S**kips **I**n **G**T**S** **s**cheduling for IEEE 802.15.4 **n**et**w**orks) é apresentada (SEMPREBOM et al., 2009c). A abordagem oferece garantias para aplicações com restrições  $(m,k)$ -firm através de uma abordagem centralizada executada por dois algoritmos, executados no nó coordenador (Coordenador PAN):

- um **teste de escalonabilidade determinístico**, executado quando o Coordenador PAN recebe uma solicitação de alocação de GTS
- uma **política de atribuição dinâmica de prioridades**, baseado no conceito de distância para falha (DBP), executado no início de cada superquadro.

Ambos algoritmos cooperam buscando aprimorar o atual mecanismo de alocação de GTSs para aplicações tempo real proposto pelo padrão IEEE 802.15.4.

Segundo o modelo de tarefas  $(m,k)$ -firm (HAMDAOUI; RAMANATHAN, 1995a), cada tarefa necessita que ao menos  $m$  ativações dentro de qualquer janela de  $k$  requisições consecutivas tenham seus *deadlines* atendidos; de outra forma a tarefa experimenta uma condição de falha dinâmica. Utilizando este modelo, um sistema pode então ser projetado para tolerar perdas de *deadlines*, assumindo que este número de falhas é limitado e precisamente espaçado.

Considerando uma rede IEEE 802.15.4, o modelo de tarefas  $(m,k)$ -firm pode ser utilizado para alocação de compartimentos no coordenador PAN. É possível considerar que uma tarefa perdeu um *deadline*, se esta não conseguiu transmitir sua mensagem no último superquadro. Desta forma, se existem mais do que sete solicitações de alocação de GTS no coordenador, e se cada requisição possui uma restrição  $(m,k)$ -firm associada, então o coordenador pode realizar um escalonamento destes compartimentos seguindo as restrições  $(m,k)$ -firm dos nós, buscando evitar a ocorrência de falhas dinâmicas. Além disto, em sistemas de tempo real que requerem garantias, análises de escalonabilidade podem ser executadas com a chegada de uma nova requisição para verificar a escalonabilidade do sistema, buscando prover garantias  $(m,k)$ -firm às aplicações. Uma nova requisição só deve ser aceita pelo

sistema se e somente se a aceitação desta não irá prejudicar a execução das outras requisições previamente admitidas no sistema.

No caso específico desta proposta, algumas importantes restrições devem ser considerados ao modelo de sistema proposto no Capítulo 4:

- $T_i = p$ : o período das tarefas coincide com o período do superquadro ( $p$ );
- $C_i = 1$ : número de compartimentos solicitados. Cada nodo é atendido em exatamente um compartimento de tempo<sup>1</sup>;
- $\forall i, T_i = T_j$  ( $j \neq i$ ): todas as tarefas possuem o mesmo período;
- $\forall i, k_i = k_j$  ( $j \neq i$ ): todas as tarefas possuem o mesmo valor  $k$  em suas restrições (m,k)-firm.

Estas restrições adicionadas ao modelo do sistema, permitem ao SIGHT o provimento de um teste de escalonabilidade exato (necessário e suficiente). Este é um pequeno preço a ser pago em detrimento das garantias temporais para além de sete tarefas alocando GTS em redes IEEE 802.15.4 e se o sistema necessita da execução de um teste de escalonabilidade exato. Ademais, o problema de se encontrar uma escala de execução viável considerando valores de  $k$ , períodos e tempo de computação arbitrários é um problema intratável computacionalmente (QUAN; HU, 2000; MOK; WANG, 2001).

#### 6.1.1.1 Algoritmo Slotted DBP

Juntamente com o modelo SIGHT, o algoritmo slotted DBP foi proposto, como uma aplicação direta do algoritmo DBP – originalmente proposto em (HAMDAOUI; RAMANATHAN, 1995a) – para um ambiente baseado em alocação de compartimentos em superquadros.

Abordagens baseadas no conceito de distância para falha (descrito em (HAMDAOUI; RAMANATHAN, 1995a)) não podem ser diretamente aplicadas sem considerar a estrutura do superquadro, pois estas abordagens tendem a inspecionar as janelas de compartimento contiguamente. Estas abordagens consideram que uma janela pode iniciar em um superquadro e terminar no próximo superquadro. Por essa razão, este trabalho propõe uma extensão ao conceito de distância para falha (DBP), inicialmente proposto por (HAMDAOUI; RAMANATHAN,

---

<sup>1</sup>A duração dos compartimentos pode ser ajustada de forma flexível pelo Coordenador PAN, através dos parâmetros BO e SO.

1995a), considerando a estrutura do superquadro no escalonamento de GTSs em redes IEEE 802.15.4.

O algoritmo Slotted DBP provê garantias  $(m,k)$ -firm para os dispositivos que solicitam alocação de GTSs em duas fases. Na primeira fase (Algoritmo 3), quando uma nova requisição de GTS é recebida pelo coordenador um teste de admissão é executado (Equação 6.1). Este teste dinâmico é realizado pelo coordenador PAN no momento da chegada de cada nova solicitação de alocação de GTS.

$$\sum_{i=1}^N \left( \frac{m_i C_i}{k_i T_i} \right) \leq 1.0, \quad (6.1)$$

onde  $N$  é o número total de nodos que solicitam alocação de GTS.

Se o teste de admissão falhar, por exemplo, se a carga do sistema (já considerando à tolerância das tarefas aos descartes) excede 100%, a solicitação de alocação de GTS é negada, em virtude da abordagem não poder acomodar mais uma nova solicitação de GTS. Porém, se a utilização do sistema for menor ou igual a 100%, uma nova requisição é aceita. Desta forma, a nova solicitação aceita é inserida na lista de requisições de GTS (lista  $L_i$ ) no coordenador PAN (Algoritmo 3).

---

**Algoritmo 3** Teste de admissão.

---

```

1: if a new GTS request  $i$  arrives then
2:   if  $\sum \left( \frac{m_i C_i}{k_i T_i} \right) \leq 1$  then
3:     insert request in list  $L_i$ 
4:     admission_test = feasible
5:   else
6:     admission_test = unfeasible
7:   end if
8: end if

```

---

Requisições não contempladas no CFP, não aceitas no teste de admissão, podem competir pelo acesso ao meio durante o período de acesso com contenção (CAP). Eventualmente, estas requisições conseguem realizar suas transmissões com sucesso (no Capítulo 5 foi proposta uma abordagem de provimento de QoS no envio de mensagens durante períodos de acesso com contenção).

A segunda fase da abordagem proposta (Algoritmo 4) é composta por um mecanismo de escalonamento dinâmico de GTS. Quando uma nova requisição é aceita pelo teste de admissão, o algoritmo de escalonamento elege dinamicamente qual dispositivo receberá acesso com



tempo garantido (GTS) no superframe atual. O critério de seleção é baseado na distância para falha de cada tarefa, a qual depende dos requisitos de QoS informados por cada dispositivo na forma de restrições  $(m,k)$ -*firm*.

---

**Algoritmo 4** Escalonamento dinâmico.

---

```

1: for all superframe period do
2:   for all GTS request  $j$  do
3:     calculate  $d_j(k)$ 
4:   end for
5:   sort  $L_i$  in  $d_j(k)$  order
6:   allocate first GTS requests in  $L_i$  on superframe empty slots
7: end for

```

---

Uma forma simples de se calcular a distância para falha foi proposta por (HAMDAOUI; RAMANATHAN, 1995a) (vide Subseção 3.1.1). Neste caso, é necessária a manutenção de um histórico de atendimento recente de cada tarefa periódica que envia mensagens. Este histórico de execução é uma  $k$ -tupla que armazena as  $k$  execuções mais recentes de uma tarefa. Seja 0 e 1 a representação para perda e atendimento de um *deadline*, respectivamente. Ao término de um superquadro um novo estado é produzido, o histórico é movido (da direita para esquerda) e o novo estado é adicionado na posição mais a direita. Por exemplo, uma tarefa com  $k = 3$  e histórico “110” representa que a execução mais recente da tarefa perdeu o *deadline*, enquanto que as outras duas execuções tiveram seus *deadlines* respeitados. A Figura 24 ilustra as distâncias para falha possíveis de uma tarefa  $(2,3)$ -*firm*.

Segundo (HAMDAOUI; RAMANATHAN, 1995a) uma tarefa  $i$ , é possível calcular a distância para falha  $d_i(k)$ : Seja  $met_i(n, h)$  a posição (a direita) referente ao  $n^{th}$  *deadline* atendido no histórico  $h$ . Se existem menos que  $n$  1s em  $h$ , então  $met_i(n, h) = k + 1$ . Por exemplo,  $met_i(1, “011”) = 1$ ,  $met_i(1, “010”) = 2$ ,  $met_i(2, “101”) = 3$ ,  $met_i(2, “001”) = 4$ . Desta forma, utilizando esta função, a distância para falha é denotada por:  $d_i(k) \leftarrow (k - met_i(m, h) + 1)$ .

### 6.1.1.2 Corretude do Slotted DBP

Objetivando demonstrar a corretude do algoritmo de escalonamento proposto, esta subseção apresenta uma demonstração informal da abordagem. A prova é realizada reduzindo o problema de escalonamento Slotted DBP para o algoritmo de escalonamento EDF. O

algoritmo de escalonamento EDF é ótimo no sentido que, se é possível produzir uma escala onde todos os *deadlines* são atendidos, tal escala pode ser produzida através do EDF.

As condições de  $C_i = 1$ ,  $D_i = T_i = p$  são impostas em virtude de o Slotted DBP tomar as decisões de escalonamento com a granularidade do superquadro. Isso permite ao Slotted DBP emular a natureza preemptiva do EDF.

Se o fator de utilização  $U > 1.0$ , então o conjunto de tarefas não pode ser escalonado por nenhum algoritmo ( $U \leq 1.0$ , é uma condição necessária).

Em uma escala EDF equivalente, deve-se garantir que  $n$  tarefas periódicas são atendidas por  $C_i$  unidades de tempo, em cada período  $k_i$ . Se  $\sum_{i=1}^N \left( \frac{C_i}{T_i} \right) \leq 1.0$  então o EDF garante que todas as tarefas terão seus *deadlines* atendidos. Este é um teste exato, ou seja, necessário e suficiente.

Com o Slotted DBP, exige-se uma escala viável para um fator de utilização mínimo  $\sum_{i=1}^N \left( \frac{m_i C_i}{k_i T_i} \right) \leq 1.0$ . Esta é a mesma utilização como a escala equivalente gerada pelo EDF. Quando o fator de utilização do Slotted DBP  $U = 1$  o algoritmo escala apenas as tarefas mandatórias; em outras palavras, a escala resultante irá atender apenas  $m_i$  *deadlines* em qualquer janela de restrições  $k_i$ .

Quando o fator de utilização é  $U < 1.0$  basta escalonar as tarefas mandatórias (distância para falha = 1) como EDF e atender as outras tarefas ( $d > 1$ ) nas folgas da utilização do sistema.

### 6.1.1.3 Exemplo de escalonamento Slotted DBP

Sem perda de generalidade para um melhor entendimento da proposta, esta subseção apresenta um exemplo de escalonamento para um superquadro com apenas dois GTSSs, formados por um compartimento de tempo cada, onde o CAP e o período inativo não são considerados.

No exemplo ilustrado pela Figura 29, existem quatro tarefas solicitando alocação de GTSSs. Em abordagens convencionais apenas os dois primeiros compartimentos tem sucesso na alocação. No entanto, estas quatro tarefas podem informar seus requisitos de QoS na forma de restrições (m,k)-firm, conforme segue:  $\tau_a = (4, 4)$ ;  $\tau_b = (2, 4)$ ;  $\tau_c = (1, 4)$ ;  $\tau_d = (1, 4)$ . Os outros parâmetros das tarefas são  $T_i = D_i = 2$  e  $C_i = 1$ , isto é, períodos e *deadlines* iguais ao período do superquadro; e tempo de execução igual a um compartimento de tempo.

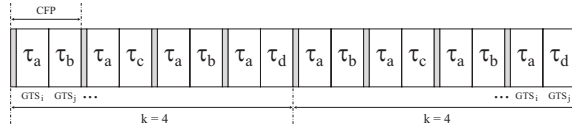


Figura 29: Escalonamento de GTS segundo Slotted DBP.

Segundo o Algoritmo 4, no primeiro superquadro, as distâncias para falha de todas as tarefas são computadas como segue:  $d_a(k) = 1$ ,  $d_b(k) = 3$ ,  $d_c(k) = d_d(k) = 4$ , e as tarefas  $\tau_a$  e  $\tau_b$  recebem o serviço garantido. No segundo superquadro:  $d_a(k) = 1$ ,  $d_b(k) = d_c(k) = d_d(k) = 3$ ; tarefas  $\tau_a$  e  $\tau_b$  ( $\tau_b$  é escolhida devido à ordem FIFO). No terceiro superquadro:  $d_a(k) = 1$ ,  $d_c(k) = d_d(k) = 2$ ,  $d_b(k) = 3$ ;  $\tau_a$  e  $\tau_c$  são alocadas. No quarto superquadro,  $d_a(k) = d_d(k) = 1$ ,  $d_b(k) = 2$ ,  $d_c(k) = 4$ ;  $\tau_a$  e  $\tau_d$  são alocadas.

Conseqüentemente, em abordagens tradicionais, se o resultado do teste de escalonabilidade  $\sum_{i=1}^N \frac{C_i}{T_i} \leq 1.0$ , todos os *deadlines* podem ser atendidos. No entanto, neste exemplo, tem-se  $\sum_{i=1}^N \frac{C_i}{T_i} = 2.0$ , desta forma, alguns *deadlines* seriam perdidos. Usando o algoritmo Slotted DBP, estas quatro solicitações de alocação de GTS podem ser atendidas, considerando suas restrições (m,k)-firm, conforme ilustrado na Figura 29. Além disso, esta figura apresenta dois intervalos  $k$ , é necessário verificar a escalonabilidade do sistema até o Mínimo Múltiplo Comum (MMC) dos valores de  $k$  das tarefas (neste caso, o valor de  $k$ ).

Por fim, importante ressaltar que mesmo que a ordem de chegada das tarefas fosse diferente, não haveria diferença para o SDBP, pois este ordena dinamicamente as tarefas conforme distância para falha. No entanto, esse comportamento seria diferente caso fosse considerado o Escalonamento Baseado em Janelas de West e Zhang (WEST; ZHANG, 2004), descrito na Subseção 3.1.4.

#### 6.1.1.4 Resultados de experimentos por simulação

Buscando investigar o comportamento do Slotted DBP em situações onde existem múltiplos nodos solicitando alocação de GTS, um conjunto de simulações foi realizado. Apesar do SDBP oferecer garantias (caso fosse considerada a inexistência de erros de comunicação), a simulação buscou também avaliar a abordagem em situações de sobrecarga, onde nem mesmo o SDBP consegue garantir as restrições (m,k)-firm dos nodos. A carga submetida ao sistema refere-se as execuções

das ativações mandatórias das tarefas, ou seja, atendimento de  $(m_i/k_i) \times C_i$  a cada intervalo  $T_i$ . Considerou-se que as requisições enviadas pelos dispositivos para o coordenador PAN contêm as restrições  $(m,k)$ -firm dos nodos. O desempenho do Slotted DBP foi comparado com:

- Escalonamento baseado em janelas (DWCS) (Subsecção 3.1.4), utilizado no escalonamento de tarefas que toleram descartes, em conformidade com parâmetros de descarte informados pelas tarefas na forma de janela de restrições.
- Escalonamento FIFO, utilizando atualmente pelo padrão IEEE 802.15.4, para alocação de GTSs durante períodos livres de contenção.

Utilizou-se um simulador<sup>2</sup> desenvolvido em linguagem C. As simulações foram realizadas considerando um conjunto de tarefas (nodos) solicitando compartimentos, variando entre 7 e 21 tarefas, assumindo que existem apenas sete compartimentos disponíveis em cada superquadro. Cada resultado obtido refere-se a uma média de 9.000 execuções, cada uma composta por dois mil períodos (2.000 superquadros).

Em cada cenário de simulação assumiu-se que todas as tarefas possuem os mesmos valores de  $k$ . Esta é uma condição importante previamente apresentada na Subsecção 6.1.1, necessária para que o SDBP ofereça garantias determinísticas às aplicações.

### *Falhas dinâmicas variando o fator de utilização*

Nos primeiros experimentos, a carga (fator de utilização) foi escolhida para cada conjunto de experimentos, enquanto o número de tarefas,  $m$  e  $k$  foram aleatoriamente gerados. Os valores de  $k$  variam de 1 a 10, e os valores de  $m$  variam de 1 até  $k$ . Os valores dos fatores de utilização foram calculados de acordo com a Inequação 6.1.

A Figura 30 apresenta os valores de falhas dinâmicas *vs.* fator de utilização (eixo  $x$  na Figura 30) e a Tabela 7 apresenta maiores informações, como perdas de *deadlines*. Os valores apresentados na última coluna na Tabela 7 referem-se a média do número de nodos gerados na simulação.

A Figura 30 e a Tabela 7 claramente mostram que ambas as abordagens SDBP e DWCS não falham quando a carga é menor ou igual a 1.0 (fator de utilização  $\leq 100\%$ ), mesmo quando o número de

<sup>2</sup>Disponível em: [http://www.das.ufsc.br/~tisemp/sight\\_simul/SIGHT.C](http://www.das.ufsc.br/~tisemp/sight_simul/SIGHT.C).

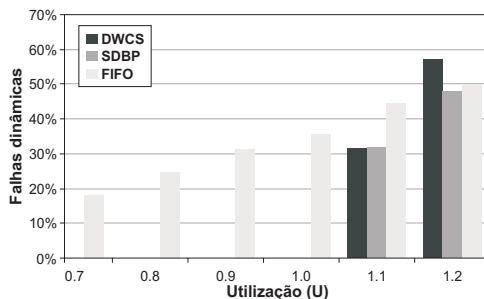


Figura 30: Falhas dinâmicas *vs.* fator de utilização.

requisições aceitas<sup>3</sup> é maior que sete. Observou-se, também, que ambas as abordagens SDBP e DWCS são inadequadas em situações de sobrecarga onde a carga (fator de utilização) é maior que 1.0. Além disso, sobre condições de sobrecarga severas, as falhas dinâmicas na abordagem DWCS tendem a apresentar uma grande degradação. A abordagem FIFO é completamente inadequada, mesmo em situações onde a carga do sistema é baixa.

Carga	Falhas dinâmicas			<i>Deadlines</i> perdidos	Média de nodos
	SDBP	DWCS	FIFO		
0.7	0%	0%	17.9%	24.9%	9.3
0.8	0%	0%	24.5%	29.5%	9.9
0.9	0%	0%	31.3%	35.3%	10.8
1.0	0%	0%	35.8%	38.2%	11.3
1.1	31.8%	31.4%	44.5%	46.6%	13.1
1.2	48.0%	57.2%	49.7%	51.5%	14.4
1.3	56.3%	81.1%	56.4%	57.0%	16.3

Tabela 7: Falhas Dinâmicas e perdas de *deadlines* em diferentes fatores de utilização.

### *Falhas dinâmicas variando o número de nodos*

Nos experimento à seguir, o número de nodos (tarefas) foram selecionados para cada simulação, mas a carga, e os parâmetros  $m$  e  $k$  de

<sup>3</sup>Importante ressaltar que a carga refere-se a execução das ativações mandatórias do conjunto de tarefas.

cada tarefa foram gerados aleatoriamente, 9.000 vezes. Em cada experimento apenas o conjunto de tarefas com fator de utilização abaixo de 1.0 foi utilizado, de acordo com o teste de admissão apresentado no Algoritmo 3. Conseqüentemente, SDBP não deve apresentar a ocorrência de falhas dinâmicas.

Nodos	Falhas dinâmicas		Perdas de <i>deadlines</i>	Média de carga
	SDBP / DWCS	FIFO		
7	0%	0%	0%	0.64
8	0%	12.4%	12.5%	0.69
9	0%	21.3%	22.2%	0.76
10	0%	27.1%	30.0%	0.82
11	0%	30.7%	36.4%	0.86
12	0%	32.5%	41.7%	0.89
13	0%	33.2%	46.2%	0.91
14	0%	33.4%	50.0%	0.92

Tabela 8: Falhas dinâmicas considerando um número fixo de nodos.

A Tabela 8 resume os resultados destacando as falhas dinâmicas em função da média do número de tarefas admitidas pelo sistema. Pode ser observado que ambas abordagens SDBP e DWCS podem aceitar simultaneamente mais do que sete requisições sem que falhas dinâmicas ocorram. Mesmo quando a taxa de perdas de *deadlines* alcança 50% ou mais, não existem falhas dinâmicas, desde que a carga no sistema seja inferior a 100%.

Todos os resultados acima indicam que ambas abordagens SDBP e DWCS não devem ser utilizadas em condições de sobrecarga, na ausência de um teste de admissão determinístico, especialmente quando a carga no sistema é superior a 1.0, os resultados demonstraram que a abordagem Slotted SBP é levemente melhor do que a abordagem FIFO.

### 6.1.2 (m,k)-*spin*: alocação de GTs com giro no padrão de classificação (m,k)-*firm*

A abordagem apresentada anteriormente assume um modelo de sistema bastante restritivo, onde os valores de  $k$ , períodos e tempo de computação das tarefas são iguais. Por conseguinte, novas abordagens foram desenvolvidas no sentido de relaxar as restrições impostas ao modelo de tarefas.

Nesta subseção, primeiramente é apresentado um algoritmo de

classificação de ativações de tarefas, baseado no trabalho apresentado em (RAMANATHAN, 1999). Posteriormente, apresenta-se um teste de admissão para avaliar a admissibilidade das tarefas em consonância com suas garantias  $(m,k)$ -firm e finalmente, um algoritmo de escalonamento, onde ativações de tarefas são classificadas como mandatórias ou opcionais, sendo as ativações escalonadas durante períodos livres de contenção.

### 6.1.2.1 Classificador de tarefas

Em (RAMANATHAN, 1999), os autores apresentaram um classificador de instâncias de tarefas com restrições  $(m,k)$ -firm, onde um algoritmo determina dinamicamente se uma instância é mandatória (1) ou opcional (0). Ativações classificadas como mandatórias, recebem prioridades, de tal forma que seus *deadlines* são garantidamente atendidos. Por outro lado, instâncias de tarefas classificadas como opcionais recebem prioridades mais baixas e não possuem garantias no atendimento de seus *deadlines*.

O padrão de classificação de uma tarefa pode ser concebido como um anel. A cada nova ativação da tarefa, um giro no sentido horário é realizado no anel, indicando se a instância da tarefa é mandatória ou opcional. A Figura 31 ilustra a formação do padrão de classificação para uma tarefa  $(2,3)$ -firm.

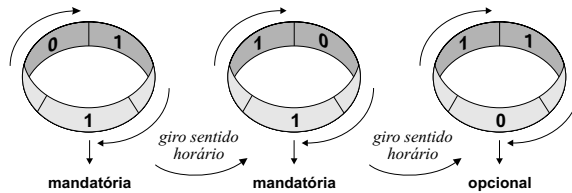


Figura 31: Padrão de classificação inicial para uma tarefa  $(2,3)$ -firm.

Apesar da simplicidade do algoritmo proposto em (RAMANATHAN, 1999), este apresenta algumas limitações. A primeira ativação de cada tarefa é sempre classificada como mandatória, o que implica que o pior caso de tempo de execução de qualquer tarefa do conjunto, ocorre sempre durante seu primeiro período. Isto significa, que o classificador adotado não está apto a considerar qualquer descarte de ativação durante o primeiro período das tarefas sem a ocorrência de falhas. Portanto, o comportamento do classificador proposto em (RAMANA-

THAN, 1999) insere um pessimismo significante na análise de escalonabilidade. Este fenômeno pode ser visualizado na Figura 32, onde um conjunto reduzido, composto por apenas duas tarefas  $\tau_1$  e  $\tau_2$  com  $C_1 = C_2 = 1, T_1 = T_2 = 1, m_1 = m_2 = 1, k_1 = k_2 = 2$  não pode ser escalonado de acordo com o teste proposto por (RAMANATHAN, 1999) (Figura 32(a)). Não obstante, uma simples alteração no padrão de classificação em  $\tau_2$  possibilitaria geração uma escala viável (Figura 32(b)).

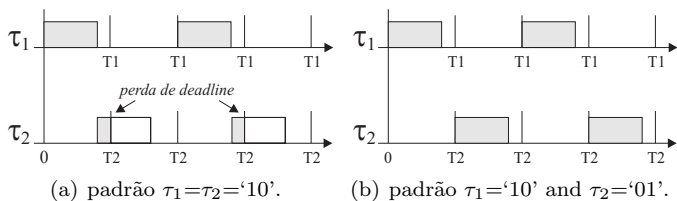


Figura 32: Classificador de tarefas (m,k)-firm.

De acordo com (QUAN; HU, 2000), a classificação ótima de ativações em mandatória e opcional é um problema que não pode ser tratado em tempo polinomial (NP-difícil), considerando tempos de computação, períodos e valores de janela  $k$  arbitrários. Neste trabalho, propõe-se uma variação no classificador de tarefas introduzido em (RAMANATHAN, 1999), onde giros (*spins*) no padrão de classificação (m,k)-firm original são admitidos.

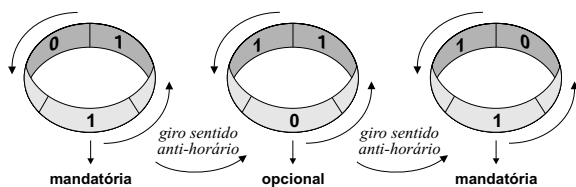


Figura 33: Padrões gerados a partir de rotações em uma tarefa (2,3)-firm.

Na abordagem proposta, um padrão (m,k)-firm pode ser alterado através de um giro para o lado esquerdo (operação *left-spin*). Por exemplo, de acordo com (RAMANATHAN, 1999), uma tarefa com restrições (2,3)-firm produz um padrão de classificação (m,k)-firm “110”, onde as duas primeiras ativações da tarefa são classificadas como mandatória e a última instância subsequente é classificadas como opcional.



Se um giro adicional para o lado esquerdo é admitido antes da primeira execução da tarefa, um novo padrão de classificação pode ser considerado, limitado por  $k-1$  giros. As novas possibilidades para os padrões de classificação  $(m,k)$ -firm de uma tarefa  $(2,3)$ -firm são ilustrados na Figura 33: um giro para esquerda (sentido anti-horário) produz o padrão  $(m,k)$ -firm = “011” e dois giros para esquerda produz um padrão de classificação  $(m,k)$ -firm = “101”.

A classificação das instâncias em mandatória ou opcional é baseada nos valores  $m_i$  e  $k_i$  de cada tarefa. Portanto, a instância de uma tarefa  $\tau_i$  ativada no instante  $w.T_i$  é classificada como mandatória se a Equação 6.2 é satisfeita, ou classificada como opcional, caso contrário.

$$w = \left\lceil \left\lfloor \frac{wm_i}{k_i} \right\rfloor \times \frac{k_i}{m_i} \right\rceil, \quad \text{para } w = \left\lfloor \frac{t}{T_i} + s_i \right\rfloor \quad (6.2)$$

O termo  $t$  refere-se ao tempo de chegada testado. O termo  $s_i$  representa o valor de giro considerado para a tarefa  $\tau_i$  (limitado a  $k_i - 1$ ). Este valor é igual a zero quando o valor de giro no padrão de classificação não é implementado. Por exemplo, considere uma aplicação de controle via rede composta por três tarefas periódicas, com os seguintes parâmetros:

$$\begin{aligned} \tau_1: C_1 = 2 \quad T_1 = 2 \quad m_1 = 7 \quad k_1 = 9 \quad s_1 = 0, \\ \tau_2: C_2 = 1 \quad T_2 = 9 \quad m_2 = 1 \quad k_2 = 2 \quad s_2 = 0, \\ \tau_3: C_3 = 2 \quad T_3 = 6 \quad m_3 = 1 \quad k_3 = 3 \quad s_3 = 1. \end{aligned}$$

Segundo a Equação 6.2 a tarefa  $\tau_i$  possui um padrão de classificação  $(m,k)$ -firm “111101110”, o que significa que as instâncias com tempos de chegada 0,2,4,6,10,12,14,... são classificadas como mandatórias, enquanto que instâncias com tempos de chegada 8 e 16 são classificadas como opcionais. A tarefa  $\tau_2$  possui “10” como padrão de classificação, implicando que instâncias com tempos de chegada 0,18,36,... são classificadas como mandatórias e instâncias com tempo de chegada 9,27,45,... são classificadas como opcionais. Para a tarefa  $\tau_3$ , um giro no padrão de classificação  $(m,k)$ -firm se faz necessário ( $s_3 = 1$ ), resultando em um padrão de classificação “001”. Considerando este giro no padrão, a tarefa  $\tau_3$  possui ativações com tempos de chegada 12,30,48... classificadas como mandatórias, por outro lado, aquelas com tempos de ativação 0,6,18,24,36,42... são classificadas como opcionais. A Subseção 6.1.2.4 demonstra que este conjunto de tarefas só é viável em virtude de  $\tau_3$  ter sofrido um giro em seu padrão de classificação.

### 6.1.2.2 Análise de escalonabilidade

Existem vários trabalhos na literatura que adotam esquemas de atribuição dinâmica de prioridades para lidar com abordagens de escalonamento  $(m,k)$ -*firm* (BERNAT; BURNS, 1997; GOOSSENS, 2008; HAMD AOUI; RAMANATHAN, 1995a; QUAN; HU, 2000). No entanto, muitas vezes ocorre que escalonamento por prioridades fixas pode ser mais atrativo quando comparado ao escalonamento por prioridade dinâmica, no que concerne sua simplicidade de implementação e baixo sobrecusto (*overhead*). Isto é particularmente, mais evidente nos casos de redes de sensores sem fios, devido às inerentes limitações em poder de processamento e consumo energético, que devem ser consideradas quando se projeta aplicações sobre redes de sensores. Portanto, neste trabalho, adota-se um algoritmo de escalonamento por prioridade fixa para o modelo tarefas  $(m,k)$ -*firm*.

O teste de escalonabilidade proposto toma decisões de escalonamento em tempo de execução (*on-line*). Portanto, sempre que existe uma solicitação de admissão de uma nova tarefa no sistema, se faz necessária a realização de um teste de escalonabilidade (Algoritmo 5) considerando a execução desta nova tarefa em conjunto com as tarefas já aceitas no sistema. A admissão da nova tarefa ocorrerá apenas se sua admissão não prejudicar as garantias oferecidas as tarefas já previamente aceitas.

O teste de escalonabilidade desenvolvido nessa proposta baseia-se no conceito de período de ocupação (*busy period*) (AUDSLEY et al., 1993; TINDELL et al., 1993). Considere o tempo de liberação de uma tarefa  $\tau_i$ . A partir deste ponto, até o término de execução da tarefa  $\tau_i$ , o processador estará executando processos com prioridade  $i$  ou maior (tarefas  $\tau_j$ ). O processador é dito estar executando um período de ocupação de nível  $i$ . Para uma dada tarefa  $\tau_i$  o cálculo do período de ocupação é realizado, considerando a execução das tarefas  $\tau_j$  ( $j < i$ ). O cálculo inicia-se no instante de chegada de  $\tau_i$  e é executado de forma interativa até o valor de período de ocupação calculado convergir ou exceder o *deadline* de  $\tau_i$ . Em outras palavras, a interação inicia com  $W_i(0) = C_i$  e termina quando  $W_i(t + 1) = W_i(t)$ . Se a interação converge e se  $W_i(t) \leq D_i$  para todas as ativações mandatórias que foram testadas, o conjunto de tarefas é dito ser viável (escalonável). Caso contrário, se  $W_i(t) > D_i$ , o conjunto de tarefas é dito ser inviável (não escalonável).

$$W_i(0) = C_i \quad (6.3)$$

$$W_i(t+1) = (C_i - \Delta_i) + \sum_{j=1}^{i-1} \left[ \frac{W_i(t) + (\phi_i - \phi_j) \cdot \frac{m_j}{k_j}}{T_j} \right] C_j \quad (6.4)$$

A Equação 6.4 apresenta o cálculo de período de ocupação, considerando tarefas com restrições (m,k)-firm, onde:

$W_i(t)$  : representa o período de ocupação de nível  $i$  calculado no instante de tempo  $t$ ,

$\Delta_i$  : representa o *offset* entre  $\phi_i$  e  $\phi_j$ , isto é, a diferença entre o tempo de chegada de  $\tau_i$  e o tempo de chegada de  $\tau_j$ , que afetam o período testado,

$\phi_i$  : representa o tempo de chegada da tarefa  $\tau_i$ ,

$\phi_j$  : representa o tempo de chegada da tarefa  $\tau_j$ .

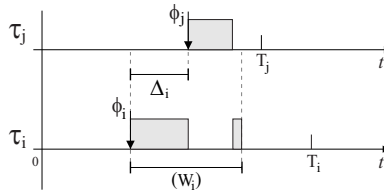


Figura 34: Cálculo do período de ocupação de nível  $i$ .

A Equação 6.4 calcula o período de ocupação de nível  $i$  considerando a execução de todas as ativações mandatórias das tarefas de maior prioridade testadas durante cada intervalo de tempo (linha 21, Algoritmo 5). Com relação aos tempos de interferência considerados pelo teste de escalonabilidade proposto, é necessário iniciar a interação no instante de tempo o qual as tarefas de maior prioridade efetivamente começam a gerar interferência. Este tempo de início é calculado (linhas 5 e 6, Algoritmo 5) e pode ser visualizado na Figura 34, referenciado como  $\Delta_i$ .

Para obter o valor período de ocupação de nível  $i$ , a Equação 6.4 é resolvida iterativamente e sua execução é concluída quando:  $W_i(t+1) = W_i(t)$  ou  $W_i(t) > D_i$ . Com a verificação desta condição tem-se a convergência da interação (linha 7, Algoritmo 5).

---

**Algoritmo 5** Análise de escalonabilidade
 

---

```

1: procedure Scheduler
2: for  $i = 0 \rightarrow N$  do
3:   for  $j = 0 \rightarrow \text{task}[i].\text{arrival.length}$  do
4:      $W_i \leftarrow \text{task}[i].C$ 
5:      $\text{start\_time} \leftarrow \text{GetStartTime}$ 
6:      $\Delta_i \leftarrow \text{task}[i].\text{arrival}[j] - \text{start\_time}$ 
7:     while  $W_i \leq \text{task}[i].\text{period}$  and  $W_i$  don't converge do
8:        $W_i \leftarrow (\text{task}[i].C - \Delta_i) + \text{hpDemand}$ 
9:     end while
10:    if  $W_i > \text{task}[i].\text{period}$  or  $W_i$  don't converge then
11:      return unfeasible
12:    end if
13:  end for
14: end for
15: return feasible
16: end procedure

17: procedure hpDemand
18: for  $j = 0 \rightarrow j < i$  do
19:   activations  $\leftarrow \left\lceil \frac{W_i + (\Phi_i - \Phi_j)}{\text{task}[i].\text{period}} \right\rceil$ 
20:   sum  $\leftarrow (\text{activations} - \text{task}[j].\text{optional}) \times \text{task}[j].C$ 
21: end for
22: return sum
23: end procedure

```

---

Uma implementação direta do algoritmo de escalonamento proposto tem custo computacional de  $O(s(N^2.MMC^2))$ , onde  $s$  representa o número de giros considerados em cada tarefa,  $N$  representa o número de tarefas no conjunto e  $MMC$  representa o mínimo múltiplo comum para  $MMC[(k_i.T_i), (k_j.T_j)]$ .

### 6.1.2.3 Prova formal do teste de escalonabilidade

Esta seção objetiva provar o teste de escalonabilidade proposto neste trabalho. Por conveniência, a prova é dividida em três partes no qual objetiva-se demonstrar que (i) se todas as instâncias de uma tarefa classificadas como mandatórias tiverem seus *deadlines* atendidos, então as restrições (m,k)-firm da tarefa são satisfeitas; (ii) para cada instante de tempo  $t$  testado, caso  $t$  seja o instante de tempo da chegada de uma tarefa mandatória, então a Equação 6.4 é válida; e finalmente (iii) a

escala de execução das tarefas repete-se após  $\text{MMC}[(k_i.T_i), (k_j.T_j)]$ .

**Parte (i):** os Lemas 1–3 apresentados a seguir seguem a mesma estrutura dos propostos originalmente em (RAMANATHAN, 1999), considerando agora a existência de giros nos padrões de classificação. Esses lemas serão utilizados para a descrição do Teorema 1, referente a primeira parte da demonstração.

**Lema 1** *Para cada tarefa  $\tau_i$ , a ativação periódica no instante  $wT_i$ ,  $w \in Z^+$ , é classificada como mandatária se e somente se existe um  $l$ , inteiro não negativo tal que  $w = \lfloor l.(k_i/m_i) \rfloor$ .*

**Prova** Suponha que a instância de  $\tau_i$  ativada no tempo  $wT_i$  é classificada como mandatária. Desta forma, segundo o classificador:

$$w = \left\lfloor \left[ \frac{wm_i}{k_i} \right] \times \frac{k_i}{m_i} \right\rfloor, \quad \text{para } w = \left\lfloor \frac{t}{T_i} + s_i \right\rfloor.$$

Assumindo  $\alpha_i = k_i/m_i$ .  $w = \lfloor l.\alpha_i \rfloor$ , e pelas propriedades da função chão tem-se que:  $l.\alpha_i + 1 < w \leq l.\alpha_i$ . Do lado esquerdo da inequação, tem-se  $l < (w + 1)/\alpha_i$ . Como, por definição,  $\alpha \geq 1$ , então  $(w + 1)/\alpha_i < (w/\alpha_i) + 1$ . Desta forma, pode-se reescrever a inequação acima da seguinte maneira  $w/\alpha_i \leq l < (w/\alpha_i) + 1$ .

Como  $l$  é um número inteiro, a inequação acima pode ser representada através da função teto:  $l = \lceil w/\alpha_i \rceil$ .

**Lema 2**  $\forall i \mid 1 \leq i \leq N$ , o algoritmo proposto classifica ao menos  $m_i$  das primeiras  $k_i$  instâncias de  $\tau_i$  como mandatárias.

**Prova** Segundo o Lema 1, as instâncias de  $\tau_i$  são classificadas como mandatárias se e somente se seus instantes de ativação ocorrem em  $\lfloor l.(k_i/m_i) \rfloor \times T_i$  para algum  $l \in Z^+$ . Entre as primeiras  $k_i$  instâncias de  $\tau_i$ , ao menos as instâncias com tempos de ativação pertencentes ao conjunto  $\lfloor l.(k_i/m_i) \rfloor T_i : 0 \leq l \leq m_i - 1$  são classificadas como mandatárias. Em outras palavras, de 0 até  $m_i - 1$  existem  $m$  valores  $l$ . Desta forma, como  $k_i \geq m_i$ , existem exatamente  $m_i$  elementos neste conjunto.

**Lema 3**  $\forall i \mid 1 \leq i \leq N$ , o algoritmo proposto classifica as instâncias de  $\tau_i$  ativadas no instante  $(wT_i + k_iT_i)$ ,  $w \in Z^+$ , como mandatária se e somente se a instância de  $\tau_i$  ativada no instante  $wT_i$  também é classificada como mandatária.

**Prova** Se a instância ativada no tempo  $(wT_i + k_iT_i)$  é classificada como mandatária, então, conforme demonstrado no Lema 1, existe um  $l \in Z^+$  tal que  $\lfloor l.(k_i/m_i) \rfloor = w + k_i$ . Como  $w + k_i \geq k_i$ , sabe-se que  $l \geq m_i$ . Ademais,  $\lfloor (l - m_i)k_i/m_i \rfloor = \lfloor l.(k_i/m_i) \rfloor - k_i = w$ . Então, existe um inteiro não negativo  $l' = l - m_i$ , tal que  $\lfloor (l'.k_i)/m_i \rfloor = w$ . Portanto, a instância ativada no instante  $wT_i$  também é classificada como mandatária. Se a instância ativada no instante  $wT_i$  é classificada como mandatária, então existe um  $l \in Z^+$  tal que  $\lfloor (l.k_i)/m_i \rfloor = w$ . Assim,  $\lfloor (l + m_i)k_i/m_i \rfloor = \lfloor (l.k_i)/m_i \rfloor + k_i = w + k_i$ . Ou seja, existe um inteiro não negativo  $l' = l + m_i$  tal que  $\lfloor (l'.k_i)/m_i \rfloor = w + k_i$ . Portanto, a instância ativada no instante  $(wT_i + k_iT_i)$  também é classificada como mandatária.

A prova do Teorema 1 considera que o padrão de classificação  $(m,k)$ -firm das tarefas admite giros para o lado esquerdo.

**Teorema 1** *se todas as instâncias de  $\tau_i$ , classificadas como mandatárias tiverem seus deadlines atendidos, então as restrições  $(m,k)$ -firm de  $\tau_i$  são satisfeitas.*

**Prova** O Lema 2, considera a classificação das instâncias mandatárias da primeira janela de tamanho  $k_i$  da tarefa  $\tau_i$ . No Lema 3, o teorema continua válido, considerando a classificação das ativações mandatárias com periodicidade  $(wT_i + k_iP_i)$ . Desta forma, para qualquer janela de  $k_i$  instâncias consecutivas de  $\tau_i$ , ao menos  $m_i$  instâncias são classificadas como mandatárias. Portanto, se todas as ativações classificadas como mandatárias, tiverem seus *deadlines* atendidos, então as restrições  $(m,k)$ -firm de  $\tau_i$  são satisfeitos.

**Parte (ii):** O Teorema 2 refere-se a segunda parte da demonstração do teste de escalonabilidade proposto.

**Teorema 2** *Para cada instante de tempo  $t$  testado, se  $t$  é o tempo de chegada de uma ativação mandatária da tarefa  $\tau_i$ , então a Equação 6.4 é válida.*

**Prova** Considerando a Equação 6.4, as ativações mandatárias de  $\tau_j$  possuem maior prioridade do que as instâncias mandatárias de  $\tau_i$ . O segundo termo da Equação 6.4, representa o somatório de todas as ativações mandatárias de  $\tau_j$  que foram ativadas até o tempo  $t$ .  $W_i(t)$  é o somatório do tempo de computação de  $\tau_i$  somados com os tempos de computação de todas as ativações mandatárias de  $\tau_j$ , mais prioritária que  $\tau_i$ , que ocorreram até o tempo  $t$ . Desta forma, se  $W_i(t+1) = W_i(t)$

e  $W_i(t) \leq D_i$ , então o *deadline* de  $\tau_i$ , para o instante de tempo  $t$  foi atendido.

Para o algoritmo  $(m,k)$ -*spin* – onde os giros no padrão de classificação ocorrem apenas na última tarefa do conjunto – o teorema permanece válido, pois apenas a última tarefa ( $\tau_i$ , menos prioritária) do conjunto sofrerá alterações no padrão de marcação, não interferindo assim nas demais tarefas  $\tau_j$ , de maior prioridade. Entretanto, deve-se verificar se a Equação 6.4 continua válida se tarefas  $\tau_j$ , de maior prioridade forem giradas.

Considerando qualquer instante de tempo  $t$ , dentro do intervalo  $(k_i T_i)$  de uma tarefa  $\tau_i$  ( $j < i$ ), a quantidade de ativações mandatórias de uma tarefa  $\tau_j$  que já sofreram alterações no padrão de marcação, nunca será maior do que o número de ativações mandatórias obtidas quando utilizado o padrão de marcação original, proposto por (RAMANATHAN, 1999) (*0-spins*). A Figura 35 ilustra esse cenário. Para qualquer tempo  $t$  de  $\tau_j$  testado, a quantidade de ativações mandatórias computadas até o instante  $t$  testado, nunca será maior do que o número de ativações mandatórias computados pelo padrão *0-spin*. Desta forma, o algoritmo  $(m,k)$ -*spin* permanece válido.

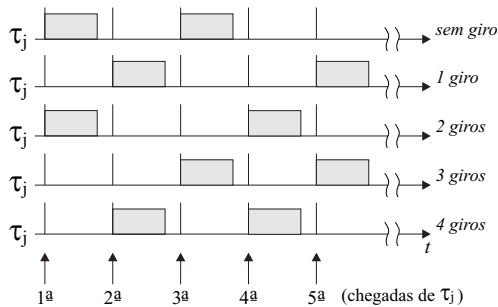


Figura 35: Tempos de chegada da tarefa  $\tau_j$ , com restrições  $(2,5)$ -*firm*.

Note que a prova do Teorema 2 segue a mesma estrutura das provas encontradas no trabalho (LEHOCZKY, 1990), onde um teste de escalonabilidade exato é proposto e (RAMANATHAN, 1999), onde um teste suficiente é derivado.

**Parte (iii):** O Teorema 3, apresenta a terceira etapa da demonstração do teste de escalonabilidade proposto.

**Teorema 3** *A escalonabilidade do sistema deve ser verificada até o valor de  $MMC[(k_i.T_i), (k_j.T_j)]$ .*

**Prova** O trabalho (LIU; LAYLAND, 1973) apresenta a definição de instante crítico para um nível de prioridade  $i$  como sendo o instante o qual ativações de nível  $i$  são liberadas juntamente com todas as ativações de maior prioridade. O classificador de tarefas apresentado por (RAMANATHAN, 1999), força com que todas as ativações mandatórias de  $\tau_i$  sejam liberadas no início de seus períodos. Desta forma, o pior caso de carga no sistema ocorre no instante  $t = 0$ .

A proposta apresentada nesse artigo considera que os padrões de marcação das tarefas podem ser girados. Desta forma, o pior caso de carga gerado pelo conjunto de tarefas não ocorre necessariamente em  $t = 0$ .

Em (JOSEPH; PANDYA, 1986) é provado que para os casos onde todas as ativações são liberadas simultaneamente em  $t = 0$ , o pior caso de carga se repete em múltiplos de  $H_i = MMC(T_i \mid 1 \leq j \leq i)$ . Ou seja o mínimo múltiplo comum de todos os valores de  $T_1$  até  $T_i$ . Assumindo que os padrões de marcação das tarefas (m,k)-firm se repetem com periodicidade  $(k_i T_i)$ , de forma análoga o pior caso de carga se repete em múltiplos de  $H_i = MMC(k_i T_i \mid 1 \leq j \leq i)$ . Desta forma, verifica-se os términos dos períodos de ocupação da tarefa  $\tau_i$  a partir dos instantes de chegada  $t$  até  $H_i$ .

#### 6.1.2.4 Exemplo de aplicação

A Figura 36 ilustra a escala de execução para o exemplo apresentado na Subseção 6.1.2.1. A Figura 36(a) apresenta a escala de execução assumindo que nenhuma tarefa do conjunto realiza giros em seus padrões de marcação (m,k)-firm. Desta forma, a tarefa  $\tau_3$  tem um *deadline* perdido em  $t = 6$ . Como consequência deste evento, o conjunto de tarefas é dito inviável. Caso contrário, se giros no padrão de classificação (m,k)-firm são considerados, novas escalas de execução viáveis podem ser viabilizadas.

A Figura 36(b) mostra a escala de execução, quando um giro (para esquerda) no padrão de classificação (m,k)-firm da tarefa  $\tau_3$  é considerado. Este giro provoca novos tempos de chegada para  $\tau_3$ , produzindo uma escala de execução factível para o conjunto de tarefas.

Observe que, o conjunto de tarefas só pode ser considerado escalonável devido ao fato da tarefa  $\tau_3$  ter recebido um giro em seu padrão de classificação.



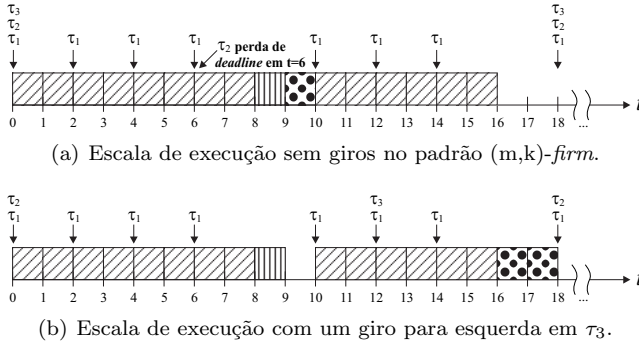


Figura 36: Escala de execução (m,k)-spin.

### 6.1.2.5 Resultados

A avaliação da abordagem (m,k)-spin proposta foi conduzida em duas etapas: (i) por simulação e (ii) através de um cenário experimental utilizando dispositivos sensores compatíveis com o padrão IEEE 802.15.4. O objetivo desta subseção é avaliar o desempenho da abordagem proposta neste trabalho frente a abordagem originalmente proposta por (RAMANATHAN, 1999).

#### Resultados de simulação

Por uma questão de conveniência, adotou-se o nome *Ramanathan* para o algoritmo apresentado em (RAMANATHAN, 1999) e (m,k)-spin para o algoritmo proposto neste trabalho (Algoritmo 5). No que diz respeito à abordagem (m,k)-spin, duas propostas foram avaliadas: (i) (m,k)-spin: os giros no padrão (m,k)-firm são aplicados apenas na última tarefa do conjunto; (ii) (m,k)-rec-spin: os giros nos padrões de classificação podem ser aplicados em qualquer tarefa do conjunto. A proposta (m,k)-rec-spin é uma versão recursiva do algoritmo (m,k)-spin e seu nome advém do fato que o teste de escalonabilidade considera giros para tarefas de maior prioridade quando uma tarefa de menor prioridade não pode ser escalonada, mesmo após realizar seus  $(k_i - 1)$  giros em seu próprio padrão de marcação (m,k)-firm.

Para avaliar os benefícios da abordagem proposta, desenvolveu-

se um simulador<sup>4</sup> em linguagem C. As simulações foram realizadas considerando um conjunto de tarefas (nodos), iniciando em 2 até 10 liberadas ao mesmo tempo. Os períodos das tarefas foram aleatoriamente selecionados e distribuídos entre 1 e 15. Considera-se que o *deadline* de cada tarefa é igual ao seu período ( $D_i = T_i$ ). Os valores  $m_i$  e  $k_i$  também foram gerados de forma aleatória, o valor  $k_i$  considerado foi entre 2 e 10 e  $m_i$  partindo de 1 até  $k_i$ . O tempo de execução das tarefas foi gerado impondo um factor de utilização em cada ponto, uniformemente distribuído em cada intervalo. O fator de utilização foi dividido em intervalos de comprimento 10%, iniciando em 0.2 até 1.0. Com o objetivo de minimizar erros estatísticos nos resultados das simulações, cada resultado obtido refere-se a 1000 execuções, o que significa que 9000 conjuntos de tarefas diferentes foram gerados durante os intervalos simulados, para cada resultado apresentado.

Na literatura (QUAN; HU, 2000), pode-se encontrar abordagens onde a classificação dos padrões das tarefas em mandatório ou opcional é definido *a priori* através de um algoritmo genético. Neste trabalho, adota-se padrões de classificação bem definidos, considerando-se giros nos padrões de classificação (m,k)-*firm*, evitando-se assim a explosão combinatória dos padrões de classificação.

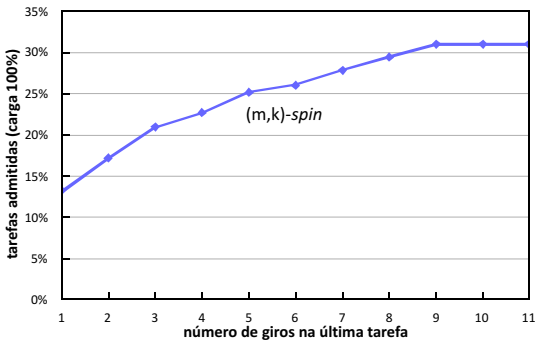


Figura 37: Porcentagem de tarefas admitidas pelo (m,k)-*spin*, que foram rejeitas por *Ramanathan* (com carga=100%).

A Figura 37 apresenta as vantagens na utilização de giros nos padrões de classificação. A avaliação apresentada refere-se à simulação com 100% de carga e com giros no padrão (m,k)-*firm* aplicados apenas na última tarefa do conjunto de tarefas. Pode-se observar que, no que

<sup>4</sup>Disponível em: [http://www.das.ufsc.br/~tisemp/\(m,k\)-spin/](http://www.das.ufsc.br/~tisemp/(m,k)-spin/).

concerne o número de tarefas admitidas, o benefício obtido alcança 31% quando até 9 giros são admitidos no padrão  $(m,k)$ -*firm*. Obviamente, além desse valor os benefícios não são relevantes, pois o número máximo de giros admitidos por uma tarefa é  $(k_i - 1)$ . É interessante notar que mesmo quando apenas 1 giro na última tarefa é considerado, já existe uma melhoria de cerca de 14% no número de tarefas aceitas.

A Figura 38 ilustra os benefícios da abordagem  $(m,k)$ -*rec-spin* para as mesmas condições supracitadas. Neste cenário, todas as tarefas pertencentes ao conjunto de tarefas podem girar seus padrões  $(m,k)$ -*firm* para beneficiar a escalonabilidade do sistema. Observou-se que cerca de 51% das tarefas foram admitidas no sistema quando 150 giros foram aplicados. Não obstante, o tempo de resposta para lidar com as operações de giro nos padrões de classificação aumentou substancialmente.

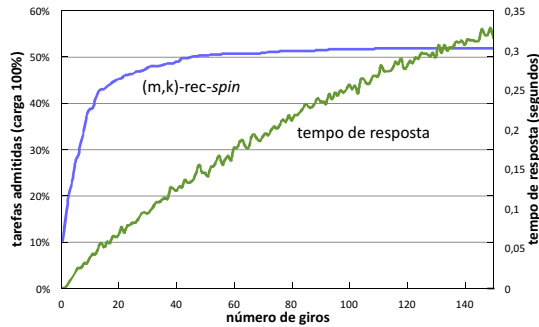


Figura 38: Porcentagem de tarefas admitidas pelo  $(m,k)$ -*rec-spin*, que foram rejeitas por *Ramanathan* (com carga=100%).

A Figura 39 apresenta os resultados variando-se a carga para três abordagens específicas. A primeira é a abordagem *Ramanathan* (*RAMANATHAN*, 1999), a segunda considera  $(k_i - 1)$  giros apenas na última tarefa do conjunto de tarefas e a terceira admite até  $(k_i - 1)$  giros para todas as tarefas do conjunto. Neste cenário, conjuntos de tarefas harmônicos foram considerados (valores  $k_i.T_i$  são múltiplos de todas as tarefas de menor prioridade), resultando na diminuição dos tempos de resposta para o conjunto de tarefas.

Mesmo quando cargas mais baixas são submetidas ao sistema, as vantagens das duas abordagens propostas são altamente significativas (Figura 39). É interessante notar que com carga iniciando em 50%, o número de tarefas admitidas em *Ramanathan* é consideravelmente

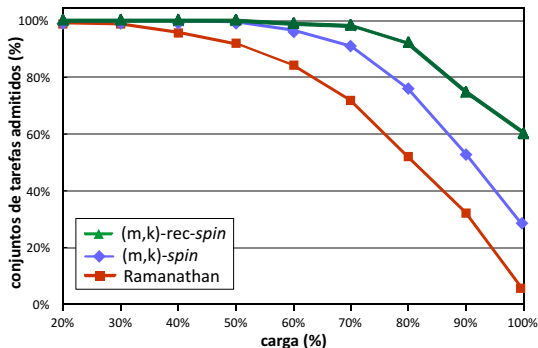


Figura 39: Porcentagem de tarefas admitidas, considerando apenas conjuntos harmônicos.

menor. Pode-se observar que para a abordagem  $(m,k)$ -*rec-spin*, cerca de 60% das tarefas foram admitidas com 100% de carga. Mesmo para a abordagem  $(m,k)$ -*spin*, onde giros são suportados apenas na última tarefa do conjunto, a melhoria na taxa de aceitação é cerca de 30% considerando 100% de carga.

Nas Tabelas 9 e 10, as colunas nomeadas como “benefício” representam os benefícios alcançados pela abordagem  $(m,k)$ -*spin* quando comparada com a abordagem Ramanathan (RAMANATHAN, 1999). A Tabela 9 refere-se a execução de conjuntos de tarefas não harmônicos. Pode-se verificar que os benefícios atingidos pela abordagem  $(m,k)$ -*spin* sobre a abordagem Ramanathan são significantes, especialmente quando a carga do sistema excede 80%.

Carga	Ramanathan	$(m,k)$ - <i>spin</i>	Benefício (%)
20%	998	999	0.0
30%	994	999	1.0
40%	989	998	1.0
50%	969	974	1.0
60%	890	924	4.0
70%	849	884	4.0
80%	665	734	10.0
90%	391	485	24.0
100%	91	179	97.0

Tabela 9: Aceitação para conjuntos de tarefas não harmônicos.

A Tabela 10 considera o caso onde conjuntos de tarefas harmônicos são considerados. Observa-se um benefício extra quando comparado com os resultados apresentados na Tabela 9. Pode-se notar que mesmo para cenários com baixo fator de utilização, os benefícios da abordagem  $(m,k)$ -*spin* são dignos de serem mencionados, com 60% de carga verifica-se um benefício de 16% na abordagem  $(m,k)$ -*spin* com relação a abordagem Ramanathan. Os benefícios tornam-se mais significantes com o aumento do fatores de utilização (chegam a mais de 460% com 100% de carga no sistema).

Carga	Ramanathan	$(m,k)$ - <i>spin</i>	Benefício (%)
20%	994	1000	1.0
30%	989	1000	1.0
40%	961	1000	4.0
50%	920	997	9.0
60%	841	968	16.0
70%	721	912	28.0
80%	520	760	46.0
90%	324	530	64.0
100%	48	276	462.0

Tabela 10: Aceitação para conjuntos de tarefas harmônicos.

Como conclusão da avaliação por simulação, observou-se que mesmo com apenas um giro no padrão de classificação  $(m,k)$ -*firm*, o número de tarefas admitidas no conjunto pode ser significativamente aumentada, quando comparado com a abordagem original, proposta por Ramanathan (RAMANATHAN, 1999). Ademais, quando um número ilimitado de giros no padrão de classificação  $(m,k)$ -*firm* é suportado, a taxa de aceitação do conjunto de tarefas é ainda maior. Porém, este giro ilimitado implica no uso de maiores recursos computacionais, devido ao elevado número de combinações possíveis nos padrões  $(m,k)$ -*firm*, que precisam ser verificados pelo teste de escalabilidade.

No tocante à periodicidade das tarefas, verificou-se que conjunto de tarefas harmônicas apresentou melhores resultados quando comparados com conjuntos não harmônicos, o que é perfeitamente compreensível, pois a abordagem proposta em (RAMANATHAN, 1999) não considera giros no padrão de classificação  $(m,k)$ -*firm*. Isto é ainda mais evidente em fatores de utilização mais elevados, quando o número de tarefas no conjunto é maior.

### Implementação em dispositivos IEEE 802.15.4

A avaliação experimental da proposta foi concebida objetivando demonstrar a factibilidade de implementação da abordagem em dispositivos sensores compatíveis com o padrão IEEE 802.15.4.

A configuração do cenário de experimentos é formado por 6 dispositivos MicaZ executando o microkernel OpenZB<sup>5</sup>. Os sensores MicaZ são baseados no microcontrolador Atmel Atmega 128L, o qual opera em 8 MHz, contendo 128 KBytes de memória flash de programação. Existem outros nodos sensores que apresentam melhor desempenho, como o SunSpot que opera em 180 MHz, com 4 MBytes de memória flash disponível.

Os dispositivos foram distribuídos segundo a topologia estrela. Um dos sensores desempenha o papel de Coordenador PAN e outros 5 foram configurados para enviar solicitações de alocação de GTS ao coordenador, contendo suas restrições  $(m,k)$ -firm. Por uma questão de simplicidade, o número de GTSs disponíveis considerado neste cenário é 3, portanto existem 5 dispositivos sensores competindo por um número limitado de compartimentos. A estrutura do superquadro foi configurada utilizando os parâmetros  $BO=SO=6$ , desta forma o valor de BI é de aproximadamente 1 segundo. Para melhor avaliar o desempenho do teste de escalonabilidade, a avaliação foi realizada 15 vezes para cada cenário analisado e os parâmetros das tarefas foram reconsiderados a cada novo experimento.

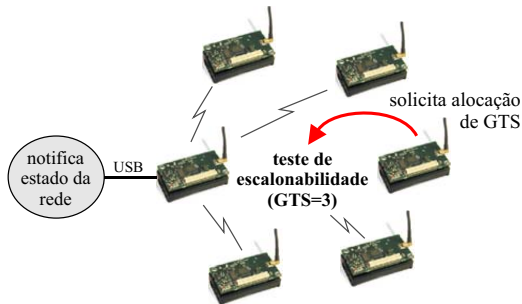


Figura 40: Cenário experimental.

Na avaliação experimental (Figura 40), o Coordenador PAN realiza o teste de escalonabilidade e notifica os dispositivos sensores sobre

<sup>5</sup><http://www.open-zb.net/>

o resultado do teste. De acordo com o padrão IEEE 802.15.4, o Coordenador PAN deve tomar decisões de escalonamento dentro de  $aGTSDescPersistenceTime$  (onde  $aGTSDescPersistenceTime = 4$ ) superquadros. Portanto, após o recebimento de uma mensagem de reconhecimento à solicitação de alocação de GTS, o dispositivo continua monitorando quadros de *beacon* e aguarda, no máximo,  $aGTSDescPersistenceTime$  superquadros.

Os valores obtidos para o tempo de resposta médio, pior caso de tempo de resposta e desvio padrão são apresentados na Tabela 11. Observa-se que mesmo quando conjuntos não harmônicos são considerados, o tempo de resposta é aceitável para dispositivos compatíveis com o padrão IEEE 802.15.4. No cenário de avaliação, o pior caso de tempo de resposta é limitado por  $aGTSDescPersistenceTime \cong 4$  segundos.

	$(k_i - 1)$ giros		1 giro	
	Não harm.	Harm.	Não harm.	Harm.
Tempo de resposta	143 ms	30 ms	80 ms	20 ms
Pior caso	300 ms	60 ms	170 ms	30 ms
Desvio padrão	100 ms	15 ms	40 ms	7 ms

Tabela 11: Tempos de resposta (ms).

Quando conjuntos de tarefas harmônicos com apenas 1 giro são considerados, os tempos de resposta obtidos são significativamente melhores devido ao fato do MMC das tarefas ser mais restrito, por conseguinte o número de períodos a serem verificados pelo teste de escalabilidade são menores.

## 6.2 COMPARAÇÃO COM TRABALHOS RELACIONADOS

A Figura 41 apresenta um quadro com as características mais relevantes relacionadas aos principais trabalhos estudados e desenvolvidos para escalonamento de compartimentos garantidos durante o CFP. O trabalho desenvolvido apresenta um teste de escalabilidade, realizado em tempo de execução (*on-line*), no qual as tarefas admitidas pelo teste serão garantidamente atendidas. O algoritmo  $(m,k)$ -*spin* toma suas decisões de escalonamento segundo alguma política de prioridade fixa (no caso, utilizou-se o RM). Considerando o classificador de tarefas utilizado nesse trabalho – no qual os padrões de classificação

são previamente gerados e, portanto, o número total de combinações possíveis é conhecido e limitado –, o teste de escalonabilidade proposto é considerado exato. Por fim, a última linha do quadro considera um algoritmo como ótimo se, caso ele falhe no atendimento de um *deadline* nenhum outro algoritmo da mesma classe é capaz de atender o *deadline*.

	(Ramanathan, 1999)	(Goossens, 2008)	(Li; Song; Simonot, 2006)	(Quan; Hu, 2000)	(Bernat; Burns, 1997)	(Semprebom, et al., 2012)
Teste	suficiente	exato	suficiente	suficiente	suficiente	exato
Preempção	preemptivo	não-preemp.	não-preemp.	preemptivo	preemptivo	preemptivo
Escalonamento	dinâmico (DBP)	dinâmico (DBP)	dinâmico (DBP)	estático	estático ( <i>dual priority</i> )	estático
Execução	<i>on-line</i>	<i>off-line</i>	<i>on-line</i>	<i>off-line</i>	<i>off-line</i>	<i>on-line</i>
Optimalidade	heurístico	ótimo	heurístico	heurístico	ótimo	ótimo

Figura 41: Análise comparativa entre os trabalhos.

É importante ressaltar que o trabalho (BERNAT; BURNS, 1997) apresenta um algoritmo de escalonamento com prioridades duplas (*dual priority*), onde as prioridades das tarefas são estabelecidas em tempo de projeto, porém são atualizadas dinamicamente em tempo de execução.

### 6.3 CONCLUSÕES DO CAPÍTULO

Com o objetivo de oferecer suporte de comunicação para aplicações tempo real em RSSFs, um mecanismo de priorização na subcamada MAC deve ser adotado para diferenciação de tráfego. Uma das soluções mais promissoras projetada para oferecer garantias temporais é o mecanismo de alocação de GTSs, proposto pelo padrão IEEE 802.15.4. Apesar de ser atrativo, este mecanismo oferece fortes restrições relacionadas com o número máximo de GTSs disponíveis em cada superquadro – sete compartimentos.

Considerando a flexibilidade de algumas tarefas de tempo real no que concerne o descarte de algumas ativações e o número reduzido de compartimentos de tempo garantido disponíveis em redes IEEE 802.15.4, um caminho lógico a ser seguido é explorar as restrições de tempo real dessas redes buscando otimizar a utilização deste recurso escasso, tal como a alocação de GTSs. Infelizmente, o uso de mecanismos de tempo real apropriados em RSSFs não tem sido totalmente investigado. O provimento de garantias temporais em RSSFs ainda é um desafio em aberto. Este capítulo apresentou dois novos testes de escalonabilidade exatos para tarefas com restrições  $(m,k)$ -*firm*.



A partir do modelo descrito no Capítulo 4 um primeiro artigo foi produzido (SEMPREBOM et al., 2009c) para um evento internacional. Um dos principais resultados deste artigo foi a definição de um modelo de escalonamento de compartimentos, denominado SIGHT. Esta proposta oferece garantias temporais para aplicações em duas fases: (i) através de um teste de escalonabilidade *on-line*, baseado no fator de utilização das tarefas e (ii) um algoritmo de escalonamento de GTSSs dinâmico, o qual estende o conceito de Distância para Falha (DBP) para redes IEEE 802.15.4. Apesar dos resultados obtidos pelo modelo SIGHT serem significantes, o mesmo impõe duas severas restrições ao modelo de tarefas: todas as tarefas possuem os mesmos valores  $k$  e mesmo período ( $T_i$ ). A adoção destas restrições são fundamentais para o funcionamento da proposta, o que pode ser um fator impeditivo para uma vasta gama de aplicações.

Contrariamente ao modelo SIGHT, em (SEMPREBOM et al., 2012a, 2012b) uma nova abordagem foi desenvolvida. A abordagem considera um modelo de tarefas flexível e genérico para tarefas com restrições de QoS representadas de acordo com o modelo  $(m,k)$ -*firm*. A análise de escalonabilidade da abordagem proposta baseia-se no conceito de período de ocupação. O teste de escalonabilidade proposto, que pode ser executado em tempo polinomial, é executado como um teste de admissão. A abordagem intitulada  $(m,k)$ -*spin* adota um padrão de classificação de tarefas bem definido, considerando giros nos padrões  $(m,k)$ -*firm* das tarefas.

Adicionalmente, as duas abordagens propostas nesse capítulo mantêm compatibilidade com dispositivos que já implementam a pilha IEEE 802.15.4 – COTS.



## 7 CONCLUSÕES

O presente capítulo conclui esta tese. Primeiramente, as motivações e objetivos traçados neste trabalho são revisados. Posteriormente, uma visão geral sobre o desenvolvimento desta tese é apresentada. Por fim, as contribuições e alguns trabalhos futuros são expostos.

### 7.1 REVISÃO DAS MOTIVAÇÕES E OBJETIVOS

Ao longo dos últimos anos as redes sem fios tem desempenhado um papel importante nos sistemas de automação industrial e controle de processos (WILLIG, 2008). No tocante aos ambientes de comunicação industriais, um grande esforço vem sendo realizado para mover parte da infraestrutura de comunicação antes cabeada, para redes sem fios. A adoção de redes sem fios em ambientes industriais possibilita a fácil configuração do maquinário industrial, de seus componentes e a implementação de funcionalidades *plug-and-play*, oferecendo mobilidade e flexibilidade para os sistemas de controle, além de reduzir os custos relacionados a construção e manutenção desses sistemas. Dentro deste contexto, a disponibilidade de soluções de redes sem fios criará um padrão *de facto* para comunicação sem fios em ambientes industriais, sendo o padrão IEEE 802.15.4 um forte candidato a torna-se este padrão para redes LR-WPANs.

Objetivando oferecer suporte de comunicação tempo real em redes de sensores sem fios, mecanismos para oferecer QoS em mensagens na Subcamada MAC deve ser adotada para diferenciação de tráfego. Uma das soluções mais promissoras objetiva o provimento de garantias de tempo real através do mecanismo de alocação de GTSs, proposto na padrão IEEE 802.15.4. Este mecanismo é bastante atrativo, pois quando a rede opera no modo com *beacon*, uma determinada largura de banda pode ser reservada para dispositivos que solicitam garantias temporais. Este mecanismo fornece um intervalo de máximo de atendimento para as aplicações através da alocação explícita de compartimentos garantidos. Desta forma, um pior caso de tempo de resposta pode ser conhecido *a priori* pelas aplicações que utilizam esse recurso.

No entanto, uma das principais limitações observadas pelo mecanismo de alocação de GTSs é que este é um recurso escasso, que deve ser compartilhado por todos nodos da rede – de acordo com o padrão IEEE 802.15.4 o número de GTSs disponíveis durante o CFP

é limitado a sete. Este recurso pode ser rapidamente exaurido quando considerada um rede com um grande número de dispositivos. Dispositivos que não possuem GTSs alocados devem tentar realizar suas transmissões durante períodos de acesso com contenção, utilizando o tradicional mecanismo de acesso ao meio CSMA/CA o qual não prevê nenhuma garantia temporal para às aplicações.

Conforme ressaltado no Capítulo 1, considerando as limitações encontradas em aplicações projetadas segundo a especificação IEEE 802.15.4, de uma forma geral, este trabalho foi guiado no sentido de estudar e propor soluções para duas questões básicas:

1. É possível desenvolver novas abordagens que estendam o Controle de Acesso ao Meio (MAC) de redes IEEE 802.15.4 de forma a explorar a capacidade de tarefas periódicas lidarem, em tempo de execução, com eventuais perdas de *deadlines* sem nenhuma alteração ao padrão IEEE 802.15.4?
2. É possível que estas abordagens atuem tanto nos mecanismos de envio e recepção de mensagens que usam tanto os compartimentos garantidos (CFP – *Contention Free Period*), quanto nas que usam o protocolo CSMA/CA (CAP – *Contention Access Period*)?

Seguindo esses objetivos, nesta tese foram propostas e desenvolvidas abordagens *on-line* para provimento de QoS para aplicações que toleram perdas de *deadlines* modeladas de acordo com restrições  $(m,k)$ -*firm*. Foram desenvolvidas uma abordagem probabilística para priorização de tráfego durante períodos de acesso com contenção e duas abordagens determinísticas para escalonamento de GTSs durante períodos livres de contenção.

## 7.2 VISÃO GERAL DO TRABALHO

Este trabalho apresentou estudos sobre o desenvolvimento de abordagens de provimento de QoS para aplicações de RSSFs, com restrições temporais que toleram descartes de algumas ativações. Estas aplicações possuem suas restrições de QoS representadas segundo o modelo de tarefas  $(m,k)$ -*firm*. Segundo este modelo a cada fluxo de mensagens  $S_i$  ao menos  $m_i$  mensagens serão transmitidas dentro de cada janela de  $k_i$  *deadlines* consecutivos.

As propostas apresentadas nesta tese buscam propor soluções para aplicações industriais que utilizam o protocolo IEEE 802.15.4 como suporte de comunicação para suas aplicações. Todas as propostas

foram projetadas objetivando manter compatibilidade com a especificação IEEE 802.15.4, tornando assim as soluções propostas aplicáveis a qualquer dispositivos que já implemente o padrão IEEE 802.15.4. O trabalho por ser dividido em duas partes principais.

Na etapa inicial, investigou-se o mecanismo de acesso com contenção oferecido pelo padrão IEEE 802.15.4. Nesse sentido, uma avaliação das capacidades temporais foi conduzida com dispositivos sensores que implementam a pilha IEEE 802.15.4. Observou-se que a capacidade de cumprimento dos períodos das tarefas e de sincronização estabelecidos pelo padrão não podem ser atendidos, devido à limitações físicas impostas pelo dispositivos sensores (eg. processador e memória).

Devido à influência do trabalho (KOUBAA et al., 2006a), um esquema de priorização de mensagens de tempo real (ex. alarmes) foi implementado. O algoritmo proposto implementa, de forma descentralizada, a priorização de alguns fluxos de mensagens através da configuração dinâmica de parâmetros do protocolo de acesso ao meio CS-MA/CA. A solução proposta foi avaliada através de um cenário real, onde dispositivos sensores foram submetidos a sobrecargas severas. A partir deste cenário experimental, observou-se que, apesar da significativa diminuição na ocorrência de falhas dinâmicas quando adotado o Distributed DBP, em algumas condições de carga (mesmo moderada) a priorização das mensagens não mostrou-se adequada.

Em um segundo momento, investigou-se o desenvolvimento de mecanismos para superar a limitação do número máximo de GTSs oferecidos durante períodos livres de contenção. Dentro deste contexto, duas soluções foram propostas:

- A primeira abordagem, intitulada SIGHT (Skips In GTS scHeduling for IEEE 802.15.4 neTworks) incorpora um esquema de alocação dinâmico de GTSs para tarefas com restrições  $(m,k)$ -firm. Esta abordagem adota um teste de escalonabilidade determinístico e um algoritmo de priorização de *on-line* chamado Slotted DBP. Para que o teste de escalonabilidade oferecido pela abordagem SIGHT continue sendo exato (necessário e suficiente), duas limitações não negligenciáveis devem ser adicionadas ao modelo de tarefas: (i) todas as tarefas possuem o mesmo valor de  $k$  e (ii) todas as tarefas possuem o mesmo período (delimitado pela periodicidade do superquadro). Tais restrições podem ser demasiadamente severas para algumas aplicações, porém este é o preço a ser pago, caso seja necessário a adoção de testes de escalonabilidade eficientes, capazes de admitir um número maior de tarefas no sistema.

- A segunda proposta de escalonamento de GTSs durante o CFP foi desenvolvida buscando flexibilizar as limitações impostas pelo SIGHT, aumentando concomitantemente a escalonabilidade do sistema, quando comparado ao trabalho originalmente proposto por (RAMANATHAN, 1999). Neste sentido, uma nova abordagem para lidar com o escalonamento de GTSs em redes IEEE 802.15.4 foi introduzida. Adicionalmente, um teste de escalonabilidade exato, baseado no conceito de período de ocupação (*busy period*) é apresentado e avaliado. O teste apresentado pode ser executado em tempo polinomial, implementado como um teste de admissão para os novos fluxos de mensagens que desejam realizar suas transmissões durante o CFP.

No que tange à implementação dessas duas soluções de provimento de QoS durante períodos de acesso com e sem contenção, CAP e CFP, respectivamente. Pode-se considerar que dispositivos que não conseguem alocar GTSs, portando não estão aptos a realizar suas transmissões durante períodos livres de contenção, podem tentar realizar suas transmissões nos próximos superquadros, disputando acesso com os outros dispositivos, durante períodos com contenção. Estes dispositivos podem ser beneficiados através de técnicas de priorização de fluxos mensagens, minimizando desta forma a ocorrência de falhas dinâmicas no sistema.

### 7.3 CONTRIBUIÇÕES DA TESE

Nesta seção apresenta-se as principais contribuições desta tese, juntamente com os resultados publicados que cada abordagem gerou. Estas contribuições vão ao encontro do cumprimentos dos objetivos estabelecidos e também são aqui abordadas.

#### 1. Especificação do modelo de sistema para facilitar a verificação de algumas propriedades de tempo real importantes para a proposta.

Um modelo de sistema para tarefas com restrições de QoS modeladas de acordo com *deadlines*  $(m,k)$ -firm é apresentado no Capítulo 4. Este modelo foi primeiramente aplicado à proposta SIGHT de escalonamento no CAP. Em um segundo momento, o modelo foi introduzido impondo algumas restrições na proposta de escalonamento no CFP. Estas restrições foram relaxadas e uma nova abordagem menos restritiva foi proposta.

2. **Criação de um algoritmo de escalonamento de tarefas com restrições de QoS especificadas em conformidade com modelo  $(m,k)$ -firm, durante períodos de acesso com contenção, em redes IEEE 802.15.4.**

O modelo delineado no Capítulo 4 foi implementado em uma abordagem de priorização de tráfego descentralizada desenvolvida durante períodos de acesso com contenção é apresentada no Capítulo 5.

3. **Proposição de um mecanismo de escalonamento de compartimentos de tempo garantido em redes IEEE 802.15.4.**

Visando aprimorar o esquema de alocação de GTSs para dispositivos que necessitam de garantias temporais. Duas soluções de escalonamento de GTSs durante períodos de acesso livres de contenção são propostas no Capítulo 6.

4. **Definição de testes de escalonabilidade exato para tarefas  $(m,k)$ -firm em redes IEEE 802.15.4.**

No Capítulo 6 dois testes de escalonabilidade determinísticos são propostos.

5. **Análise de escalonabilidade da proposta.**

Objetivando verificar a corretude do teste proposto, no Capítulo 6, uma análise de escalonabilidade completa do teste é realizada.

6. **Avaliação das abordagens através de simulações e de um cenário experimental.**

Todas as abordagens propostas nesse trabalho foram avaliadas através de simulações e/ou através da implementação das propostas em dispositivos sensores compatíveis com a especificação IEEE 802.15.4.

Uma parte das contribuições listadas acima foram publicadas em conferências internacionais (SEMPREBOM et al., 2009c), (SEMPREBOM et al., 2009b), (SEMPREBOM et al., 2009a), (CABREIRA et al., 2009), (SEMPREBOM et al., 2012b) e (SEMPREBOM et al., 2012a), e em conferências nacionais (SEMPREBOM et al., 2009), (SEMPREBOM et al., 2010) e (SEMPREBOM et al., 2011).

## 7.4 TRABALHOS FUTUROS

As propostas desenvolvidas nesta tese certamente não finalizam as discussões sobre o provimento de QoS para aplicações com restrições temporais em ambientes de RSSFs. Além do mais, as pesquisas nesta área são relativamente novas, quando comparadas com as pesquisas que consideram redes cabeadas, e sendo este assunto muito amplo, dificilmente as opções de pesquisa que o envolvem serão esgotadas.

Deste modo, alguns trabalhos relacionados com esta tese poderão ser explorados no futuro. O primeiro está relacionado com o mecanismo de priorização de mensagens durante períodos livres de contenção em redes IEEE 802.15.4, a proposição de novas abordagens devem ser melhor analisadas. O algoritmo de escalonamento baseado na distância para falha DBP não é ótimo, pois este mecanismo não leva em consideração outros parâmetros das tarefas, como períodos e tempo de computação. Deste modo, novas heurísticas, que considerem os períodos das tarefas em suas decisões de escalonamento precisam ser investigadas em trabalhos futuros.

Algumas aplicações de controle de processos que toleram descartes necessitam de uma especificação precisa de como as eventuais perdas de *deadline* estão espaçadas. Desta forma, o modelo de tarefas *Skip-Over* ((KOREN; SHASHA, 1995)) pode ser uma solução atrativa para este tipo de aplicações, uma vez que neste modelo de tarefas as restrições de QoS das aplicações são especificadas em função do espaçamento entre a ocorrências de perdas de *deadlines* suportados pela aplicação. Sendo assim, a utilização do modelo *Skip-Over* em aplicações de controle que toleram descartes de ativações deverá ser melhor estudado.

O modelo de comunicação proposto nesta tese assume que o canal de comunicação é livre de erros. Sabe-se que os sistemas de comunicação sem fios são frequentemente caracterizados por ocorrências de erros, devido a características naturais propensas a falhas de meio de comunicação. Como resultado, mensagens podem precisar de mais do que uma transmissão para serem entregues corretamente. Esta possibilidade de retransmissões deve ser tratada em pela subcamada MAC ou pelo níveis superiores, representado pelo algoritmo de escalonamento. De qualquer forma, o impacto no uso das retransmissões é certamente relevante e deve ser melhor investigado em conjunto com alterações nos testes de escalonabilidade propostos nesta tese.

O classificador de tarefas adotado neste trabalho assume rotações (*spins*) a partir de padrões de classificação bem definidos. De acordo



com (QUAN; HU, 2000) a classificação ótima de ativações de tarefas em mandatória ou opcional é um problema completo, que não pode ser tratado em tempo polinomial (NP-difícil). De qualquer forma, novas estratégias de classificação de ativações de tarefas com restrições  $(m,k)$ -*firm* podem ser consideradas. Novos parâmetros podem ser atrelados aos instantes de ativação das tarefas, objetivando por exemplo economia de energia dos sensores. Por exemplo, pode-se considerar um classificador que concentre todas as ativações mandatórias das tarefas em seus primeiros períodos, sendo que os dispositivos sensores poderiam entrar em modo de economia de energia durante ativações opcionais.



## REFERÊNCIAS

802.11. IEEE Standard for Information Technology - Wireless LAN Medium Access Control and Physical Layer Specification Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements. *ANSI/IEEE Std 802.11e-2005*, 2005.

802.11. IEEE Standard for Information Technology - Wireless LAN Medium Access Control and Physical Layer Specification. *ANSI/IEEE Std 802.11, 2007 (Revision of IEEE 802.11-1999)*, 2007.

802.15.4. Part 15.4: Wireless medium access control (MAC) and physical layer (phy) specifications for low-rate wireless personal area network (lr-wpan). *in IEEE-SA Standards Board 802.15.4-2006*, 2006.

ABRAMSON, N. The aloha system - another alternative for computer communications. *Proceedings of Fall Joint Computer Conference*, p. 281–285, 1970.

ANASTASI, G.; CONTI, M.; FRANCESCO, M. D. A comprehensive analysis of the MAC unreliability problem in IEEE 802.15.4 wireless sensor networks. *IEEE Transactions on Industrial Informatics*, v. 7, n. 1, p. 52–65, 2011.

ANDERSSON, B.; PEREIRA, N.; TOVAR, E. Analysing TDMA with Slot Skipping. *IEEE Transactions on Industrial Informatics*, v. 4, n. 4, p. 225–236, 2008.

ANDERSSON, B.; TOVAR, E.; PEREIRA, N. Analysing TDMA with Slot Skipping. *Proceedings of 26rd IEEE Real-Time Systems Symposium (RTSS'05)*, p. 10–24, 2005.

ARTIST. <http://www.artist-embedded.org/artist/Brief-State-of-the-Art,627.html>. 2009.

ATLAS, A.; BESTAVROS, A. Statistical rate monotonic scheduling. p. 123–132, 1998.

AUDSLEY, N.; BURNS, A.; RICHARDSON, M.; TINDELL, K.; WELLINGS, A. J. Applying new scheduling theory to static priority pre-emptive scheduling. *Software Engineering Journal*, v. 8, n. 5, p. 284–292, 1993.

AUDSLEY, N. C.; BURNS, A.; RICHARDSON, M.; WELLINGS, A. J. Hard real-time scheduling: The deadline monotonic approach. *Proceedings of the 8th Workshop on Real-Time Operating Systems and Software*, p. 127–132, 1991.

BELLO, L. L.; TOSCANO, E. Coexistence Issues of Multiple Co-located IEEE 802.15.4/ZigBee Networks Running on Adjacent Radio Channels in Industrial Environments. *IEEE Transactions on Industrial Informatics*, v. 5, n. 2, p. 157–167, 2009.

BERNAT, G.; BURNS, A. Combining (n/m)-hard deadlines and dual-priority scheduling. *Proceedings Real-Time Systems Symposium*, p. 46–57, 1997.

BURNS, A.; WELLINGS, A. *Real-Time Systems and Programming Languages*. Fourth edition. England: Addison-Wesley, 2001.

BUTTAZZO, G.; SPURI, M.; SENSINI, F. Value vs deadline scheduling in overload conditions. *Proceedings of the 16th Real-Time Systems Symposium*, p. 90–99, 1995.

BUTTAZZO, G.; STANKOVIC, J. Red: A robust earliest deadline scheduling algorithm. *Proceedings of the 3rd International Workshop on Responsive Computing Systems*, v. 10, n. 6, p. 110–111, 1993.

BUTTAZZO, G.; VELASCO, M.; MARTI, P. Quality-of-control management in overloaded real-time systems. *IEEE Transactions on Computers*, v. 56, p. 253–266, 2007.

BUTTAZZO, G. C. *Hard Real-Time Computing Systems - Predictable Scheduling Algorithms and Applications*. Second edition. New York: Springer, 1997.

BUTTAZZO, G. C.; LIPARI, G.; ABENI, L. Elastic task model for adaptative rate control. *Proceedings of the 19th Real-Time Systems Symposium*, p. 286–295, 1998.

CABREIRA, U.; SEMPREBOM, T.; MONTEZ, C. A coordination protocol for target tracking in wireless sensor networks. *XXXV Latin-American Conference on Informatics*, 2009.

CACCAMO, M.; ZHANG, L. Y. An implicit prioritized access protocol for wireless sensor networks. *Proceedings of 23rd IEEE Real-Time Systems Symposium (RTSS'02)*, p. 39–48, 2002.

CARLEY, T. W.; BA, M. A.; BARUA, R.; STEWART, D. B. Contention-free periodic message scheduler medium access control in wireless sensor/actuator networks. *Proceedings of 24th IEEE Real-Time Systems Symposium (RTSS'03)*, p. 298–307, 2003.

CHEN, F. Improving IEEE 802.15.4 for low-latency energy industrial applications. *Proceedings of Echtzet*, p. 61–70, 2008.

CHEN, F.; TALANIS, T.; GERMAN, R.; DRESSLER, D. Real-time enabled IEEE 802.15.4 sensor networks in industrial automation. *International Symposium on Industrial Embedded Systems*, p. 136–139, 2009.

CHEN J. FERREIRA, L. T. E. An explicit GTS allocation algorithm for IEEE 802.15.4. *16th IEEE Conference on Emerging Technologies & Factory Automation*, p. 1–8, 2011.

CHENG, L.; BOURGEOIS, A. G.; ZHANG, X. A new GTS allocation scheme for IEEE 802.15.4 networks with improved bandwidth utilization. *International Symposium on Communication and Information Technologies*, p. 1143–1148, 2007.

CHO, H.; CHUNG, Y.; PARK, D. Guaranteed dynamic priority assignment schemes for real-time tasks with  $(m, k)$ -firm deadlines. *ETRI Journal*, v. 32, n. 3, p. 422–429, 2010.

CUNHA, A.; KOUBAA, A.; SEVERINO, R.; ALVES, M. Open-zb: an open-source implementation of the ieee 802.15.4/zigbee protocol stack on tinys. *IEEE International Conference on Mobile Adhoc and Sensor Systems*, p. 1–12, 2007.

CUNHA, A.; SEVERINO, R.; PEREIRA, N.; KOUBAA, A.; ALVES, M. Zigbee over tinys: Implementation and experimental challenges. *Proceedings of the 8th Portuguese Conference on Automatic Control*, 2008. Invited Session on Real-Time Communications: from theory to applications.

DAVIS, R.; WELLINGS, A. Dual priority scheduling. *Proceedings of Real-Time Systems Symposium*, p. 100–109, 1995.

DONG, L.; MELHEM, R.; MOSSÉ, D. Scheduling algorithm for dynamic messages streams with distance constraints in tdma protocol. *Proceedings of 12th Euromicro Conference on Real-Time Systems (ECRTS'00)*, p. 239–246, 2000.

- FARINES, J. M.; FRAGA, J. S.; OLIVEIRA, R. S. *Sistemas de Tempo Real*. IME-USP, São Paulo-SP: 12<sup>a</sup> Escola de Computação, 2000.
- GAY, D.; LEVIS, P.; BEHREN, R. V.; WELSH, M.; BREWER, E.; CULLER, D. The nesc language: A holistic approach to networked embedded systems. *Proceedings of Programming Language Design and Implementation (PLDI)*, v. 38, n. 5, p. 1–11, 2003.
- GOOSSENS, J. (m,k)-firm constraints and DBP scheduling: impact of the initial k-sequence and exact schedulability test. *16th International Conference on Real-Time and Network Systems*, p. 61–65, 2008.
- HAMDAOUI, M.; RAMANATHAN, P. A dynamic priority assignment technique for streams with (m,k)-firm deadlines. *IEEE Transactions on Computers*, v. 44, n. 12, p. 1443–1451, 1995.
- HAMDAOUI, M.; RAMANATHAN, P. Evaluating dynamic failure probability for streams with (m,k)-firm deadlines. *IEEE Transactions on Computers*, v. 46, n. 12, p. 1325–1337, 1995.
- HAN, C.-C.; LIN, K.-J.; HOU, C.-J. Distance-constrained scheduling and its applications to real-time systems. *IEEE Transactions on Computers*, v. 45, n. 7, p. 814–826, 1996.
- HART. *HART Communication Foundation*. [www.hartcomm.org/](http://www.hartcomm.org/).
- HEINZELMAN, A. C. W. R.; BALAKRISHNAN, H. Energy-efficient communication protocols for wireless microsensor networks. *Proceedings Hawaiian International Conference on Systems Science*, p. 3005–3014, 2000.
- HESPANHA, J. P.; NAGHSHTABRIZI, P.; XU, Y. A survey of recent results in networked control systems. *Proceedings of the IEEE*, v. 95, n. 1, p. 138–162, 2007.
- HILL, J.; SZEWCZYK, R.; WOO, A.; HOLLAR, S.; CULLER, D.; PISTER, K. System architecture directions for networked sensors. *SIGPLAN Not.*, v. 35, n. 11, p. 93–104, 2000.
- JEFFAY, K.; STANAT, D. F.; MARTEL, C. U. On non-preemptive scheduling of periodic and sporadic task. *Proceedings of the 12th IEEE Real-Time Systems Symposium*, p. 129–139, 1991.
- JENNIC. Calculating IEEE 802.15.4 Data Rates. *Application Note*, 2006.

JONSONN, M.; KUNERT, K. Towards Reliable Wireless Industrial Communication with Real-Time Guarantees. *IEEE Transactions on Industrial Informatics*, v. 5, n. 4, p. 429–442, 2009.

JOSEPH, M.; PANDYA, P. Finding response times in a real-time system. *The Computer Journal*, v. 29, n. 5, p. 390–395, 1986.

KIM, T.; LEE, D.; AHN, J. Priority toning strategy for fast emergency notification in IEEE 802.15.4 lr-wpan. *Proceedings of the 15th Joint Conference on Communications & Information (JCCI)*, 2005.

KOPETZ, H.; GRUNSTEIDL, G. Ttp - a protocol for fault-tolerant real-time systems. *IEEE Computer*, v. 27, n. 1, p. 14–24, 1994.

KOREN, G.; SHASHA, D. D-over: An optimal on-line scheduling algorithm for overloaded real-time systems. *Proceedings Real-Time Systems Symposium*, 1992.

KOREN, G.; SHASHA, D. Skip-over: Algorithms and complexity for overloaded systems that allow skips. *Proceedings Real-Time Systems Symposium*, p. 110–117, 1995.

KOUBAA, A.; ALVES, M.; NEFZI, B.; SONG, Y.-Q. Improving the IEEE 802.15.4 slotted CSMA/CA MAC for time-critical events in wireless sensor networks. *Proceedings of the 5th International Workshop on Real-Time Networks*, p. 35–40, 2006.

KOUBAA, A.; ALVES, M.; TOVAR, E. i-GAME: An implicit GTS allocation mechanism in IEEE 802.15.4 for time-sensitive wireless sensor networks. *18th Euromicro Conference on Real-Time Systems*, p. 183–192, 2006.

KOUBAA, A.; ALVES, M.; TOVAR, E.; CUNHA, A. An implicit GTS allocation mechanism in IEEE 802.15.4 for time-sensitive wireless sensor networks: theory and practice. *Real-Time Systems Journal*, v. 39, n. 1–3, p. 169–204, 2008.

LEHOCZKY, J. P. Fixed priority scheduling of periodic task sets with arbitrary deadlines. *Proceedings of 11th IEEE Real-Time Systems Symposium*, p. 201–209, 1990.

LEHOCZKY, J. P.; SHA, L.; DING, Y. The rate monotonic scheduling algorithm: exact characterization and average case behavior. *Proceedings of 10th IEEE Real-Time Systems Symposium*, p. 166–171, 1989.

- LEMMON, M.; LING, Q.; SUN, Y. Overload management in sensor-actuator networks used for spatially-distributed control systems. *SenSys '03: Proceedings of the international conference on Embedded networked sensor systems*, v. 5, n. 7, p. 162–170, 2003.
- LEUNG, J.; WHITEHEAD, J. W. On the complexity of fixed priority scheduling of periodic real-time tasks. *Performance Evaluation*, v. 4, n. 2, p. 237–250, 1982.
- LI, J.; SONG, Y.; SIMONOT-LION, F. Providing real-time applications with graceful degradation of qos and fault tolerance according to (m,k)-firm model. *IEEE Transactions on Industrial Informatics*, v. 2, n. 2, p. 112–119, 2006.
- LI, J.; SONG, Y. Q.; SIMONOT-LION, F. Schedulability analysis for systems under (m,k)-firm constraints. *Factory Communication Systems, IEEE Computer Society*, p. 23–30, 2004.
- LIU, C. L.; LAYLAND, J. W. Scheduling algorithms for multiprogramming in hard-real-time environment. *Journal of the Association for Computing Machinery*, v. 20, n. 1, p. 46–61, 1973.
- LIU, J.; LIN, K.-J.; SHIH, W.-K.; YU, A. C.-S. Algorithms for scheduling imprecise computations. *IEEE Computer*, p. 58–68, 1991.
- LIU, J. W. S. *Real-Time Systems*. New York: Prentice-Hall, 2000.
- LOW, K.; WIN, W. N.; ER, M. J. Wireless sensor networks for industrial environments. *International Conference on Computational Intelligence for Modelling, Control and Automation and Int. Conf. on Intelligent Agents, Web Technologies and Internet Commerce*, IEEE Computer Society, p. 271–276, 2005.
- MARCHAND, A.; S.-CHETTO, M. Dynamic real-time scheduling of firm periodic tasks with hard soft aperiodic tasks. *Real-Time Systems Journal*, v. 32, n. 1-2, p. 21–47, 2006.
- MARTI, P.; FUERTES, J. M.; FOHLER, G.; RAMAMRITHAM, K. Improving quality-of-control using flexible timing constraints: Metric and scheduling issues. *Proceedings of the 23rd Real-Time Systems Symposium*, p. 91–100, 2002.
- MARTI, P.; YEPEZ, J.; VELASCO, M.; VILLA, R.; FUERTES, J. M. Managing quality-of-control in network-based control systems by controller and message scheduling co-design. *IEEE Transactions on Industrial Electronics*, v. 51, p. 1159–1167, 2004.



MOK, A. K.; WANG, W. Window constrained real-time periodic task scheduling. *Proceedings of the 22nd IEEE Real-Time Systems Symposium*, p. 15–24, 2001.

MONTEZ, C. *Um Modelo de Programação e uma Abordagem de Escalonamento Adaptativo Usando RT-Corba*. Tese (Tese de doutorado) — Universidade Federal de Santa Catarina, 1999.

MONTEZ, C.; FRAGA, J.; OLIVEIRA, R.; FARINES, J.-M. An adaptive scheduling approach in real-time corba. *The 2nd IEEE International Symposium on Object-oriented Real-time distributed Computing*, p. 301–309, 1999.

NAKAMURA, E. F.; LOUREIRO, A. A. F.; FRERY, A. C. Information fusion for wireless sensor networks: Methods, models, and classifications. *ACM Computing Surveys*, v. 39, n. 3, p. 1–55, 2007.

OLIVEIRA, R. *Escalonamento de Tarefas Imprecisas em Ambiente Distribuído*. Tese (Tese de doutorado) — Universidade Federal de Santa Catarina, 1997.

QUAN, G.; HU, X. S. Enhanced fixed-priority scheduling with (m,k)-firm guarantee. *Proceedings of the 21st IEEE Real-Time Systems Symposium*, p. 79–88, 2000.

RAMAMRITHAM, K.; STANKOVIC, J. A. Scheduling algorithms and operating systems support for real-time systems. *Proceedings of the IEEE*, v. 82, n. 1, p. 55–67, 1994.

RAMAMURTHY, H.; PRABHU, B. S.; GADH, R. Wireless industrial monitoring and control using a smart sensor platform. *IEEE Sensors Journal*, v. 7, n. 5, p. 611–618, 2007.

RAMANATHAN, P. Overload management in real-time control applications using (m,k)-firm guarantee. *IEEE Transactions on Parallel and Distributed Systems*, v. 10, n. 6, p. 549–559, 1999.

RUIZ, J.; GALLARDO, J. R.; MAKRAKIS, D.; VILLASENOR-GONZALEZ, L.; MOUFTAH, H. T. Cross-layer medium access control protocol with quality-of-service guarantees for wireless sensor networks. *EURASIP Journal on Wireless Communications and Networking*, v. 2011, n. 179, p. 1687–1499, 2011.

SCHENATO, L.; FRANCESCHETTI, M.; SASTRY, S. S. Foundations of control and estimation over lossy networks. *Proceedings of the IEEE*, v. 95, n. 1, p. 163–187, 2007.

SEMPREBOM, T.; MARYAMA, V.; ZOMER, G.; MONTEZ, C.; VASQUES, F.; PORTUGAL, P. Escalonamento dinâmico de compartimento de tempo garantido em redes IEEE 802.15.4. *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, v. 1, p. 277–290, 2010.

SEMPREBOM, T.; MONTEZ, C.; JORDAN, F.; ZOMER, G.; MARYAMA, V.; CABREIRA, U. Explorando descartes de ativações de tarefas periódicas em redes IEEE 802.15.4. *XXXV Latin-American Conference on Informatics*, 2009.

SEMPREBOM, T.; MONTEZ, C.; MORAES, R.; VASQUES, F. Distributed DBP: A (m,k)-firm based distributed approach for qos provision in IEEE 802.15.4 networks. *Proceedings of the 14th IEEE International Conference on Emerging Technologies and Factory Automation*, p. 875–882, 2009.

SEMPREBOM, T.; MONTEZ, C.; MORAES, R.; VASQUES, F.; PORTUGAL, P. Dynamic GTS scheduling of periodic skippable slots in IEEE 802.15.4 wireless sensor networks. *Proceedings of the 8th IFAC International Conference on Fieldbuses and Networks in Industrial and Embedded Systems*, p. 110–117, 2009.

SEMPREBOM, T.; MONTEZ, C.; MORAES, R.; VASQUES, F. Um modelo de alocação de GTS's em redes IEEE 802.15.4 com provisão de qos para mensagens periódicas com restrições temporais. *XXIX Simpósio Brasileiro de Telecomunicações*, v. 1, p. 1–6, 2011.

SEMPREBOM, T.; MONTEZ, C.; VASQUES, F. (m,k)-firm pattern spinning to improve the GTS allocation of periodic messages in IEEE 802.15.4 networks. *Automation and Systems Department, Federal University of Santa Catarina*, 2012. (Technical Report).

SEMPREBOM, T.; MONTEZ, C.; ZOMER, G.; VASQUES, F. Guaranteed time slot allocation fo periodic messages with (m,k)-firm constraints in IEEE 802.15.4 networks. *Proceedings of the 17th IEEE International Conference on Emerging Technologies and Factory Automation*, 2012. (aceito para publicação).

SEMPREBOM, T.; ZOMER, G.; MARYAMA, V.; MONTEZ, C.; VASQUES, F. Distributed DBP: Escalonamento (m,k)-firm aplicado a redes IEEE 802.15.4. *11th Brazilian Workshop on Real-Time and Embedded Systems*, p. 159–164, 2009.

- SEVERINO, R. *On the use of IEEE 802.15.4/ZigBee for Time-Sensitive Wireless Sensor Network Applications*. Dissertação (Dissertação de mestrado) — Polytechnic Institute of Porto, School of Engineering, 2008.
- SHA, L.; RAJKUMAR, R.; LEHOCZKY, J. Priority inheritance protocols: An approach to real-time synchronization. *IEEE Transactions on Computers*, v. 39, n. 9, p. 1175–1185, 1990.
- SHIN, K.; RAMANATHAN, P. Real-time computing: A new discipline of computer science and engineering. *Proceedings of the IEEE*, v. 82, n. 1, p. 6–24, 1994.
- SHIN, K.; RAMANATHAN, P. Real-time computing: A new discipline of computer science and engineering. *Proceedings of the IEEE*, v. 82, n. 1, p. 6–24, 1994.
- SPRUNT, B. *Aperiodic Task Scheduling for Real-Time Systems*. Tese (Tese de doutorado) — Carnegie Mellon University, 1990.
- STANKOVIC, J. A.; RAMAMRITHAM, K. What is predictability for real-time systems? *Journal of the Real-Time Systems*, v. 8, n. 3, p. 62–72, 1990.
- STANKOVIC, J. A.; RAMAMRITHAM, K. The spring kernel: a new paradigm for real-time operating systems. *IEEE Software*, v. 8, p. 62–72, 1991.
- TINDELL, K.; BURNS, A.; WELLINGS, A. J. An extendible approach for analysing fixed priority hard real-time tasks. *Real-Time Systems Journal*, v. 6, n. 2, p. 133–151, 1993.
- TOSCANO, E.; BELLO, L. L. Bandwidth-efficient Admission Control for EDF-based Wireless Industrial Communication. *IEEE International Symposium on Industrial Electronics*, p. 1186–1193, 2011.
- TOSCANO, E.; BELLO, L. L. Multichannel Superframe Scheduling for IEEE 802.15.4 Industrial Wireless Sensor Networks. *IEEE Transactions on Industrial Informatics*, v. 8, n. 2, p. 337–350, 2012.
- TOVAR, E.; VASQUES, F.; BURNS, A. Communication response time in p-net networks: Worst-case analysis considering the actual token utilization. *Real-Time Systems Journal*, v. 22, n. 3, p. 229–249, 2002.

TRIO, P.

<http://ralyx.inria.fr/2005/Raweb/trio/wid31.html?highlight=Felicioniuid31>. 2009.

WEST, R.; ZHANG, Y. Dynamic window-constrained scheduling of real-time streams in media servers. *IEEE Transactions on Computers*, v. 53, n. 6, p. 744–759, 2004.

WEST, R.; ZHANG, Y. Comments on window-constrained scheduling. *IEEE Transactions on Computers*, v. 56, n. 5, p. 718–719, 2007.

WILLIG, A. Recent and emerging topics in wireless industrial communications: A selection. *IEEE Transactions on Industrial Informatics*, v. 4, n. 2, p. 102–124, 2008.

XU, J.; PARNAS, D. L. On satisfying timing constraints in hard-real-time systems. *IEEE Transactions on Software Engineering*, v. 19, n. 1, p. 70–84, 1993.

ZHANG, L.; YU, J.; DENG, X. Modelling the guaranteed qos for wireless sensor networks: a network calculus approach. *EURASIP Journal on Wireless Communications and Networking*, v. 2011, n. 82, p. 1687–1499, 2011.

ZHANG, Y.; WEST, R.; QI, X. A virtual deadline scheduler for window-constrained service guarantees. *Proceedings of the 25th IEEE Real-Time Systems Symposium (RTSS)*, 2004.

ZIGBEE-ALLIANCE. [www.zigbee.org/](http://www.zigbee.org/). 2007.