

Distributed DBP: Escalonamento (m,k)-firm Aplicado a Redes IEEE 802.15.4

Tiago Semprebom¹, Gustavo Zomer¹, Victor Maryama¹
Frederico Jordan¹, Carlos Montez¹, Francisco Vasques²

¹Depto Automação e Sistemas (DAS) – Univ. Federal de Santa Catarina (UFSC) – Brasil

²UISPA, Depto de Engenharia Mecânica (DEMEGI) – Univ. do Porto (UP) – Portugal

Resumo. O padrão IEEE 802.15.4 adota o protocolo CSMA/CA, o qual não oferece suporte para mensagens com restrições temporais. Neste trabalho é proposta uma abordagem de provimento de qualidade de serviço para aplicações de tempo real com mínimas alterações ao protocolo existente. O mecanismo proposto melhora o CSMA/CA através da atribuição descentralizada de prioridades baseada no modelo de tarefas (m,k)-firm. Resultados obtidos através de análises experimentais são apresentados.

1. Introdução

O protocolo de acesso ao meio CSMA/CA proposto pela especificação IEEE 802.15.4 não oferece suportes de QoS para aplicações de tempo real. Considerando aplicações com restrições temporais, muitas atividades existentes precisam ser realizadas periodicamente e com tempos de resposta limitados. Alguns trabalhos recentes [Hespanha et al. 2007] da comunidade de controle investigam os efeitos de descartes de ativações no desempenho dos sistemas e propõem o uso de técnicas que toleram descartes de dados periódicos, desde que estas ocorram de forma esparsada.

Modelos de tarefas que toleram descartes já eram previstos há cerca de 15 anos atrás quando [Hamdaoui and Ramanathan 1995] introduziram o modelo de tarefas com deadlines (m,k)-firm, no qual abordagens de qualidade de serviço poderiam ser oferecidas para tarefas periódicas¹. O algoritmo apresentado naquele trabalho utilizava o conceito de distância para falha como heurística para atribuição de prioridades das atividades. Uma tarefa que estivesse próxima de perder mais que m deadlines dentro de uma janela de k requisições de serviço era classificada como mandatória, e portanto, recebia a prioridade mais alta do sistema. De forma análoga, o trabalho intitulado Skip-over [Koren and Shasha 1995] se propunha a atender a um conjunto de tarefas periódicas que tolerassem descartes. Estes descartes eram caracterizados por um parâmetro de descarte s , o qual representava a tolerância desta tarefa para perda de deadlines.

No contexto de protocolos IEEE 802.15.4, atualmente poucos trabalhos exploram a flexibilidade das tarefas tempo real. Por exemplo, no trabalho proposto em [Koubaa et al. 2006] são definidos quatro cenários com diferentes classes de tráfego e diferentes definições nos parâmetros de configuração do CSMA/CA com compartimento (*slot*). Classes de tráfego com altas e baixas prioridades são atribuídas às classes de serviço, onde cada classe possui seus próprios parâmetros configurados (parâmetros de $macMinBE$, $aMaxBE$ e CW definidos pelo IEEE 802.15.4).

Diferentemente dos trabalhos citados e considerando que o protocolo IEEE 802.15.4 não oferece mecanismos de provimento de QoS às aplicações que utilizam

¹O termo tarefa, no contexto de tempo real, significa um recurso a ser escalonado. Neste trabalho, uma tarefa refere-se a transmissão de uma mensagem enviada por um nodo.

o período com contenção, esta proposta busca atender requisições com restrições segundo o modelo (m,k)-firm. Neste trabalho, como principal contribuição, propõe-se um mecanismo de priorização distribuída durante o período com contenção em redes IEEE 802.15.4, segundo o modelo de garantias (m,k)-firm. A proposta apresenta-se em conformidade com a especificação IEEE 802.15.4, propondo alterações mínimas no protocolo CSMA/CA.

2. Protocolo MAC no IEEE 802.15.4

O protocolo IEEE 802.15.4 [IEEE 2006] especifica a sub-camada MAC e a camada física para as LR-WPANs. O protocolo MAC suporta dois modos de operação que devem ser selecionados pelo coordenador: *modo com e sem beacon*. Neste trabalho, considera-se apenas o modo com *beacon*. Quando um coordenador seleciona 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 superquadro é contido em um BI (*Beacon Interval*) e limitado por dois quadros de *beacon*, possui um período ativo (dividido em 16 compartimentos igualmente espaçados) e um período inativo. O coordenador interage com a PAN durante o período ativo, e entra em modo de economia de energia durante períodos inativos.

A estrutura do superquadro é definida por dois parâmetros: O primeiro *macBeaconOrder* (BO) define o intervalo no qual o coordenador deve transmitir os quadros de *beacon*. O segundo atributo é o *macSuperFrameOrder* (SO) este atributo define o comprimento do período ativo do superquadro. A porção ativa do superquadro é composta por três partes: *beacon*, CAP (*Contention Access Period*) e CFP (*Contention Free Period*). O *beacon* deve ser transmitido sem período de contenção (CSMA/CA), iniciando no compartimento 0. O CAP deve começar imediatamente após o quadro de *beacon*. O CFP, caso exista, deve ocorrer imediatamente após o CAP e se estende até o término da porção ativa do superquadro. Todos os GTSs devem ser alocados dentro do 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).

2.1. Protocolo de acesso ao meio CSMA/CA com compartimento

O protocolo CSMA/CA no modo *com beacon* (com compartimento) utiliza três variáveis para escalonar o acesso ao meio: **NB**: número de vezes que o algoritmo CSMA/CA entra em *backoff* enquanto tenta acessar o canal. **CW**: tamanho da janela de contenção, o qual define o número de períodos de *backoff* necessários com o canal disponível antes do início da transmissão. **BE**: expoente de *backoff* que define quantos períodos de *backoff* um dispositivo deve aguardar antes de tentar avaliar a atividade do canal. A Figura 1 apresenta o fluxograma para versão com compartimento, resumida em cinco etapas.

1. *Inicialização do NB, CW e BE.*
2. *Espera aleatória*: o algoritmo busca evitar colisões aguardando um tempo de espera aleatório gerado na faixa de $[0, 2^{BE} - 1]$ períodos de *backoff*.
3. *CCA*: operação utilizada para verificar se o canal está ocupado.
4. *Canal ocupado*: se o canal é verificado ocupado, CW é reiniciado e os valores de NB e BE são incrementados, e caso o valor de NB exceda o valor de *macMaxCSMABackoffs* (no caso 5), 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, o protocolo MAC deve iniciar sua transmissão com sucesso, de outra maneira, o algoritmo volta ao passo 3.

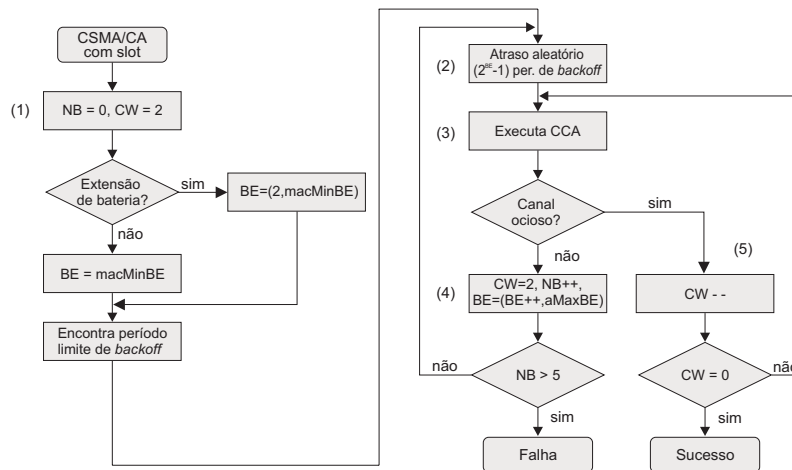


Figura 1. Algoritmo CSMA/CA com compartimento.

3. Modelo do Sistema

Uma das motivações para o desenvolvimento deste trabalho advém do fato que, apesar do protocolo IEEE 802.15.4 apresentar um mecanismo de alocação com garantias (GTS) para aplicações tempo real, a alocação precisa ser precedida por uma mensagem de solicitação de alocação de GTS. Além disso, o número máximo de compartimentos garantidos é baixo (sete *slots*). Porém, o uso do protocolo CSMA/CA conforme oferecido pela especificação também não é adequado. Este protocolo não prevê qualquer mecanismo de provimento de QoS para eventos tempo real, incluindo solicitações de alocação de GTS, e mensagens de alarme, resultando na degradação do desempenho da rede, principalmente durante condições de sobrecarga.

Para este trabalho, considera-se um cluster IEEE 802.15.4, em uma topologia estrela, composto por um conjunto de nodos sensores dentro da área de cobertura de um único nodo denominado Coordenador PAN. Assume-se que o Coordenador PAN inicializa a rede definindo a estrutura do superquadro através dos parâmetros BO e SO. Cada nodo sensor i coleta uma variável do ambiente, processa e transmite a informação para o Coordenador que, por sua vez, confirma o recebimento da mensagem através do envio de um ACK para o nodo i . Cada dispositivo possui sua tolerância em perder algumas mensagens, conhecida em tempo de projeto, representadas na forma de restrições (m,k)-firm.

O principal problema tratado neste trabalho é como os dispositivos que compõem a PAN controlam dinamicamente suas prioridades buscando minimizar o número de ocorrências de falhas dinâmicas. Ou seja, o objetivo maior é respeitar as restrições de qualidade de serviço (m,k) de cada nodo i .

Formalmente, nossa abordagem pode ser descrita como segue. Um conjunto de N tarefas periódicas independentes pode ser representado como $\Gamma = \{\tau_1, \tau_2, \dots, \tau_n\}$. Cada tarefa é não preemptiva e caracterizada por uma 5-tupla $(T_i, D_i, C_i, m_i, k_i)$, onde T_i representa o período, D_i representa o deadline, C_i representa o tempo de execução e m_i e k_i representam as restrições (m,k)-firm da tarefa i ($1 \leq m_i \leq k_i$).

No caso específico no nosso modelo, uma tarefa periódica representa uma aplicação executando em um dispositivo da rede onde os períodos das tarefas são indicados pela periodicidade do superquadro. Neste trabalho, considera-se o deadline relativo igual ao período das tarefas. Por exemplo, uma tarefa que não consegue realizar sua transmissão dentro do superquadro atual, é considerado como um *deadline* perdido. O tempo de computação da tarefa na realidade representa o tempo de envio da mensagem. Finalmente, conforme previamente mencionado, m_i e k_i representam as restrições (m,k)-firm para a tarefa τ_i , o qual ao menos m de qualquer janela de k instâncias consecutivas de um tarefa τ_i devem ter seus *deadlines* atendidos, buscando evitar uma condição de falha dinâmica (uma falha dinâmica ocorre quando uma restrição (m,k)-firm é violada).

Considerando p como o período do superquadro (BI) composto pelo CAP e pelo período inativo, assume-se que: $T_i = p$: o período das tarefas coincidem com o período do superquadro; $D_i = T_i$: *deadline* é igual ao período; e m_i e $k_i: \in \mathbf{Z}^+$, $1 \leq m \leq k$.

4. Algoritmo Distributed DBP

O algoritmo Distributed DBP é construído sobre o conceito distância para falha, o qual foi introduzido por [Hamdaoui and Ramanathan 1995]. Para tornar-se factível a aplicação deste conceito, faz-se necessário a manutenção de um histórico de atendimento recente de cada tarefa. 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. A cada fim 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.

Uma tarefa j pode estar em um dos 2^{k_j} estados possíveis, e para esta calcular a distância para falha $d_j(k)$ basta usar o seguinte procedimento [Hamdaoui and Ramanathan 1995]: Seja $met_j(n, h)$ a posição (à direita) referente ao n^{th} *deadline* atendido no histórico h . Se existem menos que n 1s em h , então $met_j(n, h) = k + 1$. Por exemplo, $met_j(1, “011”) = 1$, $met_j(1, “010”) = 2$, $met_j(2, “101”) = 3$, $met_j(2, “001”) = 4$. Desta forma, utilizando esta função, a distância para falha é denotada por: $d_j(k) \leftarrow (k - met_j(m, h) + 1)$.

Existem alguns trabalhos que propõem algoritmos de escalonamento considerando restrições (m,k)-firm das tarefas [Hamdaoui and Ramanathan 1995, West and Zhang 2004]. No entanto, esses trabalhos sempre consideram uma entidade central – um escalonador – que conhece o histórico de todas as tarefas e toma decisões baseadas em um conhecimento global do sistema. Este trabalho, por sua vez, tem interesse na investigação de algoritmos de escalonamento dinâmicos (*on-line*), onde o escalonamento é realizado considerando as condições locais de cada nodo. A Figura 2 delinea 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.

No algoritmo Distributed DBP, após o envio de uma mensagem endereçada ao Coordenador (linha 1 na Figura 2), a heurística verifica o recebimento da mensagem pelo Coordenador PAN através de uma mensagem de reconhecimento ACK, sinalizando o atendimento de uma *deadline* (linha 4). Caso um dispositivo não consiga efetuar sua

```

1  envia_mensagem();
2
3  evento(status) {
4      if(status == SUCESSO) atendeu = TRUE;
5      if(status == FALHA_CSMA || status == SEM_ACK) atendeu = FALSE;
6
7      distancia_falha = atualiza_distancia_falha(atendeu);
8      altera_prioridade(distancia_falha);
9  }

```

Figura 2. Algoritmo Distributed DBP (executado em cada nodo).

transmissão dentro do atual superquadro, ou se uma mensagem de reconhecimento ACK não é recebida dentro de um intervalo de tempo determinado pelo dispositivo que transmitiu a mensagem, uma ocorrência de não atendimento de *deadline* é sinalizada (linha 5). Posteriormente, a distância para falha dos dispositivos é atualizada baseada na situação de atendimento dos *deadlines* (variável “atendeu”) e a prioridade do dispositivo é atualizada. Caso um dispositivo encontre-se próximo a ocorrência de uma falha dinâmica ($d_j(k) = 1$), este recebe a maior prioridade do sistema, ou seja, o valor da variável *macMinBE* é atualizada para 0, aumentando desta forma a probabilidade de sucesso na transmissão no próximo superquadro; caso contrário $macMinBE = 3$.

5. Resultados

Os estudos foram conduzidos através de análise experimental a partir de um protótipo. Utilizou-se o OpenZB (<http://www.open-zb.net/>), o qual implementa a pilha de protocolos especificada pelo padrão IEEE 802.15.4, desenvolvida em NesC sobre o sistema operacional TinyOS e é compatível com os dispositivos MicaZ (<http://www.xbow.com>).

Utilizou-se 6 nodos MicaZ trabalhando com taxa de transmissão de 250 kbps e frequência de 2.4 GHz, sendo um deles nomeado Coordenador PAN, 3 nodos transmitindo dados periodicamente para o Coordenador PAN contendo restrições (m,k), e outros dois nodos transmitindo constantemente, buscando sobrecarregar o meio de comunicação.

A estrutura do superquadro foi configurada através dos valores $BO = 5$ e $SO = 3$. A periodicidade das tarefas (BI) e o período ativo (SD) foram fixados em aproximadamente 500 ms e 120 ms, respectivamente. As restrições (m,k) para os três nodos foram fixados em (1,3)-firm. Cada nodo transmite uma mensagem de 4 bytes dentro do superquadro e solicita ainda uma mensagem de reconhecimento ACK ao Coordenador. Os três nodos iniciam a transmissão de mensagens após o recebimento de um quadro de *beacon* após 40 ms, 60 ms e 80 ms, respectivamente. Foram utilizados dois nodos transmitindo mensagens (*broadcast*) periodicamente para os outros nodos que compõem a PAN. O tempo de cada experimento foi de aproximadamente 450 segundos (foram geradas aproximadamente um total de 900 mensagens).

No primeiro cenário de experimentos, as mensagens de sobrecarga foram transmitidas com periodicidade de 20 ms. A Figura 3 apresenta o comportamento do sistema nessas condições. Objetivando submeter o sistema a uma condição de sobrecarga severa, em um segundo cenário de experimentação considerou-se transmissões com periodicidade de 7 ms por parte dos nodos que geram carga de fundo (Figura 4).

Observa-se nas tabelas das figuras 3 e 4 que o número de falhas dinâmicas

Exp.	Total de mensagens	Perdas de <i>deadlines</i>	Falhas dinâmicas	Exp.	Total de mensagens	Perdas de <i>deadlines</i>	Falhas dinâmicas
1	905	36,7%	7,3%	1	764	26,7%	1,1%
2	860	38,0%	7,2%	2	846	19,1%	0,9%
3	936	55,5%	18,9%	3	797	21,4%	1,1%

(a) Abordagem convencional

(b) Distributed DBP

Figura 3. Resultados dos experimentos com periodicidade de 20ms.

Exp.	Total de mensagens	Perdas de <i>deadlines</i>	Falhas dinâmicas	Exp.	Total de mensagens	Perdas de <i>deadlines</i>	Falhas dinâmicas
1	881	43,8%	11,0%	1	857	32,0%	1,4%
2	832	31,1%	6,0%	2	856	28,6%	1,5%
3	894	26,1%	7,2%	3	941	29,0%	1,5%

(a) Abordagem convencional

(b) Distributed DBP

Figura 4. Resultados dos experimentos com periodicidade de 7ms.

foram consideravelmente maiores na abordagem convencional oferecida na especificação padrão, do que quando utilizada a abordagem Distributed DBP.

6. Conclusões

Este trabalho apresentou uma abordagem descentralizada para atribuição de prioridades em RSSFs, buscando reduzir o número de ocorrências de falhas dinâmicas nos nodos. Resultados experimentais apresentados apontam redução no número de falhas dinâmicas nos nodos que utilizam o Distributed DBP em contraste com a abordagem tradicional, prevista pelo padrão. A abordagem pode ser facilmente adaptada em dispositivos IEEE 802.15.4, necessitando apenas de mínimas alterações.

Importante destacar que os resultados foram obtidos em um cenário bem específico (mensagens curtas, poucos nodos). Há necessidade de investigar o comportamento da abordagem proposta frente a outros cenários submetido a outras cargas.

Agradecimento Os autores gostariam de agradecer a CAPES e a FAPESC (contrato 4519/2008-5) pelo apoio ao desenvolvimento deste trabalho.

Referências

- Hamdaoui, M. and Ramanathan, P. (1995). A dynamic priority assignment technique for streams with (m,k)-firm deadlines. *IEEE Trans. Computers*, 44(12):1443–1451.
- Hespanha, J. P., Naghshtabrizi, P., and Xu, Y. (2007). A survey of recent results in networked control systems. *Proc. of the IEEE*, 95(1):138–162.
- IEEE (2006). 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*.
- Koren, G. and Shasha, D. (1995). Skip-over: Algorithms and complexity for overloaded systems that allow skips. *Proc. Real-Time Systems Symp.*, pages 110–117.
- Koubaa, A., Alves, M., Nefzi, B., and Song, Y. (2006). Improving the ieee 802.15.4 slotted csma/ca mac for time-critical events in wireless sensor networks. *Proc. of the 5th Int. Workshop on Real-Time Networks*, pages 35–40.
- West, R. and Zhang, Y. (2004). Dynamic window-constrained scheduling of real-time streams in media servers. *IEEE Trans. on Computers*, 53(6):744–759.