

# UMA ARQUITETURA DE SENSORES INTELIGENTES DISPARADA POR EVENTOS: UMA SOLUÇÃO BASEADA NO PADRÃO IEEE 802.15.4

ERICO MENESES LEÃO\*, LUIZ AFFONSO GUEDES\*, FRANCISCO VASQUES†

\**Departamento de Engenharia da Computação e Automação - CT - UFRN  
59078-900, Natal, RN, Brasil*

†*Departamento de Engenharia Mecânica - FEUP - Universidade do Porto  
Rua Dr. Roberto Frias, 4200-465 Porto, Portugal*

Emails: ericoleao@dca.ufrn.br, affonso@dca.ufrn.br, vasques@fe.up.pt

**Abstract**— A smart transducer is the integration of a sensor or an actuator element, a processing unit, and a network interface. Smart sensor networks are composed of smart transducer nodes interconnected through of a communication network. This work proposes a smart sensor network architecture driven by events (asynchronous data) and your respective implementation based in the IEEE 802.15.4 standard. The events are derived from the implementation of a data compression algorithm embedded in the smart sensor. The proposed architecture also provides configuration and monitoring activity of all distributed system.

**Keywords**— smart sensor, network, architecture, events, IEEE 802.15.4, compression algorithm.

**Resumo**— Um transdutor inteligente é a integração de um sensor analógico ou digital ou um atuador, uma unidade de processamento e uma interface de rede. Redes de sensores inteligente são constituídas de nós de transdutores inteligente interligados entre si através de uma rede de comunicação. Este trabalho propõe uma arquitetura de rede de sensores inteligente baseada em eventos (dados assíncronos) e sua respectiva implementação baseada no padrão IEEE 802.15.4. Os eventos são provenientes da implementação de um algoritmo de compressão de dados embarcado no sensor inteligente, que passará a enviar apenas seus dados relevantes. A arquitetura proposta fornecerá também atividades de monitoramento e configuração de todo sistema distribuído.

**Keywords**— sensor inteligente, rede, arquitetura, eventos, IEEE 802.15.4, algoritmo de compressão

## 1 Introdução

A automação industrial é fundamental para o ganho de competitividade em todo setor industrial. A partir de um ponto de vista sistêmico, a automação industrial pode ser caracterizada como um conjunto de técnicas capazes de construir sub-sistemas ativos. Estes sub-sistemas devem ter capacidade de interagir com o processo industrial para controle, monitoramento e propósitos de supervisão.

Na área de automação industrial, em poucos anos se passou de uma tecnologia fortemente centralizada para uma abordagem essencialmente distribuída. Os sensores tradicionais baseados no padrão 4-20mA estão sendo substituídos por sensores inteligentes, providos de comunicação, processadores digitais e memória. Neste contexto, um transdutor inteligente é definido como a integração de um sensor analógico ou digital, um elemento atuador, uma unidade de processamento e uma interface de comunicação (Elmenreich, 2006).

Neste caminho, modernas arquiteturas de automação industrial são caracterizadas por usar um conjunto de transdutores inteligentes, usualmente conectados através de barramento de comunicação com propriedades de tempo real. A abordagem distribuída propiciou uma melhora acentuada na flexibilidade e escalabilidade dos processos industriais, porém, trouxe vários novos desafios científicos e tecnológicos, como a previsão de modelos e algoritmos para comunicação de tempo real e

comunicações críticas-seguras, sempre sobre restrições de custos e impactos ambientais.

O objetivo deste artigo é apresentar uma arquitetura de rede de sensores inteligentes disparada por eventos, detalhada em Leão et al. (2007) e com sua respectiva implementação baseada no padrão IEEE 802.15.4 (IEEE 802.15.4 Standard, 2006). Esta arquitetura é baseada no padrão do OMG (*Object Management Group*) de transdutores inteligentes (Kopetz and Wien, 2003). Porém, diferentemente desse padrão, utiliza um mecanismo disparado por eventos assíncronos, alcançado pela implementação de um algoritmo de compressão de dados embarcado no sensor inteligente, o qual irá filtrar os dados do sensor e passará a enviar apenas os dados mais relevantes da massa de dados brutos.

O resto do artigo é organizado da seguinte forma: Seção 2 introduz as propriedades de redes de sensores inteligentes e o padrão de transdutor inteligente adotado pelo OMG. Seção 3 descreve a arquitetura de sensores inteligentes disparada por eventos e suas características. Na Seção 4 uma implementação de um estudo de caso utilizando o padrão IEEE 802.15.4. O artigo é concluído na Seção 5.

## 2 Rede de Sensores Inteligentes

Com o advento dos microcontroladores e a grande disponibilidade de ferramentas e recursos para o processamento de sistemas digitais foi possível in-

troduzir uma elevada capacidade de computação aos transdutores (Chong and Kumar, 2003).

O conceito de redes de sensores inteligentes não está somente ligado à troca de informações de um transdutor para outro, mas também com o compartilhamento e disponibilização das informações em tempo real. O funcionamento dos sensores inteligentes em rede através de interfaces padronizadas torna possível o compartilhamento de informações e recursos de um sistema, a supervisão de um processo como todo, além de acesso e monitoramento remoto das variáveis envolvidas.

O fator mais crítico para projetar um transdutor inteligente é a construção de suas interfaces de forma padronizada para suportar todos os tipos de transdutores (atuais e futuros). Uma interface deve enquadrar-se para um padrão mundial de comunicação (Elmenreich and Obermaisser, 2002).

Elmenreich (2006) aponta os seguintes requisitos para os transdutores inteligentes:

- **Operação de tempo real:** muitas aplicações para transdutores têm necessidades temporais.
- **Gerência complexa:** o transdutor inteligente precisa fornecer meios para gerenciar a complexidade do sistema quando trocar ou conectar uma rede de transdutores.
- **Suporte à manutenção:** sistemas em operação por um período extensivo de tempo usualmente requerem acesso de manutenção para os transdutores.
- **Comportamento determinístico:** um sistema é determinístico se dado um conjunto de entrada sempre conduz à mesma saída.

Um sistema distribuído com um grande número de transdutores inteligentes deve prover uma abordagem genérica para configuração automatizada (funcionalidade de *plug and play*) (Pitzek and Elmenreich, 2003).

### 2.1 Padrão de Transdutor Inteligente do OMG

Em resposta à chamada do *Object Management Group* (OMG) em dezembro de 2000, um novo padrão de transdutor inteligente foi proposto que compreende um serviço de transporte disparado por tempo dentro de uma rede de transdutores inteligentes distribuídos e uma interface bem definida para um ambiente CORBA, possibilitando atividades de monitoramento (Elmenreich, 2006; Elmenreich and Obermaisser, 2002). Este padrão foi adotado pelo OMG em janeiro de 2002 (Kopetz and Wien, 2003).

O padrão do OMG consiste de vários *clusters* com até 250 nós de transdutores inteligentes conectados em um barramento. Cada *cluster* é ligado em um *gateway* CORBA via um nó mestre,

responsável por controlar a comunicação entre os transdutores.

O acesso aos dados dos transdutores inteligentes no padrão do OMG é feito atribuindo três interfaces diferentes a cada nó (Kopetz and Wien, 2003; Elmenreich, 2006): interface de diagnóstico e gerência, interface de planejamento e configuração e interface de serviço de tempo real.

As transferências de informações entre um transdutor e seu cliente são alcançadas por informações compartilhadas que são contidas em um sistema de arquivos da interface (*Interface File System*), o qual é encapsulado dentro de cada transdutor inteligente (Elmenreich and Obermaisser, 2002).

## 3 Arquitetura de Sensores Inteligentes Disparada por Eventos

A arquitetura de sensores inteligentes disparada por eventos (Leão et al., 2007) é baseada no padrão do *Object Management Group* (OMG), devido sua simplicidade na definição das interfaces de acesso aos dados dos transdutores inteligentes. Porém, diferentemente do padrão do OMG, a abordagem desta arquitetura é orientada a eventos gerados pelos sensores inteligentes, que enviarão apenas seus dados mais relevantes de forma assíncrona.

A arquitetura é formada de vários nós de transdutores inteligentes (nós escravos) ligados ao mesmo barramento, formando um *cluster*. Cada *cluster* possui um nó mestre de maior capacidade de processamento que tem a finalidade de gerenciar os nós de sensores inteligentes. Como suporte de tolerância a falhas, podem existir nós mestres secundários redundantes caso o nó mestre principal falhe. Os *clusters* são conectados a uma rede supervisória através de seus nós mestres, podendo realizar atividades de configuração dos sensores, bem como monitoramento de todo o sistema, como pode ser visualizado na Figura 1.

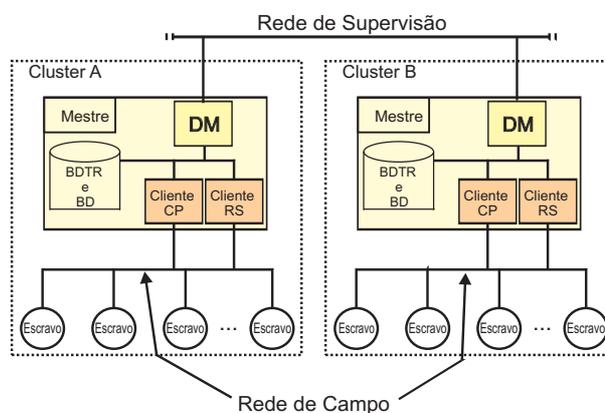


Figura 1: Arquitetura de Sensores Inteligentes Disparada por Eventos.

### 3.1 Estrutura do Nó Mestre

O nó mestre é composto de dois tipos de banco de dados: um banco de dados de tempo real (DBTR) e um banco de dados tradicional (BD). Os BDTR além de processar transações e garantir a integridade dos dados, características fundamentais em banco de dados tradicionais, devem também satisfazer às restrições temporais impostas pelas transações e dados, que podem ser considerados válidos unicamente por um intervalo de tempo específico (Ramamritham, 1993). O nó mestre é responsável por armazenar em seus bancos de dados as informações mais relevantes dos sensores inteligentes presentes no *cluster* para as atividades de monitoramento, além de possuir a capacidade de reconstruir os dados compactados pelo sensor para as atividades de controle.

Através da interface de diagnóstico e manutenção é possível o acesso aos bancos de dados tradicional e de tempo real do nó mestre através da rede supervisória. Através do cliente de RS (tempo real) e cliente CP (configuração e planejamento) é possível a comunicação com as interfaces dos sensores inteligentes.

### 3.2 Estrutura do Nó Escravo

Os nós de sensores inteligentes possuem duas interfaces para acesso aos seus dados, um algoritmo de compressão embarcado e um *buffer* para armazenar dados temporários para aplicação da compressão dos dados, como ilustrado na Figura 2. O algoritmo de compressão dentro do sensor é responsável por filtrar seus dados mais relevantes, fornecendo, assim, uma característica assíncrona na geração dos dados.

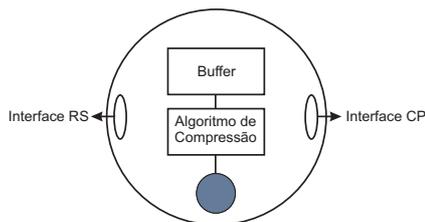


Figura 2: Transdutor Inteligente com um algoritmo de Compressão embarcado.

Com uma maior autonomia fornecida ao sensor inteligente, que passará a enviar apenas dados relevantes, haverá uma grande redução dos dados que serão transmitidos pela rede de comunicação, resultando em uma menor utilização de banda. Assim, pode-se conectar uma maior quantidade de sensores inteligentes à rede. Porém, eles terão um aumento de complexidade e demandarão maior capacidade de processamento. Contudo, com o crescente avanço tecnológico é possível construir sensores inteligentes de baixo-custo com grande

capacidade de processamento. Com isso, o algoritmo de compressão implementado dentro do sensor torna-se o ponto chave da arquitetura de rede de sensores inteligentes disparada por eventos.

Os sensores possuem dois tipos de interface para acesso aos dados (RS e CP), onde a comunicação é suportada por dois modelos de comunicação diferentes (*publisher-subscriber* e cliente-servidor):

- **Interface RS** - interface de serviço de tempo real. É utilizada para o transporte de dados com requisitos de tempo real dentro do cluster. Os dados gerados pelos sensores inteligentes serão publicados na rede e consumidos pelas funções do sistema.
- **Interface CP** - interface de configuração e planejamento. Através desta interface é possível configurar novos nós ligados à rede, passando parâmetros de configuração para o sensor, como valores de desvio de compressão, tempo mínimo e tempo máximo para o algoritmo de compressão, além de informações como identificação do sensor no sistema distribuído.

#### 3.2.1 Modelo *Publisher-Subscriber*

O mecanismo de comunicação utilizado pela interface RS é o *publisher-subscriber de tempo real* (PSTR). O modelo de troca de dados PSTR permite a troca de mensagens com parâmetros de tempo entre dispositivos através de duas entidades: o *publisher*, responsável pelo envio das mensagens, e os *subscribers*, responsáveis por consumir essas mensagens caso tenha interesse (Dolejs et al., 2004; Ocera, 2002). As mensagens enviada pelo nó escravo através da interface RS serão consumidas pelas várias funções do sistema que tiverem interesse nos dados. Estas mensagens podem ser enviadas como comandos para atuadores ou então armazenadas em bancos de dados de acordo com seus requisitos, como pode ser visualizado na Figura 3.

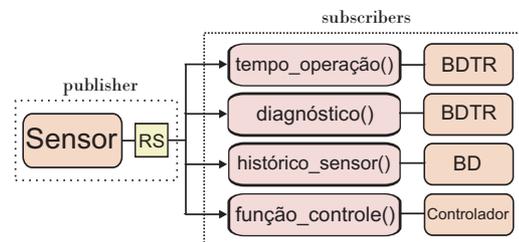


Figura 3: Modelo *Publisher-Subscriber* para a Arquitetura Disparada por Eventos.

Os pacotes de dados enviados pelos sensores inteligentes através da interface RS contém vários campos com informações importantes dos sensores inteligentes, que serão consumidos pelas

funções do sistema para diversas finalidades. Por exemplo, a função *tempo\_operação()* irá acessar o campo TEMPO e armazenar esse dado no banco de dados de tempo real para fornecer à rede supervisória o tempo de funcionamento de um determinado sensor inteligente, enquanto que a função *historico\_sensor()* irá acessar as informações no campo DADO afim de disponibilizar o histórico de marcações de um sensor inteligente. O campo ID do pacote é responsável pela identificação do sensor, o campo TIPO pela identificação do tipo do pacote a ser transmitido e o campo QUALIDADE conterá informações da qualidade da marcação do dado (bom, regular, ruim, não determinado), fornecido pelo sensor inteligente. O campo CRC é responsável pelo controle de erro do pacote. A estrutura do pacote enviado pelo sensor é ilustrado na Figura 4.



Figura 4: Estrutura de uma mensagem do nó escravo.

### 3.2.2 Modelo Cliente-Servidor

A interface CP é acessada diretamente pelo nó mestre, que pode realizar atividades de configuração, passagem de parâmetros e reconfiguração de nós da rede, como pode ser visto na Figura 5. Para um exemplo de uma configuração de um novo nó ligado a rede através da interface CP, o nó mestre através da função *novo\_dispositivo()* fica monitorando o sistema, afim de buscar novos nós ligados a rede. Assim, o novo sensor inteligente irá solicitar sua inclusão no sistema e receberá do nó mestre, através da função *configura()*, os parâmetros necessários para o seu correto funcionamento na rede. A partir deste ponto, após confirmar o recebimento dos parâmetros (*confirma()*) e receber do nó mestre uma autorização para transmitir seus dados (*iniciar()*), o sensor começará a enviar seus pacotes de dados através da interface RS. A Figura 6 ilustra esse procedimento. Este procedimento fornece a característica de *plug-and-play* ao sistema.

Algumas funções importantes do sistema são definidas a seguir:

- *requer\_inclusão()*: através da interface CP o nó escravo requisita ao mestre sua inclusão na rede.
- *configura()*: através da interface CP o nó mestre envia os parâmetros de configuração a um novo nó escravo.
- *checar\_sensor()*: através da interface CP o nó mestre verifica se um ou mais nós escravos estão ativos.

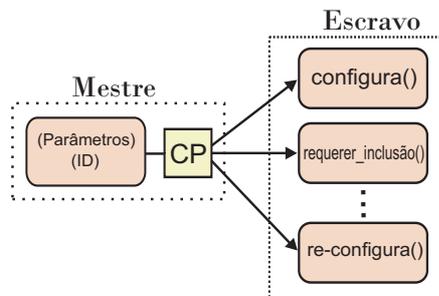


Figura 5: Modelo *Cliente-Servidor* para a Arquitetura Disparada por Eventos.

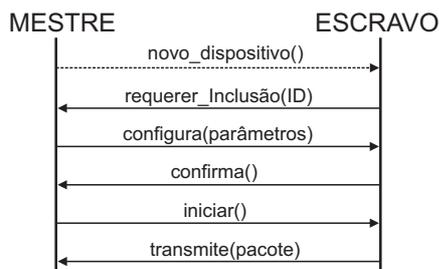


Figura 6: Procedimento de Inclusão de um novo nó.

- *transmite()*: através da interface RS o nó escravo envia um determinado pacote para a rede.
- *tempo\_operação()*: através da interface RS verifica-se o tempo de funcionamento de um sensor.

## 4 Estudo de Caso

Esta seção apresenta a implementação de um estudo de caso da arquitetura de rede de sensores inteligentes disparada por eventos utilizando o padrão IEEE 802.15.4 (IEEE 802.15.4 Standard, 2006) como infraestrutura de comunicação.

### 4.1 Implementação baseada no padrão IEEE 802.15.4

O padrão IEEE 802.15.4 define a camada física e de acesso ao meio para dispositivos de baixo custo, com limitações de energia e que enviam dados a baixas taxas (Baronti et al., 2007).

Para nossa implementação foi utilizado as placas de comunicação sem fio do kit de desenvolvimento da *freescale* - MC13192EVK, que utiliza o padrão IEEE 802.15.4 (Figura 7). O software desenvolvido nas placas foi baseado nas premissas do *SMAC* (*Simple MAC*) do kit de desenvolvimento.

O *SMAC* é uma linguagem simples de programação *C ANSI* que pode ser usada para desenvolver aplicações proprietárias para os *transceivers* utilizando circuitos integrados dos módulos MC13192X existentes nas placas sem fio

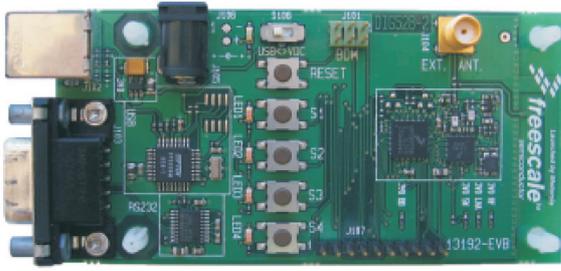


Figura 7: Placa ZigBee da Freescale.

do kit de desenvolvimento da *freescale*. Este protocolo foi desenvolvido para trabalhar com qualquer processador HCS08 realizando comunicação entre ele e o *transceiver* através do módulo SPI (*Serial Peripheral Interface*).

Dessa forma, para o nó escravo foi implementado o algoritmo de compressão de dados *Swinging Door* (Bristol, 1990) embarcado na placa de comunicação sem fio.

O princípio básico do algoritmo de compressão *Swinging Door* é colocar o primeiro dado marcado como ponto inicial do primeiro intervalo de compressão. O ponto final deste intervalo de compressão também atua como ponto inicial do próximo intervalo de compressão (Xiaodong et al., 2002).

O algoritmo contém três parâmetros principais: o desvio de compressão, o tempo mínimo e o tempo máximo de compressão. O algoritmo define um paralelogramo com largura igual ao dobro do desvio de compressão definido, compreendido entre o ponto inicial e o ponto final.

A partir do ponto inicial se nenhum valor recebido extrapolar o paralelogramo até o ponto final, nenhum ponto intermediário é armazenado. Neste caso, o ponto final será armazenado, ao mesmo tempo que será definido como o novo ponto inicial. Se caso algum valor intermediário extrapolar o paralelogramo e não estiver dentro do tempo mínimo determinado, o valor anterior será armazenado. O tempo mínimo de compressão serve para filtrar sinais ruidosos.

Assim, a placa escrava recebe via porta serial dados oriundos de um sistema de monitoramento em tempo real de distribuição de gás, especificamente uma variável de vazão, aplica o algoritmo de compressão filtrando os dados mais relevantes e envia para o nó mestre.

O nó mestre implementado, por sua vez, é responsável por receber os dados enviados pelos nós escravos e, via porta serial, enviá-los para um PC industrial e guardá-los em seus bancos de dados. A partir dos bancos de dados do nó mestre é possível ter acesso a todos os dados mais relevantes dos nós escravos ligados a esse nó mestre.

## 4.2 Resultados Obtidos

Foram realizados vários testes com as placas, utilizando diferentes parâmetros de compressão. É importante destacar que o algoritmo *Swinging Door* depende dos parâmetros definidos inicialmente. Assim, as taxas de compressão dos valores dos sensores serão dependentes da configuração estabelecida pelo nó mestre, via interface de configuração, para uma aplicação específica, por exemplo, um sensor de pressão terá configuração de parâmetros diferente de um sensor de temperatura.

Para reconstrução dos dados compactados é utilizado o método de interpolação linear para estimar os dados perdidos com a compressão e recuperar seus tempos.

A Figura 8 mostra o gráfico com os dados originais do sensor inteligente. As Figuras 9 e 10 mostram, respectivamente, os dados reconstruídos depois de uma compressão dos dados originais de 85.72% e 94.40% pelo algoritmo implementado.

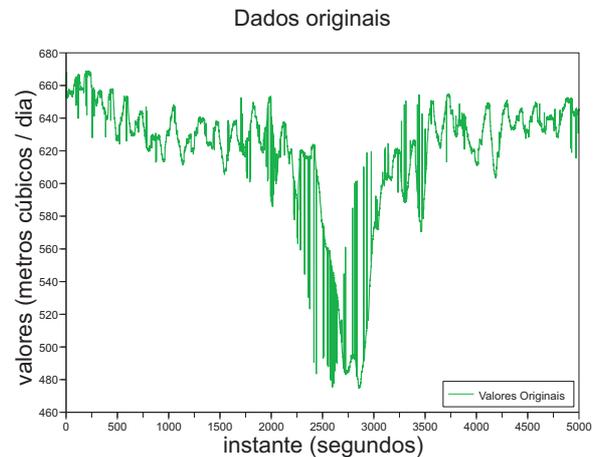


Figura 8: Dados Originais do Sensor.

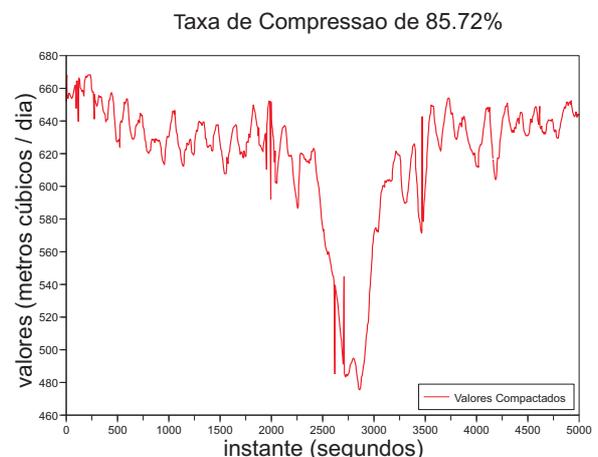


Figura 9: Compressão de Dados de 85.72%.

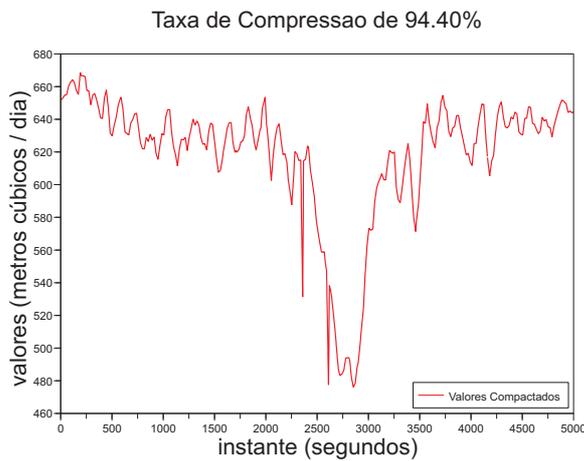


Figura 10: Compressão de Dados de 94.40%.

Analisando os gráficos, podemos visualizar a fidelidade na reconstrução dos dados originais, mesmo com uma compressão elevada de até 94.40%, que mostra a relevância dos dados enviados pelos sensores inteligentes. Com isso, a implementação do algoritmo embarcado no sensor inteligente diminui a quantidade de dados de um sensor trafegados na rede de comunicação.

## 5 Conclusão

O artigo apresenta uma implementação baseada no padrão IEEE 802.15.4 da arquitetura de rede de sensores inteligentes disparada por eventos. A principal característica desta arquitetura é justamente a implementação do algoritmo de compressão de dados embarcado no sensor inteligente. Esta abordagem salva passagem de mensagens através da rede de comunicação, pois os sensores apenas enviam dados relevantes.

Com os resultados alcançados através de experimentos realizados é possível concluir que o processo de compressão local em cada sensor inteligente diminui consideravelmente as trocas de dados na rede de comunicação.

Em trabalhos futuros serão incluídos o controle da comunicação na arquitetura de sensores inteligentes disparada por eventos, usando técnicas de escalonamento com prioridades para as mensagens, com o propósito de evitar possíveis colisões e perdas de pacotes.

## Referências

Baronti, P., Pillai, P., Chook, V., Chessa, S., Gotta, A. and Hu, Y. F. (2007). Wireless sensor networks: A survey on the state of the art and the 802.15.4 and ZigBee standards, *Computer Communications* **30**: 1655–1695.

Bristol, E. H. (1990). Swinging Door Trending: adaptive trend recording?, *ISA National Conf. Proc.*, pp. 749–753.

Chong, C.-Y. and Kumar, S. P. (2003). Sensor networks: Evolution, opportunities, and challenges, *Proceedings of the IEEE*, Vol. 91, pp. 1247–1256.

Dolejs, O., Smolik, P. and Hanzalek, Z. (2004). On the ethernet use for real-time publish-subscribe based applications, *Proceedings. 2004 IEEE International Workshop on Factory Communication Systems*, pp. 39–44.

Elmenreich, W. (2006). Time-triggered smart transducer networks, *IEEE Transactions on Industrial Informatics* **2**: 192–199.

Elmenreich, W. and Obermaisser, R. (2002). A standardized smart transducer interface, *Proceedings of the 2002 IEEE International Symposium on Industrial Electronics*, Vol. 1, pp. 164–169.

IEEE 802.15.4 Standard (2006). Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs), *Specification*.

\*<http://standards.ieee.org/getieee802/>

Kopetz, H. and Wien, T. (2003). Omg smart transducer specification II, *Specification*.

\*<http://doc.omg.org/formal/2003-01-01/>

Leão, E. M., Guedes, L. A. and Vasques, F. (2007). An Event-Triggered Smart Sensor Network Architecture, *5th IEEE International Conference on Industrial Informatics (INDIN 2007)*, Vienna, Austria.

Ocera (2002). Wp2 - architecture specification. deliverable d2.1 - architecture and components integration.

\*<http://mnis.fr/en/support/doc/architecture>

Pitzek, S. and Elmenreich, W. (2003). Configuration and management of a real-time smart transducer network, *IEEE Conference Emerging Technologies and Factory Automation, ETFA 2003*, Vol. 1, pp. 407–414.

Ramamritham, K. (1993). Real-Time Databases, *International Journal of Distributed and Parallel Databases* **1**: 199–226.

Xiaodong, F., Changling, C., Changling, L. and Huihe, S. (2002). An improved process data compression algorithm, *Proceedings of the 4th World Congress on Intelligent Control and Automation*, Vol. 3, pp. 2190–2193.