

Lucas Bortolaso Torri

*A norma IEEE 1451 aplicada a redes
heterogêneas de sensores sem fio*

Florianópolis – SC

Outubro / 2008

Lucas Bortolaso Torri

*A norma IEEE 1451 aplicada a redes
heterogêneas de sensores sem fio*

Trabalho de conclusão de curso apresentado
como parte dos requisitos para obtenção do
grau Bacharel em Ciências da Computação.

Orientador:

Prof. Dr. Antônio Augusto M. Fröhlich

UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
BACHARELADO EM CIÊNCIAS DA COMPUTAÇÃO

Florianópolis – SC

Outubro / 2008

Trabalho de conclusão de curso sob o título “A norma IEEE 1451 aplicada a redes heterogêneas de sensores sem fio”, defendido por Lucas Bortolaso Torri sob aprovação, em Florianópolis, Santa Catarina, pela banca examinadora constituída:

Prof. Dr. Antônio Augusto M. Fröhlich
Departamento de Informática e Estatística
Orientador

Banca Examinadora

Prof. Dr. Carlos Barros Montez
Departamento de Automação e Sistemas

Dr. Daniel T. Cobra
Fundação CERTI

Prof. Dr. Mário Antônio Ribeiro Dantas
Departamento de Informática e Estatística

“It’s a magical world, Hobbes, ol’ buddy...Let’s go exploring!”

Calvin, Calvin and Hobbes

Bill Watterson

Agradecimentos

De forma sucinta, gostaria de agradecer a todos que me ajudaram, de forma direta ou não, para a realização deste trabalho.

Ao professor Guto agradeço não só pela atenção dada, mas principalmente pela confiança depositada durante o desenvolvimento deste projeto. Aos membros da banca agradeço por participarem da avaliação deste trabalho, principalmente ao Daniel, que me ajudou a definir a escolha do tema. Igualmente, agradeço ao meu companheiro de apartamento, Wagner, por ser paciente e amigo, e ao colega Bruno, pela nossa amizade e por todas as conquistas que realizamos.

Agradeço ainda toda minha família. À Isadora, namorada, companheira e melhor amiga. Ao meu irmão Bruno, que é o melhor presente que já recebi. Aos meus pais, Paulo e Nice, que sempre acreditaram no seu primogênito. Isa, Bruno, Mãe e Pai, agradeço vocês de todo o coração, por sua ajuda e compreensão, pelos momentos proporcionados, mas acima de tudo, por serem sempre meus exemplos e motivos de orgulho.

Obrigado!

Resumo

Devido às vantagens oferecidas pela sua utilização, as Redes de Sensores sem Fio têm atualmente uma importância de destaque na área tecnológica. Redes de Sensores sem Fio, ou RSSF, são agrupamentos de dispositivos, conhecidos como nós sensores, com a capacidade de comunicação por radiofrequência e dotados de sensores capazes de monitorar os fenômenos acontecidos no ambiente à sua volta. No entanto, devido às inúmeras plataformas disponíveis para a construção destas redes, cada uma com seus atributos de hardware e software, o fator de heterogeneidade se torna um ponto vulnerável nas configurações das RSSF. A fim de atenuar este problema nos sistemas dotados de transdutores, foi criada a família IEEE 1451, formada por diferentes normas com o intuito de definir um conjunto de interfaces para conectar transdutores a sistemas e instrumentos de uma maneira independente de redes de comunicação, permitindo a auto-identificação dos transdutores e a interoperabilidade entre estes e a rede em que se encontram.

Esse trabalho apresenta um estudo feito sobre a família IEEE 1451 e a sua aplicação a redes de sensores sem fio, de forma a contemplar as restrições apresentadas pelas RSSF e permitir que os nós sensores, conhecidos na família IEEE 1451 como WTIMs, sejam os responsáveis por tomar a iniciativa nas comunicações com o gateway de rede.

Palavras-chave: Redes de Sensores sem Fio, RSSF, IEEE 1451, WTIM

Abstract

Given the advantages offered by its use, Wireless Sensors Networks take on an important role in present day technology. Wireless Sensor Networks, or WSNs, are groups of devices, known as motes, with the ability to communicate using radio frequency signals and equipped with sensors capable of monitoring the phenomena occurring in the surrounding environment. However, given the diversity of available platforms for building such networks, each with its features of hardware and software, the factor of heterogeneity becomes a point of vulnerability in the configurations of WSNs. In order to attenuate this problem in transducer-equipped systems, the IEEE 1451 family was created, which is formed by different standards that define a set of interfaces for connecting transducers to different systems and instruments in a communication network independent manner, enabling the self identification of transducers and the interoperability between them and the networks where they are.

This paper presents a study about the IEEE 1451 family and its application in wireless sensor networks, addressing the constraints presented by WSNs and enabling motes, known in the 1451 family as WTIMs, to take the initiative of communicating with the network gateway.

Key-words: Wireless Sensor Networks, WSN, IEEE 1451, WTIM

Sumário

Lista de Figuras

Lista de Tabelas

| | |
|---|-------|
| Lista de acrônimos e abreviações | p. 11 |
| 1 Introdução | p. 14 |
| 1.1 Objetivo | p. 16 |
| 1.1.1 Objetivos Específicos | p. 16 |
| 1.2 Organização do Trabalho | p. 17 |
| 2 Família IEEE 1451 | p. 18 |
| 2.1 IEEE 1451.0 | p. 20 |
| 2.1.1 TransducerChannel | p. 21 |
| 2.1.2 Modos de Operação | p. 23 |
| 2.1.3 Especificação do TEDS | p. 26 |
| 2.1.4 Mensagens e Comandos | p. 28 |
| 2.1.5 APIs | p. 29 |
| 2.2 IEEE 1451.5 | p. 29 |
| 2.2.1 PHY TEDS | p. 31 |
| 2.3 Implementações existentes | p. 32 |
| 3 Nós Sensores | p. 33 |
| 3.1 Gateway | p. 34 |

| | | |
|----------|--|--------------|
| 3.2 | Sun SPOT | p. 34 |
| 3.2.1 | eDemo Board | p. 35 |
| 3.3 | Crossbow MICA | p. 37 |
| 3.3.1 | MTS300 | p. 39 |
| 3.3.2 | EPOS | p. 39 |
| 4 | Desenvolvimento | p. 42 |
| 4.1 | Operação dos WTIMs | p. 44 |
| 4.2 | Construção dos TEDS | p. 45 |
| 4.2.1 | Meta-TEDS | p. 46 |
| 4.2.2 | TransducerChannel TEDS | p. 47 |
| 4.2.3 | User's Transducer Name TEDS | p. 50 |
| 4.2.4 | PHY TEDS | p. 53 |
| 4.3 | Implementação dos WTIMs | p. 54 |
| 4.4 | Implementação NCAP | p. 59 |
| 4.4.1 | Aplicações NCAP | p. 62 |
| 4.5 | Módulos de Comunicação sem fio | p. 64 |
| 4.6 | Servidor de Dados | p. 65 |
| 5 | Conclusão | p. 68 |
| 6 | Trabalhos Futuros | p. 70 |
| | Referências Bibliográficas | p. 71 |
| | Apêndice A – Artigo | p. 76 |

Lista de Figuras

| | | |
|-----|---|-------|
| 1.1 | Arquitetura de uma RSSF(AKYILDIZ et al., 2002b) | p. 15 |
| 2.1 | Relação das normas pertencentes à família IEEE 1451 | p. 20 |
| 2.2 | Modelo de referência proposto pela IEEE 1451.0(IEEE. . . , 2007) | p. 22 |
| 2.3 | Modelo alternativo de referência proposto pela IEEE 1451.0(IEEE. . . , 2007) | p. 22 |
| 2.4 | Blocos lógicos de um TransducerChannel(IEEE. . . , 2007) | p. 23 |
| 2.5 | Contexto funcional da IEEE 1451.5(IEEE. . . , 2007) | p. 30 |
| 2.6 | Configuração típica da IEEE 1451.5(IEEE. . . , 2007) | p. 31 |
| 3.1 | Blocos básicos de um nó sensor(KHEMAPECH; DUNCAN; MILLER, 2005) | p. 34 |
| 3.2 | Nó sensor Sun SPOT | p. 36 |
| 3.3 | Nó sensor Crossbow MICA | p. 38 |
| 3.4 | Visão geral do projeto EPOS(FROHLICH, 2001) | p. 40 |
| 4.1 | Arquitetura definida para a implementação das normas IEEE 1451 | p. 43 |
| 4.2 | Diagrama UML do WTIM implementado no Crossbow MICA | p. 57 |
| 4.3 | Diagrama UML do WTIM implementado no Sun SPOT | p. 58 |
| 4.4 | Diagrama UML do NCAP implementado | p. 61 |
| 4.5 | Diagrama UML das aplicações NCAP | p. 63 |
| 4.6 | Página exibindo TIMs conectados ao Servidor | p. 66 |
| 4.7 | Gráfico gerado pelo servidor Web com as leituras de um sensor de tem- peratura | p. 67 |

Lista de Tabelas

| | | |
|-----|---|-------|
| 2.1 | Formato padrão dos TEDS | p. 27 |
| 2.2 | Definição da estrutura TLV | p. 27 |
| 2.3 | Estrutura da TLV de identificação dos TEDS | p. 28 |
| 3.1 | Hardware do Sun SPOT | p. 36 |
| 3.2 | Hardware do Crossbow MICA | p. 38 |
| 4.1 | Estrutura da mensagem iniciada pelo TIM para envio das leituras | p. 45 |
| 4.2 | Campos obrigatórios do Meta-TEDS | p. 47 |
| 4.3 | Octetos do Meta-TEDS | p. 47 |
| 4.4 | Campos utilizados no TransducerChannel TEDS de um sensor | p. 50 |
| 4.5 | Octetos do TransducerChannel TEDS de um sensor de temperatura | p. 51 |
| 4.6 | Campos obrigatórios do User's Transducer Name TEDS | p. 52 |
| 4.7 | Octetos do User's Transducer Name TEDS do sensor de temperatura | p. 52 |
| 4.8 | Campos obrigatórios do PHY TEDS | p. 55 |
| 4.9 | Octetos do PHY TEDS do Crossbow MICA | p. 56 |

Lista de acrônimos e abreviações

A

ADC Analog to Digital Converter

API Application Programming Interface

C

CLDC Connected Limited Device Configuration

D

Dot5AR IEEE 1451.5 approved radios

E

EPOS Embedded Parallel Operating System

H

HTML HyperText Markup Language

HTTP HyperText Transfer Protocol

I

IDE Integrated Development Environment

IEEE Institute of Electrical and Electronics Engineers

I/O Input/Output

I

JDDAC Java Distributed Data Acquisition and Control

L

LED Light Emitting Diode

LSB Least Significant Byte

M

MAC Media Access Control

ME Mobile Edition

MSB Most Significant Byte

N

NCAP Network Capable Application Processor

NIST National Institute of Standards and Technology

P

PHY Physical layer

R

REST Representational State Transfer

RSSF Redes de Sensores sem Fio

S

SPOT Small Programmable Object Technology

T

TEDS Transducer Electronic Data Sheet

TIM Transducer Interface Module

TLV Type/Length/Value

U

UFSC Universidade Federal de Santa Catarina

URI Uniform Resource Identifier

URL Uniform Resource Locator

USB Universal Serial Bus

UTF Unicode Transformation Format

UUID Universal Unique Identifier

V

VM Virtual Machine

W

WTIM Wireless Transducer Interface Module

X

XML Extensible Markup Language

1 *Introdução*

A miniaturização dos micro-processadores, juntamente com o aumento de sua capacidade computacional e da redução dos seus custos de fabricação, permitiu que os mesmos pudessem ser integrados a diversos dispositivos de baixo custo e consumo de energia (WARNEKE; PISTER, 2002). Este acontecimento estabeleceu uma enorme pervasividade da tecnologia dos computadores em nosso cotidiano, mudando radicalmente a forma de interpretação e interação com o nosso ambiente (WEISER, 1999).

Um dos mercados que se beneficiou com esta evolução do hardware foi a área de redes de sensores sem fio (RSSF), que estuda a utilização de sistemas autônomos formados por dispositivos micro-processados, com a capacidade de se comunicarem através de sinais de rádio-frequência e que possuem sensores capazes de traduzir em formato digital as condições do ambiente físico em que se encontram (TILAK; ABU-GHAZALEH; HEINZELMAN, 2002). Esses dispositivos, conhecidos como nós sensores, podem ser instalados em diferentes lugares de um espaço físico, monitorando os fenômenos que ocorrem nesse meio, permitindo assim, utilizar as informações coletadas para entender e tomar decisões adequadas sobre o ambiente estudado. A figura 1.1 mostra a arquitetura de uma rede de sensores com a capacidade de coletar informações e enviá-las ao usuário final através de um nó com a responsabilidade de agregar os dados.

Segundo a revista *Technology Review*, as RSSF são consideradas uma das 10 tecnologias mais promissoras do mundo (WADE; MITCHELL; PETTER, 2003). O monitoramento através dessa tecnologia foi primeiramente usado para fins militares, mas hoje são empregadas em diversos tipos de aplicações, como por exemplo, monitoramento de habitats, sistemas de observação e previsão do tempo, área da saúde, entre outras aplicações comerciais e civis, tornando essa uma área de natureza multidisciplinar e muito ampla (KHEMAPECH; DUNCAN; MILLER, 2005; XU, 2003; GARCÍA-HERNÁNDEZ et al., 2007; KIM et al., 2007).

As RSSF trazem uma série de novos desafios a serem estudados, já que muitas vezes

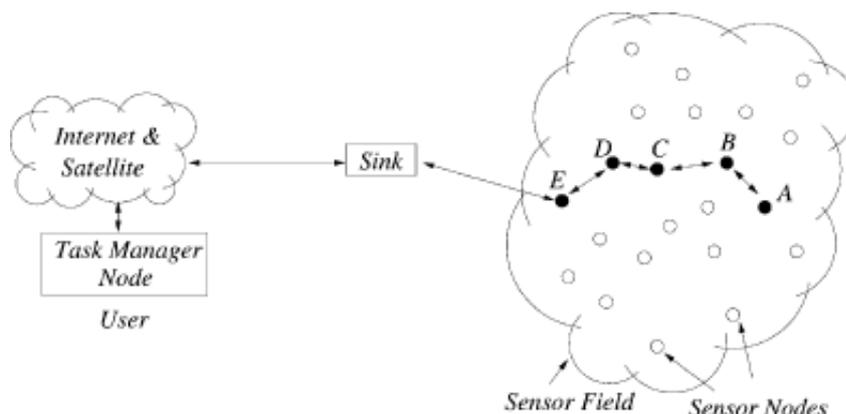


Figura 1.1: Arquitetura de uma RSSF (AKYILDIZ et al., 2002b)

essas redes devem operar por longos períodos de tempo, usando quantidades restritivas de energia, limitando assim a capacidade de processamento e de transmissão de dados (AKYILDIZ et al., 2002a; CULLER; ESTRIN; SRIVASTAVA, 2004). Como o número de aplicações possíveis é muito grande, cada uma com características e necessidades únicas, torna-se difícil especificar requisitos gerais para as redes de sensores. Dessa forma, foram criadas diversas plataformas para facilitar o desenvolvimento e criação das RSSF, além da existência de milhares de fabricantes e modelos de sensores, cada uma introduzindo novos conceitos, abstrações, formas de manipular dados e algoritmos, gerando diversas combinações possíveis de hardware (ROMER; MATTERN, 2004; WANNER, 2003).

Uma vez que o mercado de transdutores, definidos aqui como sensores ou atuadores, é muito diverso e o fator de heterogeneidade é um ponto muito importante no design das redes de sensores, já que ele afeta toda a complexidade do sistema (COULOURIS; DOLLIMORE; KINDBERG, 2005; HADIM; MOHAMED, 2006; ROMER; MATTERN, 2004), fabricantes estão buscando por maneiras de baixo custo para construir sensores e atuadores inteligentes e com capacidade de comunicação sem fio. Atualmente, é demasiado oneroso para os fabricantes criarem transdutores para cada tipo de rede existente no mercado. Além disso, uma vez que as RSSF são constituídas por um grande número de nós sensores, o custo de um único nó é muito importante para justificar o custo global das redes. Se o custo da rede é mais caro do que implantar sensores tradicionais, então a implantação da rede de sensores não é justificada em termos de custos. Devido a esse fato, o custo de cada nó sensor tem de ser mantido baixo (AKYILDIZ et al., 2002b). No entanto, através da utilização de sistemas de comunicação digital, através do uso de microprocessadores e em redes sem fio em transdutores, pode-se eliminar um grande número de fiações analógicas e reduzir assim os custos de instalação, manutenção e atualização

dos sistemas baseados em sensores.

A fim de atenuar o problema de heterogeneidade de sistemas dotados de transdutores, foi criada a família de padrões IEEE 1451 (LEE, 2000). Definidos pelo Instituto Nacional de Padrões e Tecnologia (NIST), agência governamental não-regulatória da Administração de Tecnologia do Departamento de Comércio dos Estados Unidos, em conjunto com o Instituto de Engenheiros Eletricistas e Eletrônicos (IEEE), maior organização profissional do mundo, as normas têm como intuito definir um conjunto de interfaces para conectar transdutores a sistemas e instrumentos de uma maneira independente de redes de comunicação, permitindo a auto-identificação dos transdutores e a interoperabilidade entre sensores/atuidores e as redes em que se encontram, através de capacidade “plug and play”¹. Dessa forma, é possível simplificar o desenvolvimento de dispositivos inteligentes e de controle de múltiplos transdutores através de comandos independentes de rede de comunicação.

1.1 Objetivo

Este trabalho visa a aplicação das normas IEEE 1451 relacionadas à Redes de Sensores sem Fio, de forma a possibilitar a integração de diferentes redes a um mesmo sistema, permitindo reconhecer os sensores disponíveis e a configuração automática dos mesmos, em caso de necessidade.

1.1.1 Objetivos Específicos

- Conhecer a área de Redes de Sensores sem Fio e as tecnologias envolvidas;
- Estudar as normas pertencentes à família IEEE 1451;
- Estudar a integração da IEEE 1451 a redes de sensores sem fio;
- Implementar as normas IEEE 1451 que cabem às RSSF, adequando-as a este tema.

¹Plug and play é uma característica que permite adicionar novos dispositivos periféricos a um sistema. Esses dispositivos são automaticamente reconhecidos e configurados, sem a necessidade de instalação manual dos mesmos.

1.2 Organização do Trabalho

Além deste primeiro capítulo, que faz uma breve introdução do trabalho, são apresentados outros cinco capítulos, que têm o objetivo de introduzir e discutir o tema abordado. Para permitir um melhor entendimento, o segundo e terceiro capítulo fazem uma apresentação dos conceitos utilizados durante o projeto. O segundo capítulo visa descrever rapidamente todas as normas pertencentes à família IEEE 1451, mostrando os detalhes e os conceitos introduzidos por aqueles padrões que se relacionam com redes de sensores sem fio. Esses padrões, por sua vez, serão responsáveis por encaminhar os desenvolvimentos efetuados no projeto. Neste capítulo também são discutidas, de forma sucinta, as características das implementações existentes da IEEE 1451. No capítulo 3, serão apresentados maiores detalhes acerca da organização e funcionamento de nós sensores, mostrando quais os elementos que os compõem e como eles são organizados. Além disso, serão apresentados os nós utilizados neste trabalho, juntamente dos detalhes e características de software e hardware presentes em cada um deles.

O quarto capítulo expõe todo o desenvolvimento efetuado no trabalho, apresentando como foi possível adaptar as normas pertencentes à IEEE 1451 a área de redes de sensores sem fio. Serão apresentados os componentes a serem desenvolvidos e minuciosamente discutidos a implementação dos mesmos. Este capítulo mostrará em detalhes a utilização das normas IEEE 1451 que foram escolhidas, quais opções foram feitas para essas implementações, o porquê destas escolhas e como elas foram implementadas nas plataformas e componentes escolhidos.

Os capítulo 5 e 6, juntamente, fazem as considerações finais do trabalho. O quinto capítulo faz uma conclusão geral do trabalho, resumindo tudo aquilo que foi apresentado no decorrer do projeto, expondo também as vantagens existentes nas soluções adotadas e porque as mesmas introduzem conceitos importantes para as RSSFs. Por último, o sexto capítulo propõe trabalhos futuros, derivados deste projeto, que possam ser feitos dentro do tema abordado, como pesquisas futuras ou melhorias que podem ser feitas àquilo que foi desenvolvido.

2 *Família IEEE 1451*

A IEEE 1451 é uma família de padrões que define e descreve interfaces para conexão de transdutores (sensores e atuadores), de forma independente de rede de comunicação, a microprocessadores, sistemas de instrumentação e controle, e outros sistemas dotados de transdutores (SONG; LEE, 2008). A característica fundamental dessas normas é a definição do *Transducer Electronic Data Sheets* (TEDS). O TEDS é uma estrutura de memória que acompanha o transdutor, armazenando dados de identificação, calibração, correção de dados, medição, e de informações relacionadas com a fabricação do transdutor. Uma das metas da 1451 é permitir o acesso dos dados dos transdutores através de um conjunto comum de interfaces, estejam os sensores e atuadores ligados a sistemas de redes com ou sem fio.

Atualmente, a família é formada por cinco normas ativas e uma arquivada, além de outras em formato preliminar. As ativas e arquivadas, junto com suas respectivas funcionalidades, são:

IEEE 1451.0: (IEEE. . . , 2007) é a principal norma da família 1451 e destina-se a fornecer uma base para todos os membros da 1451 que utilizam interfaces digitais, proporcionando um elevado grau de compatibilidade entre os membros da família. Três das normas 1451 foram criadas antes do início da 1451.0, fazendo com que elas não estejam de acordo com o padrão enquanto não forem revisadas. Elas são IEEE Std 1451.1TM-1999, IEEE Std 1451.2TM-1997 e IEEE Std 1451.3TM-2003.

IEEE 1451.1: (IEEE. . . , 2000) esta foi suplantada pela 1451.0 e hoje não faz parte do grupo de normas ativas, fazendo com que sua implementação não seja obrigatória. Apesar disso, ela foi a responsável por introduzir os conceitos característicos da IEEE 1451 e no futuro, após um processo de revisão, será responsável por definir aplicações entre a rede do usuário e a IEEE 1451.0.

IEEE 1451.2: (IEEE. . . , 1998) define o meio físico de comunicação, através de uma interface digital de 10 fios e um protocolo de comunicação entre transdutores e

microprocessadores. Além disso, especifica TEDS e seu formato, assim como a lógica para acessar os TEDS e os dados dos transdutores. No presente momento esta norma não especifica nada sobre conversões de sinal do meio analógico para o digital, ou a forma como o TEDS é utilizado nas aplicações, portanto será modificada no futuro, de forma a poder ser utilizada em conjunto da 1451.0.

IEEE 1451.3: (IEEE. . . , 2004) esta norma é muito semelhante ao padrão IEEE 1451.2, mas a 1451.3 prevê a interconexão de vários transdutores em uma rede multiponto, utilizando um conjunto comum de conexão por fios entre estes pontos.

IEEE 1451.4: (IEEE. . . , 2004) apresenta o modo que transdutores analógicos podem herdar a habilidade de auto-descrever suas capacidades. A norma define uma interface mista, que mantém a tradicional interface analógica para o sinal do sensor, mas acrescenta um canal digital de baixo custo para acessar TEDS embarcadas no próprio transdutor. Esta, apesar de não obrigatória e promover o uso de interfaces analógicas, pode ser utilizada em conjunto com a 1451.0, podendo servir de entrada dos TEDS para as outras normas da família. Por tanto, este vem sendo o padrão da família 1451 mais adotado pela indústria(YURISH, 2005).

IEEE 1451.5: (IEEE. . . , 2007) o objetivo desta norma é definir um padrão para acesso de transdutores através de interfaces sem fio. Além disso, define TEDS para descrever os protocolos de rádio. Foi criada com o intuito de apoiar usuários, fabricantes de transdutores e integradores de sistemas a desenvolverem comunicação sem fio para sensores e também aumentar a aceitação de tecnologias sem fios para conectividade de transdutores.

A relação entre as diferentes normas pertencentes à família IEEE 1451 estão apresentadas na figura 2.1. As partes em cinza não são cobertas pela IEEE 1451, mas podem ser utilizadas em conjunto ao padrão.

Através de uma análise rápida, é possível verificar, dentre as normas pertencentes à família 1451, que aquelas que melhor se relacionam com as características de redes de sensores sem fio são os padrões IEEE 1451.0, já que serve de base para os demais padrões, e a IEEE 1451.5, que padroniza a utilização de comunicação sem fio entre os dispositivos. Assim, as próximas seções se destinam a fazer uma síntese dos conceitos e requisitos apresentados por estas duas normas.

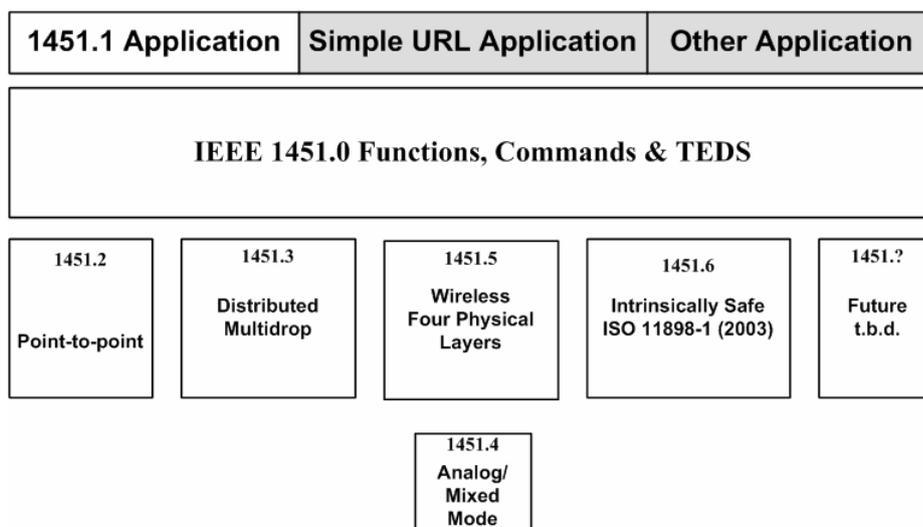


Figura 2.1: Relação das normas pertencentes à família IEEE 1451

2.1 IEEE 1451.0

A norma IEEE 1451.0 recebe o nome de “Standard for a Smart Transducer Interface for Sensors and Actuators - Common Functions, Communication Protocols, and Transducer Electronic Data Sheet (TEDS) Format”, ou em Português, “Padrão para Interface de Transdutores Inteligentes para Sensores e Atuadores - Funções Comuns, Protocolos de Comunicação, e formato do Transducer Electronic Data Sheet (TEDS)”. A norma introduz o conceito de *transducer interface module* (TIM) e de *network capable application processor* (NCAP). O TIM é um módulo que contém TEDS, a interface lógica para o transdutor, o conversor analógico-digital (ADC) e, em muitos casos, o transdutor em si. O TIM, dependendo da complexidade da aplicação pode conter de um até vários sensores e/ou atuadores. Por sua vez, um NCAP é o conjunto de hardware e software que fornece a função de ligação entre os TIMs e a rede do usuário, atuando na conversão ou outra função de processamento dos dados. TIMs e NCAPs são conectados por uma interface física especificada por um outro membro da família IEEE 1451.

Na IEEE 1451.0, um transdutor é denotado como inteligente quando atende a três requisitos básicos:

- É descrito por Transducer Electronic Data Sheets (TEDS) que são passíveis de serem lidas por computadores;
- O controle e os dados associados ao transdutor são digitais;

- Gatilhos, status e sinais de controle são fornecidos para apoiar o bom funcionamento do transdutor.

As figuras 2.2 e 2.3 mostram o modelo de referência proposto pela IEEE 1451.0 e a relação com as demais normas, cabendo a estas¹ a implementação do módulo de comunicação entre TIMs e NCAP através da interface física (PHY). A 1451.0 também define as APIs disponíveis entre os diferentes blocos do modelo, a estrutura das mensagens trocadas do NCAP ao TIM, e vice-versa, o formato dos TEDS e a forma de acesso a eles, os modos de operação e transferência que podem ser disponibilizados pelo TIM, o conjunto de comandos, descrição das funções que devem ser realizadas por um TIM, além das funções e serviços providos ao NCAP, através do módulo de comunicação, e às aplicações NCAP. O padrão permite a operação em conjunto com a IEEE 1451.4, de forma que os TEDS do transdutor possam ser obtidas de um sensor que esteja de acordo com a norma IEEE 1451.4. Diferentes fabricantes podem desenvolver NCAPs e TIMs e, caso ambas as implementações estejam de acordo com a norma, estas devem ser interoperáveis.

A parte relacionada aos componentes transdutores está fora do escopo da IEEE 1451, da mesma forma que a rede do usuário está fora do escopo da 1451.0, podendo ser ela qualquer uma que usuário deseje ou necessite. O único requisito é que o NCAP possua o hardware e software necessário para ser capaz de acessar a mesma. Os tipos de dados utilizados e seus formatos também são descritos na norma. Outras áreas do padrão são deixadas a cargo dos fabricantes, mas evitando que essas escolhas comprometam as características apontadas pela norma.

2.1.1 TransducerChannel

TransducerChannel é o nome dado pela norma ao conjunto formado pelo transdutor e os componentes de condicionamento e conversão do sinal associados a ele. Cada TransducerChannel é associado a um número de identificação de 16 bits, valor este que é definido pelo TIM. Esse número de identificação varia do valor 1 até 65535 e serve, além de distinguir os TransducerChannels existentes no TIM, de endereço de destino dos comandos enviados pelo NCAP. Dessa forma, um TIM pode conter de 0 até 65535 TransducerChannels diferentes. O TIM, por sua vez, recebe o número identificador 0, sendo qualquer comando destinado a este endereço encaminhado para o próprio TIM.

1

Até o momento, os padrões disponíveis para a interface física são IEEE Std 1451.2-1997, IEEE Std 1451.3-2003 e IEEE 1451.5TM-2007.

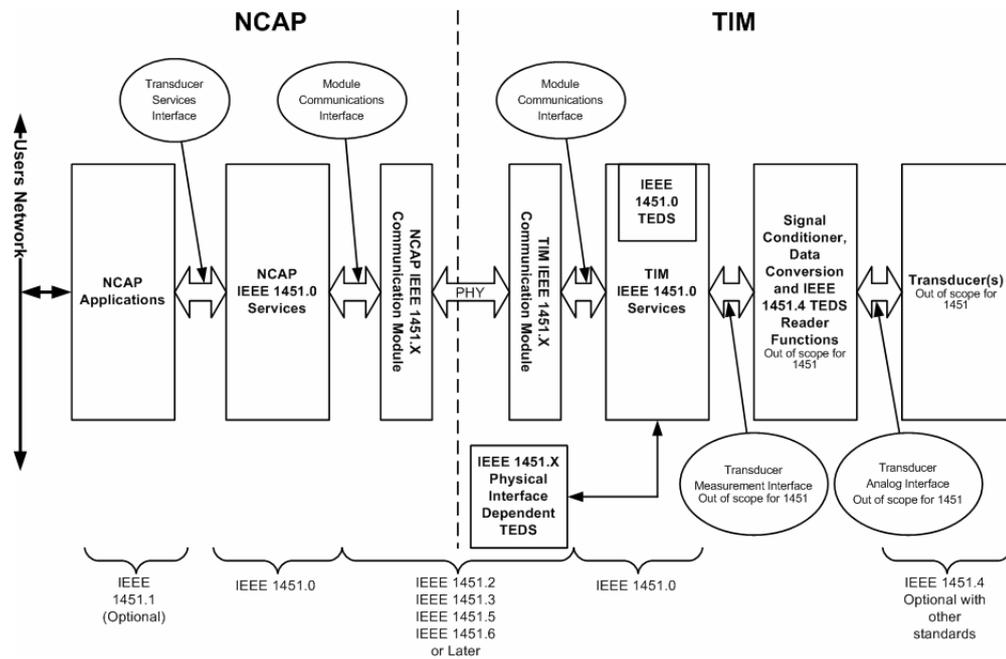


Figura 2.2: Modelo de referência proposto pela IEEE 1451.0(IEEE. . . , 2007)

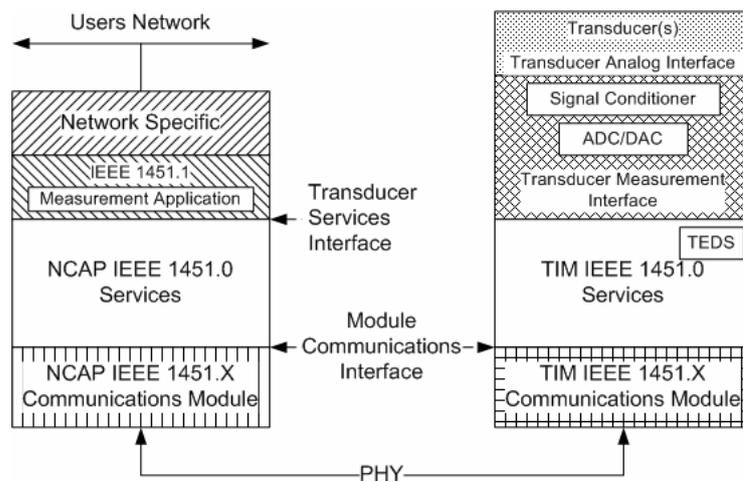


Figura 2.3: Modelo alternativo de referência proposto pela IEEE 1451.0(IEEE. . . , 2007)

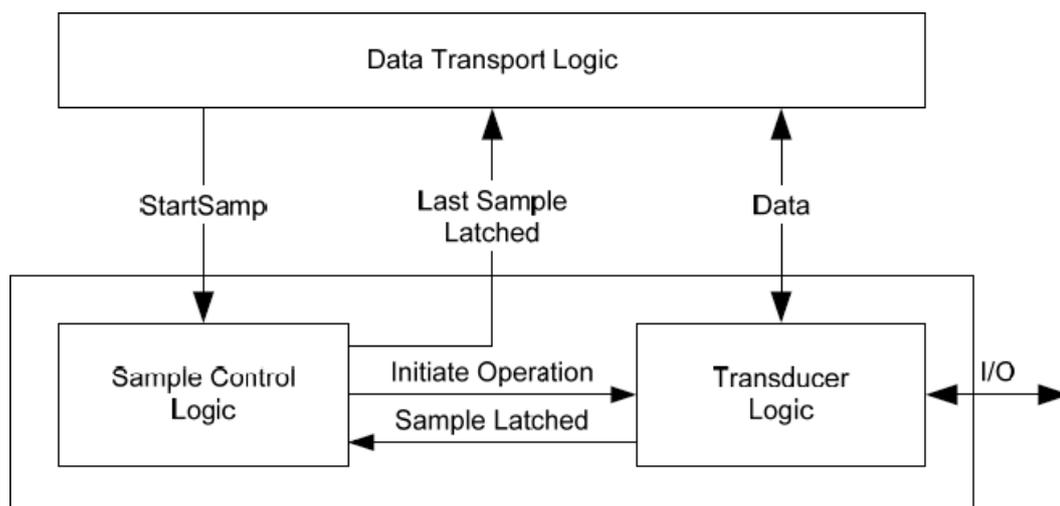


Figura 2.4: Blocos lógicos de um TransducerChannel(IEEE... , 2007)

Um diagrama de alto nível de um TransducerChannel é mostrado na figura 2.4. Após a inicialização do sistema, o TransducerChannel entra no estado “Transducer Idle”. Ao receber um comando do tipo *Operate* o TransducerChannel passa para o estado de “transducer operate”, colocando em funcionamento as lógicas do transdutor e de coleta de dados, seguindo o modo de operação definido para ele. Uma vez coletados os dados, esses são enviados através da lógica de transporte, respeitando também o modo de transferência escolhido para o canal.

Nas próximas subseções serão apresentados os componentes especificados pela norma e que constituem os dispositivos compatíveis com a IEEE 1451.0.

2.1.2 Modos de Operação

Cada TransducerChannel pode suportar diferentes modos de amostragem e de transmissão (*sampling e transmission modes*), ambos descritos na norma IEEE 1451.0. A combinação dos dois forma o modo de operação de um TransducerChannel, que descreve de forma geral como será o funcionamento do sensor ou atuador. O primeiro, o modo de amostragem, descreve como o TransducerChannel deve proceder para efetuar a obtenção de dados, no caso de um sensor, ou como os dados recebidos devem ser utilizados, caso o transdutor seja um atuador. Este modo também pode definir o intervalo, seja de tempo ou de qualquer outra grandeza definida, em que esses dados devem ser utilizados. Já o segundo, o modo de transmissão, determina como é feita a transferência dessas informações entre o NCAP e o TIM. O modo, ou os modos de operação que um TransducerChannel

pode operar estão indicados em atributos no TransducerChannel TEDS de cada um dos transdutores.

Um transdutor pode operar em um dos cinco modos de amostragem descritos na norma, sendo alguns mutuamente exclusivos e outros podendo ser habilitados ao mesmo tempo. Os modos de amostragem e suas descrições de funcionamento são:

Trigger-initiated: neste modo, um sensor iniciará a adquirir dados assim que receber um comando do tipo trigger. Já um atuador deverá usar os dados assim que receber em um comando do mesmo tipo. O processamento continua até que todos os dados do conjunto tenham sido computados, em um ritmo determinado pelo TIM.

Free-running without pre-trigger: caso o TransducerChannel seja um sensor, quando estiver em modo de operação, o mesmo deve permanecer continuamente efetuando leituras, que, entretanto, são descartadas. Uma vez que um comando do tipo trigger é recebido, as leituras seguintes passam a ser armazenadas até que um conjunto de dados esteja completo. Ao completar o conjunto, as próximas leituras voltam a ser descartadas.

Free-running with pre-trigger: um sensor deve efetuar e armazenar leituras de forma autônoma e contínua, até que um comando do tipo trigger seja recebido ou um contador chegue ao seu fim. Quando reiniciado, os primeiros dados do buffer devem ser substituídos pelas novas leituras. O comportamento deste modo, uma vez que o conjunto de dados esteja completo, é descrito pelos modos “Free-running with pre-trigger without buffers enabled” e “Free-running with pre-trigger and buffers enabled”.

Free-running with pre-trigger without buffers enabled: uma vez que o conjunto de dados de um sensor tenha sido totalmente preenchido, as próximas leituras são descartadas até que o TransducerChannel receba um comando de leitura ou de trigger. Após o recebimento de um destes comandos, o sensor faz a leitura de um novo conjunto de dados, aguardando ao final deste processo por um novo comando. Neste modo, um conjunto de dados obtido pode ser lido apenas uma vez, já que uma leitura subsequente retornará um conjunto vazio.

Free-running with pre-trigger and buffers enabled: quando um conjunto de dados é totalmente preenchido, o TransducerChannel deve trocar para o próximo buffer vazio que possuir e continuar a armazenar as leituras. Caso não existam mais buffers

vazios, as próximas leituras são descartadas até que um buffer seja disponibilizado mediante a leitura do mesmo.

Continuous sampling: nesse modo de amostragem contínua, um sensor deve começar a adquirir amostras e armazená-las em um de seus buffers quando receber um evento inicial. A operação é semelhante ao modo “free-running without pre-trigger”, exceto que o TransducerChannel não pára quando todo um conjunto de dados é adquirido, mas muda para o próximo buffer e continua disponível para a aquisição de dados. Depois que todos os buffers estão cheios, os dados contidos no buffer mais antigo começam a ser descartados, independentemente de terem sido transmitidos ao NCAP ou não.

Immediate operation: caso o TransducerChannel seja um sensor, quando ele receber um comando de leitura do conjunto de dados, o mesmo deve começar a efetuar e armazenar leituras e deverá transmiti-las, como uma resposta ao comando, assim que o conjunto estiver cheio. Já para um atuador, os dados, recebidos junto a um comando de escrita de conjunto de dados, serão utilizados imediatamente ao recebimento do comando.

Já os modos de transmissão descritos na norma e seus funcionamentos são os seguintes:

Only when commanded: um TransducerChannel só deve transmitir um conjunto de dados em resposta a um comando de leituras do conjunto de dados.

Streaming when a buffer is full: um sensor deve transmitir um conjunto de dados assim que o conjunto estiver completamente preenchido por dados que tenham sido adquiridos. Um atuador não envia fluxo de dados, portanto não pode ser operado nesta modalidade.

Streaming at a fixed interval: quando neste modo, um TransducerChannel deve transmitir um conjunto de dados completo ou parcial com regularidade e periodicidade fixa. O número de amostras em um conjunto de dados será determinado a partir da taxa de amostragem e do intervalo de transmissão.

A escolha do modo de amostragem “continuous sampling” em combinação com o modo de transferência “streaming when a buffer is full” forma um modo de operação conhecido dentro da norma como “streaming operation”. Um sensor operando nesse modo de operação deve adquirir dados e transmiti-los ao NCAP assim que o conjunto de leituras tenha

sido preenchido, sem a necessidade de receber comandos adicionais do NCAP. Dessa forma, o modo “streaming operation” permite que as leituras feitas por um sensor sejam enviadas para o NCAP de forma autônoma e automática.

2.1.3 Especificação do TEDS

TEDS são blocos de informação com o objetivo de descrever um TIM e os Transducer-Channels existentes. TEDS preferencialmente devem ser armazenados em memória não volátil no TIM. Entretanto, existem certas aplicações em que isso não é possível, fazendo com que o padrão permita que estes sejam armazenados em outros lugares do sistema. Para tanto, deve haver garantia da disponibilidade da informação que identifica o sistema de forma única e universal, o UUID. Quando armazenadas em outra localização que não seja o TIM, elas são referenciadas como TEDS virtuais.

De maneira geral, um TEDS não é mais mudado uma vez que o fabricante ou usuário definem seu conteúdo. Entretanto, é possível criar TransducerChannels que podem ter o conteúdo de seus TEDS alterados durante operação.

Todos os TEDS tem o formato padrão apresentado na tabela 2.1. O primeiro campo em qualquer TEDS é o *TEDS length*. Trata-se de um inteiro sem sinal formado por quatro octetos². O próximo bloco, de dados, apresenta o conteúdo informativo do TEDS. A estrutura e o tamanho são específicos para cada tipo de TEDS, podendo a informação armazenada ser binária ou baseada em texto. A última área em qualquer TEDS é um checksum, valor que é utilizado para verificar a integridade do TEDS. O checksum é o complemento da soma de todos os octetos anteriores, incluindo o campo inicial de comprimento do TEDS e todo o bloco de dados do TEDS, excluindo o campo de checksum. O cálculo é definido pela seguinte fórmula:

$$checksum = 0xFFFF - \sum_{i=1}^{n-2} TEDSOctet(i)$$

Os campos que compõem a estrutura do bloco de dados (data block) mudam conforme o tipo do TEDS que está sendo descrito. Todos os TEDS utilizam no bloco de dados uma estrutura de dados conhecida como TLV (Type/Length/Value), formado por três campos com diferentes funcionalidades e mostrada com detalhes na tabela 2.2. No caso de TEDS

2

Um octeto, usualmente conhecido como byte, é um grupo de 8 bits. Essa unidade é definida junto ao modelo de dados da IEEE 1451.0.

Tabela 2.1: Formato padrão dos TEDS

| Campo | Descrição | Tipo | Octetos |
|---------|-------------|----------|----------|
| — | TEDS length | Uint32 | 4 |
| 1 até N | Data block | Variável | Variável |
| — | Checksum | Uint16 | 2 |

baseadas em texto, é usada uma TLV para fornecer um diretório onde se possam acessar as TEDS que utilizam a linguagem de marcação XML para descrever o conteúdo informativo.

Dessa maneira cada entrada do bloco de dados é armazenada como uma tupla TLV. O “tipo” é um campo de um octeto que identifica o TLV, similar a tags HTML ou XML. O “comprimento” especifica o número de octetos que formam o campo “valor”, que por sua vez é onde os dados informativos são realmente guardados. Cada entrada pode ser composta por uma ou mais TLVs. A estrutura, tipo de dados e o valor que os campos devem receber são definidos na especificação de cada TEDS presente na norma.

Tabela 2.2: Definição da estrutura TLV

| Campo | Descrição |
|--------|--|
| Type | Este código identifica o campo no TEDS que é o valor contido dentro do campo “value”. Exceto para os tipos dois e três, os mesmos números terão significados diferente em cada TEDS. |
| Length | O conteúdo deste campo apresenta o número de octetos no campo “value”. O tamanho desse campo é controlado por uma entrada no TLV de identificação do TEDS. |
| Value | Este campo contém as informações do TEDS, referente ao tipo identificado no campo “type”. |

Quatro TEDS são obrigatórias para todos os TIMs, sendo as outras opcionais. As necessárias são:

Meta-TEDS: fornece parâmetros de tempo a serem utilizados pelo NCAP, definindo valores que determinam quando o TIM não está respondendo nas comunicações. Os outros itens deste TEDS descrevem o relacionamento entre os TransducerChannels que existem no TIM.

TransducerChannel TEDS: fornece informações detalhadas sobre um determinado transdutor. Informa o parâmetro físico a ser medido ou controlado, a faixa de operação

do transdutor, as características de I/O digital, os modos operacionais da unidade, bem como informação de intervalo.

User’s Transducer Name TEDS: se destina a fornecer um local para o usuário armazenar o nome pelo qual o sistema irá conhecer o TIM ou o TransducerChannel.

PHY TEDS: são dependentes da mídia de comunicação física utilizada para conectar o TIM ao NCAP. Não é definido nesta norma, embora o método de acesso seja.

A primeira TLV no bloco de dados de todos os TEDS devem seguir a estrutura indicada na tabela 2.3. Essa TLV é chamada de “TEDS identification header”, ou “cabeçalho de identificação do TEDS”, e tem o propósito de identificar o TEDS sendo descrito, assim como atributos básicos deste TEDS.

Tabela 2.3: Estrutura da TLV de identificação dos TEDS

| Campo | Função |
|--------------|--|
| Type | Identificação desta TLV, devendo seu valor ser sempre igual a três. |
| Length | Este campo tem sempre valor igual a quatro, indicando que o campo “value” tem 4 octetos. |
| Family | Parte do campo “value” que indica a qual das normas da família 1451 este TEDS pertence. Deve ser igual a zero para todas os TEDS apresentados na IEEE 1451.0. |
| Class | Parte do campo “value” que identifica qual o tipo deste TEDS. |
| Version | Este campo identifica a versão do TEDS. O valor é o número da versão identificada na norma. Um valor de zero neste campo indica que o TEDS não está em conformidade com qualquer norma liberada. |
| Tuple Length | Este campo indica o número de octetos no campo “length” de todas as TLVs, exceto por ela própria. |

2.1.4 Mensagens e Comandos

Mensagens são informações com estruturas definidas na IEEE 1451.0, passadas entre dispositivos através dos módulos de comunicação. A norma define três tipos de mensagens:

Command: são comandos enviados pelo NCAP aos TIMs e têm papéis tais como permitir leitura e escrita de TEDS, a configuração e controle dos módulos transdutores, bem como ler e escrever os dados utilizados pelo sistema. São divididos em duas categorias, que são os comandos padrões definidos pela norma e aqueles que podem ser definidos por fabricantes;

Command Reply: são mensagens enviadas dos TIMs ao NCAP em resposta aos comandos;

TIM Initiated Message: são estruturas que permitem o envio de comandos dos TIMs aos NCAP.

2.1.5 APIs

Duas APIs de software são definidas na norma IEEE 1451.0:

Transdutor Services: é uma API utilizada pelas aplicações NCAP IEEE 1451.0 para medição e controle de acesso. Esta API contém métodos para ler e escrever TransducerChannels, ler e escrever TEDS e enviar operações de configuração, controle e de comando para o TIM.

Module Communication: apresentada entre o padrão 1451.0 e outro pertencente à família 1451. Trata-se de uma interface que seria executada ambos em NCAP e TIM. Os métodos dessa API são implementados de acordo com outro membro da IEEE 1451 escolhido para comunicação.

A utilização dessas APIs é opcional, sendo estas apresentadas com o intuito de facilitar uma concepção modular do sistema, na medida em que vários fornecedores diferentes podem fornecer funcionalidade e ainda assim se integrarem perfeitamente. No entanto, mesmo não se utilizando estas APIs, permanece ainda a obrigação de que as mensagens trocadas pelas interfaces atendam o padrão.

2.2 IEEE 1451.5

A IEEE 1451.5 estabelece um padrão para métodos de comunicação de dados e formato de dados de transdutores para uso em redes sem fio. A norma introduz o conceito de Wireless Transducer Interface Module, ou WTIM, que é um TIM baseado na norma IEEE 1451.0, mas com a capacidade de comunicação em redes sem fio. Além disso, a IEEE 1451.5 também define o PHY TEDS, baseada na IEEE 1451.0, descrevendo a interface física de comunicação. O WTIM se conecta a um módulo de serviços NCAP através de uma rede sem fio e um módulo de rádio-comunicação. O padrão apresenta algumas tecnologias de rádio pré-aprovadas ³, conhecidas como Dot5AR (IEEE 1451.5 approved

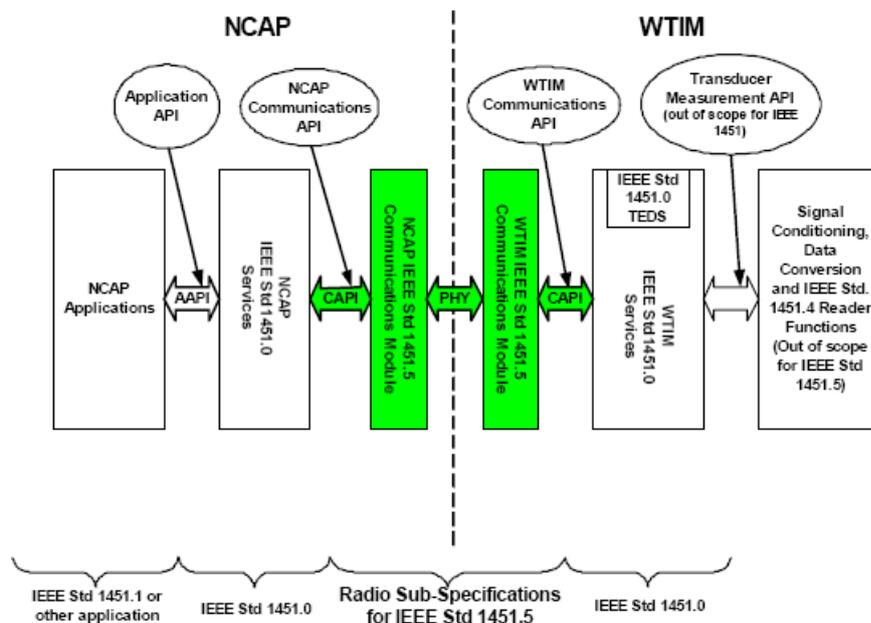


Figura 2.5: Contexto funcional da IEEE 1451.5 (IEEE. . . , 2007)

radios), mas permite a utilização de tecnologias de comunicação sem fio escolhidas pelo desenvolvedor. Não cabe à norma definir o design de transdutores, condicionamento de sinal, o uso dos TEDS ou design do sistema físico de comunicação sem fio.

O módulo de comunicação é definido pelo padrão IEEE 1451.0, cabendo à 1451.5 fazer a descrição dos aspectos físicos da interface de comunicação, conforme mostrado na figura 2.5. O NCAP, definido pela IEEE 1451.0, se comunica com o módulo de comunicação NCAP IEEE 1451.5 através da *Communications API*. O módulo de comunicação do NCAP por sua vez se comunica com o módulo de comunicação do WTIM, através da interface física de rádio. No WTIM, a interface de serviços do WTIM também interage com o módulo de comunicação WTIM IEEE 1451.5 através da *Communications API*.

A figura 2.6 apresenta uma configuração típica da IEEE 1451.5, onde as funções dos módulos são definidas da seguinte forma:

NCAP: este dispositivo se conecta a um ou mais WTIMs via protocolos Dot5AR e pode ser conectado a uma rede de usuário.

WTIM: este dispositivo está ligado a transdutores e é ligado através de protocolos Dot5AR a um NCAP. Um WTIM é um TIM, tal como definido na IEEE 1451.0, incluindo comunicação sem fio com o NCAP.

as tecnologias de comunicação sem fio pré-aprovadas pela IEEE 1451.5 são: IEEE 802.11, IEEE 802.15.4, Bluetooth e ZigBee.

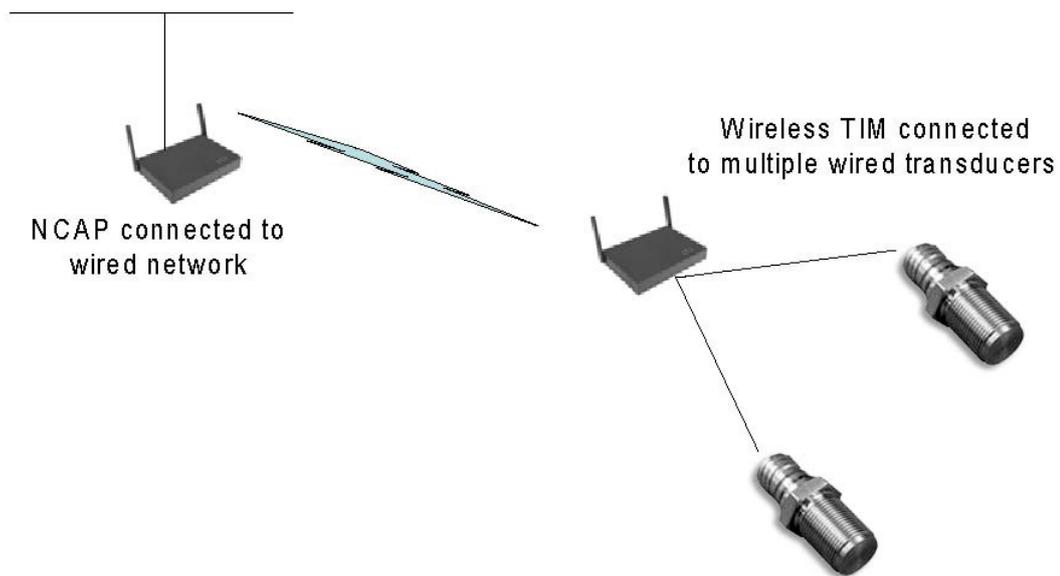


Figura 2.6: Configuração típica da IEEE 1451.5 (IEEE. . . , 2007)

As seguintes regras também são definidas pela norma:

- Um NCAP pode encaminhar comandos e dados para uma rede externa de e para os transdutores conectados no WTIM;
- Um NCAP pode registrar vários WTIMs;
- Um WTIM só pode se registrar em um único NCAP;
- Um WTIM pode ser interface para múltiplos transdutores;
- É permitidas a troca de informação entre WTIMs.

2.2.1 PHY TEDS

A função do PHY TEDS é de tornar disponíveis na interface todas as informações necessárias para ter acesso a qualquer canal, juntamente com a informação comum a todos os canais. Os octetos do PHY TEDS são constantes e somente para leitura, seguindo o formato geral definido na IEEE 1451.0 e utilizando a arquitetura de TLVs.

Para os módulos de comunicação que utilizam interfaces físicas de comunicação sem fio, as PHY TEDS são definidas na norma IEEE 1451.5.

2.3 Implementações existentes

Apesar de existirem, boa parte das implementações da família IEEE 1451 são anteriores a disponibilização da norma IEEE 1451.0 (WOBSCHALL, 2006; LEE, 2000; OOSTDYK et al., 2006; RAMOS; RAMOS; PACES, 2007).

A “Java Distributed Data Acquisition and Control”, é um exemplo de projeto de implementação da IEEE 1451 anterior à IEEE 1451.0. A JDDAC disponibiliza uma plataforma para a construção de redes de sensores e sistemas de aquisição de dados, fornecendo os meios para gerenciar o acesso e controle de transdutores. Mais especificamente, ela é baseada nas normas IEEE 1451.1 e IEEE 1451.2, e construída sobre a plataforma Java, permitindo que aplicações Java interajam com o mundo físico.

Porém, mesmo a IEEE 1451.1 sendo a introdutora dos conceitos da família IEEE 1451, ela não define os requisitos e obrigações das implementações. Esse papel, como visto no capítulo anterior, é assumido pela norma IEEE 1451.0, que serve de base para a implementação de todas as outras normas⁴. Isso faz com que não só as normas anteriores a 1451.0 sejam incompatíveis com a mesma, mas também com que as implementações feitas anteriormente a sua disponibilização também sejam.

Tratando-se de implementações contemporâneas à IEEE 1451.0, estas ainda se apresentam em pequena quantidade (COURSES; SURVEYS, 2008). E destas, um número menor ainda são de trabalhos aplicados a redes de sensores sem fio, e mesmo quando aplicadas uma pequena parte delas é baseada em nós sensores comerciais (LEE; SONG, 2007; SONG; LEE, 2006). Além disso, estas implementações têm se mostrado mais como provas de conceito, apresentando apenas as características básicas da norma e deixando de demonstrar algumas capacidades que tornam esta uma família realmente única, como por exemplo, a possibilidade de integração e interoperabilidade entre diferentes redes (WOBSCHALL; MUPPARAJU, 2008; NEMETH-JOHANNES; SWEETSER; SWEETSER, 2007; KULARATNA; SUDANTHA, 2008).

4

A IEEE 1451.4, é uma exceção, já que a mesma pode ser implementada independentemente das demais normas.

3 *Nós Sensores*

Nós sensores, também conhecidos pela denominação inglesa *motes*, são os componentes básicos das redes de sensores sem fio. O conceito de motes nasceu junto ao projeto *Smart Dust* (WARNEKE et al., 2001), que descreve uma rede hipotética formada por dispositivos do tamanho de um grão de areia, capazes de atuar e coletar dados sobre o meio em que estão presentes. Esses nós são dispostos sobre o ambiente de estudo, de maneira semelhante à mostrada na figura 1.1, de forma a serem capazes de monitorar os fenômenos que acontecem no local onde estão.

Motes podem ser imaginados como computadores extremamente básicos em termos de interface e componentes. Segundo Akyildiz et al (2002b), um nó sensor é formado basicamente por quatro partes. Como mostrado na figura 3.1, esses blocos básicos são:

- Unidade de sensoriamento;
- Unidade de processamento;
- Unidade transceptora;
- Unidade de energia.

Além dessas, podem ser encontradas unidades adicionais dependentes de aplicação. A unidade de sensoriamento é geralmente formada por outras duas sub-unidades: sensores e conversores analógicos digitais, cabendo a estes medir a grandeza de um fenômeno físico e converter a leitura em um valor digital, respectivamente. Dessa forma, é possível utilizar na unidade processadora as informações convertidas. À unidade processadora, programável conforme a aplicação da RSSF e que costuma estar associada a uma pequena unidade de armazenamento, cabe a tarefa de comunicar e colaborar, através da unidade transceptora, com os outros nós sensores, de forma a cumprir as tarefas incumbidas. A unidade de energia é responsável por fornecer corrente elétrica para todo o sistema.

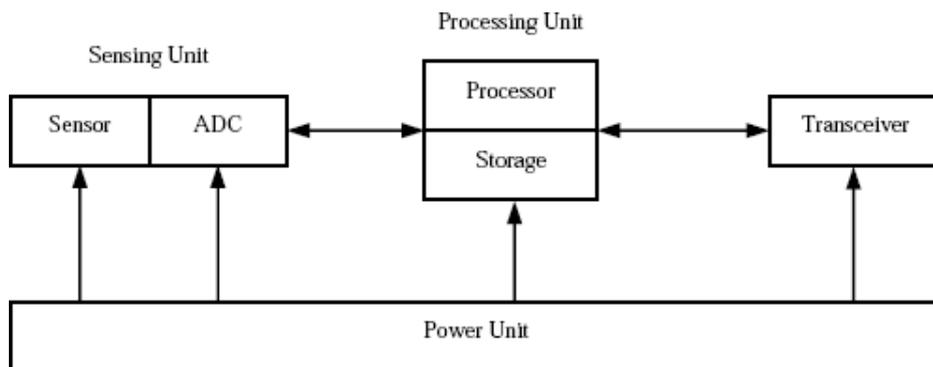


Figura 3.1: Blocos básicos de um nó sensor(KHEMAPECH; DUNCAN; MILLER, 2005)

Para a implementação do projeto, foram escolhidos dois tipos de nós sensores, devido à disponibilidade e peculiaridades dos mesmos. Esses nós são descritos nas seções seguintes.

3.1 Gateway

Uma função importante da camada de rede RSSF é a de proporcionar contato externo com outras redes, que possam papéis tais como sistemas de comando e controle através da Internet ou armazenamento de dados. O Gateway, algumas vezes também chamado de *sink* na notação de redes de sensores sem fio, é um caso especial de um nó sensor(AKYILDIZ et al., 2002b; ABOELAZE; ALOUL, 2005; SCHMID; DUBOIS-FERRIERE; VETTERLI, 2005). Ele age como intermediário entre a rede sem fio e redes externas, permitindo ligar os nós sensores pertencentes a RSSF à rede do usuário.

3.2 Sun SPOT

Sun SPOT, acrônimo de “Sun Small Programmable Object Technology”, é um nó de redes de sensores sem fio desenvolvido pela Sun Microsystems(ARSENEAU et al.,). Os detalhes acerca do hardware do dispositivo, que pode ser visto na figura 3.2, estão apresentados na tabela 3.1. Diferentemente da maior parte dos motes existentes, o Sun SPOT possui uma máquina virtual Java, chamada *Squawk*, que atua na execução dos programas desenvolvidos para este, utilizando a linguagem Java(SIMON et al., 2006; SIMON; CIFUENTES, 2005; GOSLING et al., 2000). A Squawk é uma máquina virtual (VM) Java escrita principalmente em Java e projetada para dispositivos com recursos limitados, sendo compatível com a “Connected Limited Device Configuration” (CLDC), configuração base do Java Micro Edition (Java ME), que define APIs para dispositivos que

dispõem de poucos recursos computacionais(MICROSYSTEMS, 2000).

O SPOT proporciona tanto o hardware quanto o software necessário para facilmente construir diversas aplicações de sensoriamento. A plataforma de hardware oferece recursos e capacidades suficientes para permitir aos desenvolvedores construir e experimentar aplicativos sem a preocupação de limitações e/ou expansão de capacidades. O dispositivo é equipado com uma bateria que pode ser recarregada através da porta USB e através de conexões físicas é possível acoplar até duas placas externas ao SPOT, encaixadas uma acima da outra, ampliando a capacidade do dispositivo. A plataforma de software introduz um conjunto completo do uso da tecnologia Java em sistemas dotados de transdutores e comunicação sem fio. Também inclui serviços adaptados especificamente para estes tipos de sistemas, como instalação e depuração de aplicações através do rádio, integração com desktop e servidores, gestão automática de energia, entre outros recursos. Combina também, ao processo de desenvolvimento, a possibilidade de utilizar as ferramentas disponíveis para tecnologia Java, como por exemplo, os ambientes de desenvolvimento ou IDEs.

A necessidade de conservar energia é uma questão importante para os dispositivos sensores sem fio. O Sun SPOT tem um poderoso processador, rádio, grande quantidade de memória e, por isso, tem o potencial de esgotar sua bateria dentro de poucas horas. Uma solução comum para esse problema é permitir que a aplicação faça um pedido para desligar os componentes que não esteja utilizando, entrando em um modo de baixo consumo. Isto coloca a gestão de consumo de energia sobre o desenvolvedor, que quer apenas se concentrar no domínio da aplicação, além de serem grandes as chances de o desenvolvedor perder oportunidades de poupar energia. Em vez disso, o SPOT tenta automatizar este gerenciamento tanto quanto possível. Quando nenhuma thread está programada para ser executada por mais de um determinado intervalo mínimo de tempo, o sistema automaticamente entra em modo de poupança de energia.

3.2.1 eDemo Board

Para facilitar o desenvolvimento junto ao Sun SPOT é distribuída uma placa externa demonstrativa chamada “eDemo Board”, juntamente com as classes Java que permitem controlar os componentes da placa. A eDemo possui classes Java compreensíveis, firmware pré-instalado e várias aplicações de demonstração para os componentes que a acompanham:



Figura 3.2: Nó sensor Sun SPOT

Tabela 3.1: Hardware do Sun SPOT

| | |
|--------------------------|--------------------------------------|
| Nome | Sun SPOT |
| Processador | 180 MHz 32-bit CPU |
| Memória Flash | 4 MB |
| Memória RAM | 512 KB |
| Rádio | 2.4 GHz IEEE 802.15.4 |
| Bateria | 3.7V 720mAh rechargeable lithium-ion |
| Consumo em sleep | 32 uA |
| Funcionamento, sem rádio | 80 mA |
| Funcionamento, com rádio | 98 mA |

- Acelerômetro de 3 eixos;
- Sensor de luz;
- Sensor de temperatura;
- Conversor analógico-digital (ADC);
- 8 LEDs de 3 cores;
- 2 interruptores (switches);
- 6 entradas analógicas;
- 5 pinos de I/O digital;
- 4 saídas de alta corrente.

3.3 Crossbow MICA

Os motes MICA, mostrados na figura 3.3, são uma das plataformas disponíveis comercialmente para pesquisadores e desenvolvedores de redes de sensores sem fio. Criado por um grupo de pesquisa da Universidade de Berkeley como uma solução de fonte aberta, hoje é montado e comercializado pela companhia Crossbow Technology (HILL; CULLER, 2002).

O MICA tem todas as características típicas de um mote, usando um processador Atmel ATmega 128L rodando em uma frequência de 4 megahertz. O 128L é um microcontrolador de 8 bits que é acompanhado de 128 kilobytes de memória flash para armazenar o programa do mote. A grande diferença é que o ATmega consome apenas 8 miliampéres quando em funcionamento e apenas 15 microampéres quando no modo de dormência. Isso permite um baixo consumo de energia, funcionando por mais de um ano com duas pilhas AA. Os detalhes de hardware do MICA estão dispostos na tabela 3.2. Sensores externos podem ser acoplados através de um cartão externo e um conector de 51 pinos, a fim de expandir as capacidades do mote com uma ampla variedade de periféricos.

Por padrão, o software para o MICA mote é construído sobre um sistema operacional chamado TinyOS (LEVIS et al., 2005). O TinyOS provê abstrações e métodos para o acesso aos sensores, rádio e ao hardware em geral, tornando muito mais fácil de escrever software para o nó sensor.



Figura 3.3: Nó sensor Crossbow MICA

Tabela 3.2: Hardware do Crossbow MICA

| | |
|--------------------------|---------------------------|
| Nome | Crossbow MICA |
| Processador | 4 MHz 8-bit CPU |
| Memória Flash | 128 KB |
| Memória RAM | 4 KB |
| Rádio | 868/916 MHz Multi-Channel |
| Bateria | 2x AA |
| Consumo em sleep | < 1 uA |
| Funcionamento, sem rádio | 8 mA |
| Funcionamento, com rádio | 27 mA |

3.3.1 MTS300

A MTS300 é uma das placas de sensores disponibilizadas pelas Crossbow Technology (REV, 2005). Os recursos dispostos por ela são:

- Sensor de luz;
- Sensor de temperatura;
- Sensor acústico;
- Emisor de som;
- Circuito detector de tons.

3.3.2 EPOS

O projeto EPOS (Embedded Parallel Operating System) visa automatizar o desenvolvimento de sistemas dedicados de computação, de modo que os desenvolvedores possam concentrar no que realmente importa: suas aplicações (MARCONDES et al., 2007). EPOS se baseia no conceito “Application-Oriented System Design Method” (AOSD), proposto por Fröhlich (2001), para conceber e implementar ambos os componentes de software e hardware que podem ser automaticamente adaptados de forma a satisfazer às exigências de uma aplicação específica. Para tanto, o EPOS disponibiliza uma série de ferramentas que permitem configurar e acrescentar componentes, montando um framework específico para a aplicação que por sua vez habilita a geração de uma instância de um sistema orientado à aplicação. Essa instância consiste de uma plataforma de hardware implementada em termos de lógica programável e do sistema de execução correspondente implementado.

Uma aplicação concebida e implementada de acordo com a estratégia proposta pelo EPOS pode ser submetida a uma ferramenta que irá efetuar análises sintáticas e de fluxo de dados para extrair um plano para o sistema operacional a ser gerado e, quando instanciado, a descrição de alto nível para a plataforma de hardware a ser sintetizada. O plano é então refinado através de análise de dependências das informações sobre o cenário de execução, fornecidas pelo usuário a partir de ferramentas visuais. O resultado deste processo é um conjunto que irá apoiar tanto a compilação do sistema operacional orientado a aplicações, quanto a síntese de componentes de hardware em um dispositivo lógico programável. Esses passos podem ser resumidos através da figura 3.4, que apresenta a visão geral do projeto.

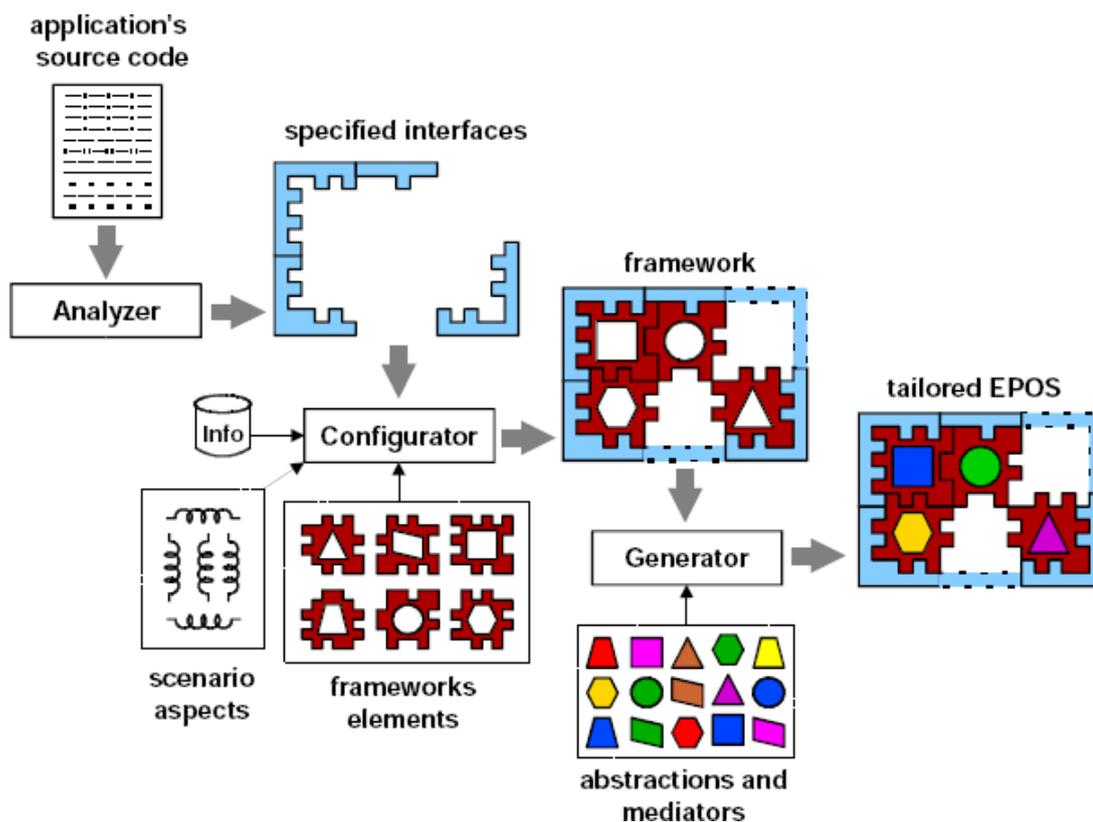


Figura 3.4: Visão geral do projeto EPOS(FROHLICH, 2001)

O trabalho de Wanner (2003) originou um ambiente de suporte à execução de aplicações em redes de sensores sem fios a partir do sistema operacional EPOS, possibilitando, através de um porte do sistema, que o mesmo pudesse ser utilizado sobre a plataforma de sensoriamento MICA. Além do porte, o trabalho de Wanner incluiu o projeto e implementação do protocolo de controle de acesso ao meio C-MAC (Configurable MAC) e uma abstração do sistema de aquisição de leituras dos sensores.

No TinyOS, o sistema operacional mais utilizado para redes de sensores sem fios na atualidade, não há abstrações independentes de plataforma específicas para dispositivos sensores além da interface do canal do ADC. Isto faz com que as aplicações tenham que complementar as funcionalidades dos drivers dos sensores, comprometendo a portabilidade das mesmas. O sistema de aquisição de dados de sensores desenvolvido no EPOS é capaz de abstrair famílias de dispositivos sensores de maneira uniforme, sem ocasionar sobre custo excessivo, apresentando vantagens significativas com relação a outras soluções encontradas em outros sistemas operacionais para redes de sensores(WANNER et al., 2006). Além disso, testes do C-MAC no EPOS apresentaram resultados de vazão e taxa de recepção na rede equivalentes ou melhores do que o protocolo B-MAC no TinyOS,

quando os dois protocolos foram configurados de maneira idêntica.

4 *Desenvolvimento*

Segundo Lee (2000), a família de padrões IEEE 1451 foi criada com o intuito de ser aplicável a qualquer sistema dotado de transdutores, portanto, podendo ser aplicada a redes de sensores sem fio. Comparando o modelo de objetos proposto pela norma IEEE 1451.0 com o contexto de RSSF, poderíamos igualar o NCAP a um gateway, e o WTIM aos nós sensores.

As RSSF costumam ter grandes restrições quando se trata da quantidade de energia disponível, já que dependem de fontes de energia limitada e ainda assim devendo permanecer operantes por um longo período de tempo. Sendo este um dos principais gargalos da tecnologia de RSSF, certas aplicações optam por manter os nós sensores em estado de dormência, religando-os em intervalos de tempo e fazendo as devidas leituras, processamento e comunicação com os outros nós (MAINWARING et al., 2002; SZEWCZYK et al., 2004; WERNER-ALLEN et al.,). Essa forma de operação mostra uma tendência onde os motes são responsáveis por tomarem a iniciativa de comunicação com o gateway da rede a qual eles pertencem, enquanto na IEEE 1451.0 essa ação costuma ser tomada pelo NCAP (LEE; SONG, 2007). Apesar disso, o padrão fornece as bases necessárias para implementar um sistema em que o TIM toma as iniciativas de comunicação. A norma 1451.0 descreve as estruturas de “TIM initiated message“, mas só demonstra sua utilização no envio do status dos TIMs ao NCAP. Nesta pesquisa, procura-se desenvolver uma solução para utilizar a norma IEEE 1451 em redes de sensores sem fio onde os nós sensores tomem a iniciativa de comunicação através da utilização de mensagens iniciadas pelo TIM, mantendo ainda a capacidade da norma de adicionar diferentes tipos de nós sensores devido a existência da padronização.

Seguindo as informações contidas nas normas IEEE 1451.0 e IEEE 1451.5, selecionadas para serem usadas neste trabalho em conjunto de redes de sensores sem fio, definiu-se para implementação a arquitetura apresentada na figura 4.1. Nela, são apresentadas as duas RSSF, formadas pelos nós Sun SPOT e Crossbow MICA rodando o sistema EPOS, atuando no papel de Wireless TIMs, se comunicando através de seus respectivos gateways

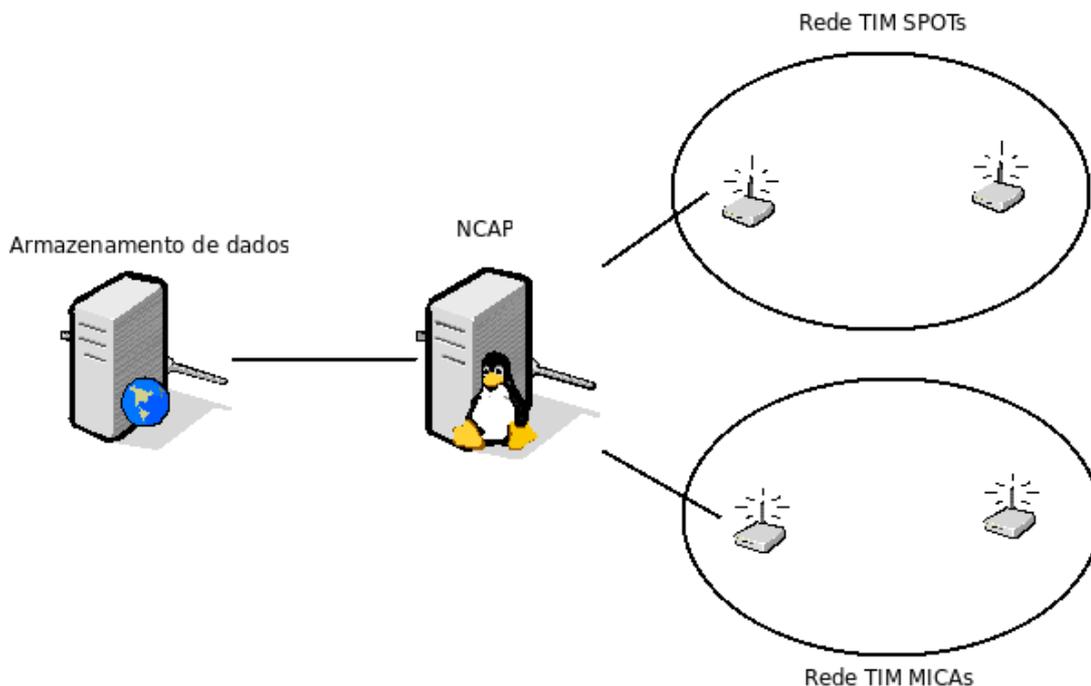


Figura 4.1: Arquitetura definida para a implementação das normas IEEE 1451

com o NCAP. As vantagens apresentadas pelo EPOS tornam-no uma ótima opção de sistema operacional para ser utilizado junto à plataforma MICA, por isto foi feita sua escolha neste trabalho. O NCAP é um software escrito em linguagem Java (GOSLING et al., 2000), sendo executado em uma máquina virtual Java em um computador com sistema operacional Linux. Apesar de Java ser independente de plataforma, o sistema Linux foi escolhido devido à necessidade de acesso a código nativo do sistema. Pelo meio de duas aplicações NCAP, os TEDS dos WTIMs que se registram ao NCAP são armazenados assim como todos os dados lidos pelos sensores e enviados ao NCAP pelos TransducerChannels, através de mensagens iniciadas pelo TIM, são encaminhados a uma aplicação web externa, que tem o papel de armazenar essas informações e gerar gráficos a partir delas.

Por motivos de simplicidade, programou-se apenas um TransducerChannel em cada um dos WTIMs, formado por um sensor de temperatura. Por conseguinte, ambos os motes estarão reportando leituras em graus Kelvin, unidade de base do Sistema Internacional de Unidades (SI) para a grandeza temperatura termodinâmica (WEIGHTS et al., 2008).

4.1 Operação dos WTIMs

Como discutido nos capítulos anteriores, um dos maiores problemas em redes de sensores sem fio é a quantidade de energia disponível. Devido a este fato, certas aplicações optam por manter os nós sensores em estado de dormência durante a maior de seu funcionamento, religando-os em intervalos de tempo e fazendo as devidas leituras, processamento e comunicação com os outros nós, enviando os dados coletados pelos mesmos. Dos modos de operação disponíveis na IEEE 1451.0, aquele que melhor se encaixa com o cenário no qual o mote ficara a maior parte da sua vida em estado de dormência é o modo de operação “Streaming operation“, que determina a utilização do modo de transferência “streaming when a buffer is full“ juntamente com o modo de amostragem “continuous sampling“. Porém, a norma não descreve qual estrutura deve ser utilizada para enviar os dados coletados para o NCAP, tornando-se importante determinar um modo para que essa transferência seja feita, possibilitando a um TIM tomar a iniciativa de enviar os dados coletados por seus TransducerChannels ao NCAP em que ele está registrado.

Neste trabalho, adotou-se a estrutura de “TIM Initiated Message“ descrita pela norma, preenchendo os campos existentes com os valores conforme orientados pelo comando “Read TransducerChannel data-set segment“. Dessa forma em conjunto com o modo de operação “streaming operation“, a aplicação NCAP, uma vez que tenha posto o TransducerChannel em modo de operação, deve esperar pelas amostras obtidas pelo sensor através de um comando de leitura do conjunto dos dados enviado pelo WTIM ao NCAP. A estrutura dessa mensagem, com os campos preenchidos conforme o comando de leitura de conjunto de dados, é demonstrada na tabela 4.1. Os dois primeiros octetos são formados pelo endereço do TransducerChannel que está enviando a mensagem, sendo respectivamente o byte mais e menos significativo (MSB e LSB) o primeiro e o segundo. Como nessa implementação há apenas um TransducerChannel, o valor desse será igual a 1, indicando o primeiro sensor, o de temperatura. Em seguida, há dois octetos especificando o tipo de comando que essa mensagem é. O byte mais significativo, com valor três, indica a classe de comandos a serem usados quando um transdutor está no modo de operação (transducer operating state commands) e o menos significativo a função do comando. O valor um indica a função “read TransducerChannel data-set segment“, usada para ler um conjunto de dados do TransducerChannel. A terceira parte, também formada por dois octetos, indica o número de octetos dependentes da mensagem, ou seja, quantos octetos são esperados no campo de dados. O TEDS dos transdutores de temperatura descrevem que os conjuntos de dados são formados por 10 leituras, sendo cada uma delas um número

de ponto flutuante de 32 bits, seguindo o padrão IEEE 754 (IEEE . . . , 1985). Dessa forma, o campo de comprimento recebe o valor 40. Por último estão os 40 octetos contendo em seqüência de leituras, as informações obtidas pelo sensor de temperatura.

Tabela 4.1: Estrutura da mensagem iniciada pelo TIM para envio das leituras

| | | |
|-----------------------------|-----|------|
| Número do TransducerChannel | msb | 0x00 |
| | lsb | 0x01 |
| Comando | msb | 0x03 |
| | lsb | 0x01 |
| Comprimento | msb | 0x00 |
| | lsb | 0x28 |
| Dados | 1 | 0x?? |
| | ... | 0x?? |
| | 40 | 0x?? |

Resumidamente, uma vez que um WTIM tenha se registrado ao NCAP, os seguintes passos são seguidos:

1. O NCAP envia um comando do tipo “operate” para colocar o TransducerChannel escolhido do TIM em operação;
2. O TransducerChannel entra em modo de operação, desconectando se do NCAP caso possível.
3. O TransducerChannel começa a fazer leituras no sensor em intervalos regulares até que o conjunto de dados esteja completo, colocando o WTIM em modo de dormência entre esses intervalos;
4. O WTIM se conecta novamente e o conjunto de dados é enviado ao NCAP, utilizando a estrutura de uma mensagem iniciada pelo TIM e os campos seguindo o formato de um comando de leitura do conjunto de dados;
5. O WTIM se desconecta novamente, retomando as operações a partir do segundo passo.

4.2 Construção dos TEDS

As próximas subseções detalham a construção dos TEDS utilizados nos WTIMs implementados. O processo de construção de TEDS consiste em seguir os formatos descritos

pela norma, definindo e atribuindo os valores de cada um dos campos conforme o TIM que está sendo descrito. O primeiro TLV de todos os TEDS é sempre o campo TEDSID, que descreve informações básicas para compreensão do TEDS sendo descrito. Os TEDS montados descrevem um TIM capaz de operar no modo “streaming operation” e utilizando para comunicação um protocolo de radiofrequência. Além disso, o sistema é alimentado por bateria e formado por um sensor de temperatura que não possui nenhuma forma de calibração. Esse sensor fornece leituras em graus Kelvin, unidade física suportada pela norma.

Os octetos atribuídos para os TEDS são aqui apresentados no sistema de numeração hexadecimal. Para simplificar o processo de desenvolvimento, somente os TEDS obrigatórios e os campos necessários de cada um foram montados. Mesmo assim, a estrutura e os detalhes utilizados seguem fielmente a forma descrita nas normas.

4.2.1 Meta-TEDS

Os campos obrigatórios dos Meta-TEDS estão apresentados na tabela 4.2 e foram preenchidos da seguinte forma:

TEDSID: os quatro octetos que formam o campo de valor da TLV de identificação tomam valores 0, 1, 1 e 1, que, respectivamente, indicam que esse TEDS pertence à norma IEEE 1451.0, é um TEDS da classe Meta-TEDS, o formato do TEDS é descrito pela versão mais recente do padrão e que os campos de tamanhos das demais TLVs terão tamanho igual a 1 octeto.

UUID: como esse WTIM e TEDS não tem a intenção de serem colocados em funcionamento, preencheram-se todos os octetos do campo de valor com o número 0.

OHoldOff: após 5 segundos, um comando sem resposta enviado para este WTIM deve ser considerado como uma operação falha. Esse valor é representado em notação de ponto flutuante seguindo o padrão IEEE 754.

TestTime: devido a inexistência da capacidade de auto teste por parte do WTIM nesta implementação, essa TLV recebe valor 0.

MaxChan: já que foi escolhido implementar apenas um TransducerChannel em cada WTIM, esse campo recebe o valor de 1, indicando que apenas um transdutor está presente.

Tabela 4.2: Campos obrigatórios do Meta-TEDS

| Tipo | Nome | Descrição | Dados | Octetos |
|------|----------|--|---------|---------|
| — | Length | Comprimentos do TEDS | UInt32 | 4 |
| 3 | TEDSID | Cabeçalho de identificação do TEDS | UInt8 | 4 |
| 4 | UUID | Identificador globalmente único | UUID | 10 |
| 10 | OHoldOff | Tempo limite operacional | Float32 | 4 |
| 12 | TestTime | Tempo de teste | Float32 | 4 |
| 13 | MaxChan | Número de TransducerChannels implementados | UInt16 | 2 |
| — | Checksum | Checksum | UInt16 | 2 |

Finalmente, os octetos definidos neste trabalho para descrever o Meta-TEDS estão na tabela 4.3. Os 4 primeiros octetos indicam que esse TEDS tem um tamanho de 36 octetos, sem contar com o próprios octetos que indicam o tamanho do TEDS, e os dois últimos apresentam o checksum calculado para estes octetos.

Tabela 4.3: Octetos do Meta-TEDS

| Tipo | Comprimento | Valor |
|------|-------------|--|
| — | — | 0x00 0x00 0x00 0x24 |
| 0x03 | 0x04 | 0x00 0x01 0x01 0x01 |
| 0x04 | 0x0a | 0x00 |
| 0x0a | 0x04 | 0x40 0xa0 0x00 0x00 |
| 0x0c | 0x04 | 0x00 0x00 0x00 0x00 |
| 0x0d | 0x02 | 0x00 0x01 |
| — | — | 0x00 0x00 |

4.2.2 TransducerChannel TEDS

O TransducerChannel TEDS montado neste trabalho irá se referir a um sensor de temperatura linear. Apesar dos componentes sensores serem diferentes em cada um dos nós, a mesma descrição foi utilizada para ambos, já que ambos farão os mesmos tipos de leituras. Alguns campos do TEDS são obrigatórios apenas para atuadores ou sensores geradores de eventos, não havendo necessidade de sua utilização neste caso. Neste TEDS, existem campos formados por sub-campos, utilizando internamente a mesma estrutura de TLVs descrita anteriormente. A tabela 4.4 resume os campos obrigatórios e não obrigatórios que foram utilizados para descrever os sensores de temperatura. A descrição do

preenchimento dos valores dos campos recebe os seguintes valores:

TEDSID: os quatro diferentes atributos definidos nessa TLV receberam, em ordem, os valores 0, indicando que este TEDS é descrito pela norma IEEE1451.0, 3, denotando esse como um TEDS da classe TransducerChannel, 1, referindo-se à norma mais atual e 1, informando que os campos de tamanho das TLVs a seguir serão sempre formados por apenas um octeto.

CalKey: este WTIM não irá fornecer nenhum tipo de calibração. Segundo o padrão, o valor para representar esse comportamento é zero.

ChanType: como o transdutor deste TransducerChannel é um sensor de temperatura, o campo de valor deve receber o número 0, seguindo a definição provida pela norma.

PhyUnits: este campo é formado por outros dez campos, que em conjunto descrevem a unidade de medida, segundo o sistema internacional de unidades (SI) (WEIGHTS et al., 2008), das leituras efetuadas por este sensor. O primeiro destes sub-campos, identificado pelo tipo de valor 50, é uma enumeração que descreve como devem ser interpretadas as unidades descritas nos campos que são apresentados em seguida. Fixou-se o valor zero para este campo, informando que a unidade é descrita pelo produto das potências das unidades bases do sistema internacional. Os outros nove campos são formados pela unidade básicas do SI. Quando o expoente dessas unidades for igual a zero, o campo deve receber o valor 128. Como os sensores que estão descritos neste WTIM são de temperatura, a medição será representada em graus Kelvin, uma das unidades pertencentes ao sistema internacional de medições. Portanto, o sub-campo 57, que descreve o expoente Kelvin, será fixado no valor 130, indicando que as leituras de temperatura do sensor são lineares.

LowLimit: definiu-se como o menor valor possível de leitura deste sensor em $-40\text{ }^{\circ}\text{C}$, que devido à necessidade de ser descrito segundo o sistema internacional de unidades é convertido para 233.15 K . Como esse é um valor aproximado, arredondou-se o valor para 233 K .

HiLimit: para valor máximo de uma leitura provida por este sensor estabeleceu-se a temperatura de $80\text{ }^{\circ}\text{C}$. Usando as unidades do sistema internacional isso seria equivalente a 353.15 K , arredondando-se para 353 K , já que este é uma valor aproximado.

Oerror: estipulou-se uma variação de erro de até $0.5\text{ }^{\circ}\text{C}$ para estes sensores. Como a unidade de medida Kelvins é proporcional à unidade Celsius, não é necessária uma

conversão deste valor.

SelfTest: o TIM não tem a capacidade de efetuar testes em seu sistema, por isso é fixado o valor 0 para este campo.

Sample: este campo é formado por três sub-campos obrigatórios. Para o primeiro sub-campo, que informa o modelo dos dados que serão coletados, foi fixado o valor um, indicando que as leituras efetuadas serão do tipo ponto flutuante de 32 bits. Por conseguinte, o segundo sub-campo recebeu o valor 4, já que o tamanho de cada amostra será de quatro octetos. O último campo recebeu valor 0 para precisão, por ser desnecessário.

DataSet: o campo, também formado por sub-campos, detalha o conjunto de dados coletados pelo sensor. O primeiro indica que o tamanho, em número de leituras do conjunto de dados é 10. O segundo e terceiro campo, com valores iguais a zero e 10 respectivamente, indicam que as leituras começam a partir do momento que o TransducerChannel entra em modo de operação e o intervalo entre cada leitura. O quarto campo informa que esse intervalo é dado em segundos, usando a mesma representação definida para os dados do campo PhyUnits.

UpdateT: o valor 0 foi atribuído a este campo, já que estes sensores não terão a capacidade de operar no modo “free-running”.

RSetupT: como o sensor poderá ser lido a qualquer momento, foi fixado o valor 0 para o campo, indicando que não é necessário esperar para acessar o sensor.

SPeriod: o período de amostragem mínimo dos sensores é de 1 décimo de segundo, valor que é expresso no campo em segundos.

WarmUpT: assim que o TransducerChannel entrar em modo de operação o sensor já estará pronto para leitura. Assim, esse campo assume valor 0.

RDelayT: o intervalo entre o momento que um comando de leitura é recebido pelo TransducerChannel, até o momento que o mesmo entre em operação, não deverá ser maior do que meio segundo.

Sampling: este campo é formado por outros dois sub-campos, ambos obrigatórios. Os sensores de temperatura funcionarão no modo “streaming operation”, por isso ambos os campos recebem o valor três, indicando que ele funcionará no modo de amostragem “continuous sampling” e que esse será o modo padrão de amostragem.

DataXmit: como discutido anteriormente, o modo de transmissão escolhido para este transdutor é “streaming when a buffer is full”. Isso é indicado atribuindo-se o valor dois a este campo.

Tabela 4.4: Campos utilizados no TransducerChannel TEDS de um sensor

| Tipo | Nome | Descrição | Dados | Octetos |
|------|----------|--|---------|---------|
| — | Length | Comprimentos do TEDS | Uint32 | 4 |
| 3 | TEDSID | Cabeçalho de identificação do TEDS | Uint8 | 4 |
| 10 | CalKey | Capacidades de calibração do TransducerChannel | Uint8 | 1 |
| 11 | ChanType | Tipo do TransducerChannel | Uint8 | 1 |
| 12 | PhyUnits | Unidades físicas da leitura | UNITS | 11 |
| 13 | LowLimit | Limite mínimo operacional | Float32 | 4 |
| 14 | HiLimit | Limite máximo operacional | Float32 | 4 |
| 15 | Oerror | Incerteza existente no transdutor | Float32 | 4 |
| 16 | SelfTest | Descrição das capacidades de auto teste | Uint8 | 1 |
| 18 | Sample | Detalhes das leituras | — | — |
| 19 | DataSet | Detalhes do conjunto de dados | — | — |
| 20 | UpdateT | Intervalo de amostragem para o modo “free-running” | Float32 | 4 |
| 22 | RSetupT | Tempo de inicialização de leitura | Float32 | 4 |
| 23 | SPeriod | Tempo de leitura | Float32 | 4 |
| 24 | WarmUpT | Tempo de inicialização | Float32 | 4 |
| 25 | RDelayT | Tempo de espera de leitura | Float32 | 4 |
| 31 | Sampling | Atributos de leitura | — | — |
| 34 | DataXmit | Modo de transmissão de dados | Uint8 | 1 |
| — | Checksum | Checksum | Uint16 | 2 |

Desse modo, os octetos definidos para o TransducerChannel TEDS dos sensores de temperatura dos nós estão resumidos na tabela 4.5. O TEDS é encabeçado por quatro octetos descrevendo seu tamanho e terminado por dois octetos de checksum.

4.2.3 User’s Transducer Name TEDS

Os campos obrigatórios do User’s Transducer Name TEDS estão demonstrados na tabela 4.6. Para implementação dessas TLVs foram usados os seguintes valores:

TEDSID: o primeiro octeto desta TLV identifica que o formato deste TEDS é descrito pela norma IEEE 1451.0, recebendo valor 0. Através do valor 12, o segundo octeto identifica que esse TEDS faz parte da classe User’s Transducer Name TEDS. O

Tabela 4.5: Octetos do TransducerChannel TEDS de um sensor de temperatura

| Tipo | Comprimento | Valor |
|------|-------------|---|
| — | — | 0x00 0x00 0x00 0x77 |
| 0x03 | 0x04 | 0x00 0x03 0x01 0x01 |
| 0x0a | 0x01 | 0x00 |
| 0x0b | 0x01 | 0x00 |
| 0x0c | 0x06 | 0x32 0x01 0x00 0x39 0x01 0x82 |
| 0x0d | 0x04 | 0x43 0x69 0x00 0x00 |
| 0x0e | 0x04 | 0x43 0xb0 0x80 0x00 |
| 0x0f | 0x04 | 0x3f 0x00 0x00 0x00 |
| 0x10 | 0x01 | 0x00 |
| 0x12 | 0x09 | 0x28 0x01 0x01 0x29 0x01 0x04 0x2a 0x01 0x00 |
| 0x13 | 0x1c | 0x2b 0x02 0x00 0x0a 0x2c 0x04 0x00 0x00 0x00 0x00 0x2d 0x04 0x41 0x20 0x00 0x00 0x2e 0x06 0x32 0x01 0x00 0x37 0x01 0x82 |
| 0x14 | 0x04 | 0x00 0x00 0x00 0x00 |
| 0x16 | 0x04 | 0x00 0x00 0x00 0x00 |
| 0x17 | 0x04 | 0x3d 0xcc 0xcc 0xcd |
| 0x18 | 0x04 | 0x00 0x00 0x00 0x00 |
| 0x19 | 0x04 | 0x3f 0x00 0x00 0x00 |
| 0x1f | 0x06 | 0x30 0x01 0x03 0x31 0x01 0x03 |
| 0x22 | 0x01 | 0x02 |
| — | — | 0x00 0x00 |

terceiro octeto aponta que o TEDS esta de acordo com a versão mais recente da norma indicada, recebendo valor um. O quarto e último campo informa que todos os campos de tamanho da TLV terão tamanho igual a um.

Format: o nome do WTIM ou TransducerChannel será representado por um formato de texto definido pelo usuário. Portanto, seu valor é igual a 0.

Name: o formato escolhido para representação é a codificação UTF-8(YERGEAU, 1998). Para a identificação dos WTIMs foram escolhidos os nomes “Sun SPOT” para o nó Sun SPOT e “EPOS/Mica” para o nó Crossbow MICA. Os sensores de temperatura implementados em ambos os motes foram identificados com o nome “Onboard Temperature Sensor”.

Tabela 4.6: Campos obrigatórios do User’s Transducer Name TEDS

| Tipo | Nome | Descrição | Dados | Octetos |
|------|----------|------------------------------------|--------|---------|
| — | Length | Comprimentos do TEDS | Uint32 | 4 |
| 3 | TEDSID | Cabeçalho de identificação do TEDS | Uint8 | 4 |
| 4 | Format | Formato da descrição do TEDS | Uint8 | 1 |
| 5 | TCName | Nome do TIM ou TransducerChannel | — | 1 até N |
| — | Checksum | Checksum | Uint16 | 2 |

Os octetos definidos nesta implementação do User’s Transducer Name TEDS, para os TransducerChannel dos sensores de temperaturas de ambos os motes, estão mostrados na tabela 4.7.

Tabela 4.7: Octetos do User’s Transducer Name TEDS do sensor de temperatura

| Tipo | Comprimento | Valor |
|------|-------------|--|
| — | — | 0x00 0x00 0x00 0x27 |
| 0x03 | 0x04 | 0x00 0x0c 0x01 0x01 |
| 0x04 | 0x01 | 0x00 |
| 0x05 | 0x1a | 0x05 0x1a 0x4f 0x6e 0x62 0x6f 0x61 0x72 0x64 0x20 0x54 0x65 0x6d 0x70 0x65 0x72 0x61 0x74 0x75 0x72 0x65 0x20 0x53 0x65 0x6e 0x73 0x6f 0x72 |
| — | — | 0x00 0x00 |

4.2.4 PHY TEDS

Todos os campos das PHY TEDS são obrigatórios, com exceção daqueles voltados para um tipo de rádio específico. Os campos comuns a todos os tipos de comunicação física por rádio frequência podem ser vistos na tabela 4.8.

TEDSID: o campo dessa TLV descreve quatro atributos, cada um usando um octeto. O primeiro octeto, recebe o valor 5, indicando que esse TEDS é descrito pela norma IEEE 1451.5. Para o segundo octeto, é designado o valor 13, que indica que é um PHY TEDS que está sendo acessado. O terceiro e quarto valor são iguais a 1, simbolizando respectivamente que este TEDS é descrito pela versão mais nova da norma e que o tamanho dos campos de tamanho das TLVs será de 1 octeto.

Radio: já que o tipo dos rádio do Sun SPOT e Crossbow MICA não é melhor descrito ou suportado pela norma, esse campo recebe o valor de 255, notificando que o tipo do rádio é definido pelo fabricante.

MaxBPS: para o Sun SPOT o máximo throughput é de 250 Kbps. Para o Crossbow MICA esse valor é igual a 38,4 kbps.

MaxCDev: para esta implementação, os WTIMs só se conectaram ao NCA, não necessitando da capacidade de conectar a mais de um dispositivo. Portanto, o valor desta TLV é igual a um.

MaxRDev: sabido que apenas um dispositivo pode se conectar ao WTIM, então não há necessidade para que este campo seja diferente de 1.

Encrypt: as comunicações não serão cifradas. Logo, o valor deste campo é zero.

Authent: também não será usada autenticação, colocando o campo de valor em 0.

MinKeyL: como não será utilizada criptografia nas comunicações de rádio, essa TLV recebe valor 0.

MaxKeyL: recebe valor 0 pelo mesmo motivo da TLV MinKeyL, de não ser usada criptografia nas comunicações.

MaxSDU: o tamanho máximo de um comando recebido por um WTIM é 65536, já que o campo de tamanho é formado por 2 octetos. Entretanto, os notes escolhidos possuem restrições de memória, optando-se por restringir o tamanho máximo de uma mensagem para 255.

MinALat: para iniciar a transmissão a um dispositivo em que não se está conectado levará meio segundo, descrito em nanosegundos nesse campo.

MinTLat: fixou-se um valor teórico de 1000 nanosegundos para o tempo necessário para transmitir o menor pacote possível entre os dispositivos.

MaxXact: os WTIMs aceitarão o pedido de apenas uma operação por vez.

Battery: ambos os motes são alimentados por bateria, definindo-se o valor 1.

RadioVer: pelo fato de a versão do rádio usado não ser um requisito necessário nesta implementação, o valor definido é igual a 0.

MaxRetry: para os WTIMs, foi definido que após 5 tentativas de envio de um comando será efetuada uma operação para desconectar o canal de comunicação com o dispositivo.

Como exemplo, estão mostrados na tabela 4.9 os octetos para o TEDS do canal físico de comunicação usado nos Crossbow MICA, antecedidos pelo número total, mostrados nos 4 primeiros octetos, e terminados pelos 2 octetos de checksum.

4.3 Implementação dos WTIMs

O módulo WTIM contém a parte de comunicações por radiofrequência, os “Transducer Electronic Data Sheet”, a lógica para implementar a interface dos transdutores, o transdutor ou conexão ao transdutor e a conversão ou condicionamento do sinal do transdutor. Foram feitas duas implementações para este trabalho, compreendendo os motes Sun SPOT e Crossbow MICA. Nestes, os TEDS, montados anteriormente, são armazenados em memória de execução dos dispositivos. Visto que não existia a necessidade de que todos os comandos apresentados pela norma fossem utilizados, os WTIMs receberam apenas a implementação dos comandos obrigatórios. O acesso aos transdutores e às interfaces físicas de comunicação é feito através das bibliotecas disponibilizadas pelas próprias plataformas. Os WTIMs implementados seguem as regras gerais definidas pela norma IEEE 1451.5:

- Um WTIM pode estar registrado a somente um único NCAP.
- Um WTIM pode servir de interface para múltiplos transdutores.

Tabela 4.8: Campos obrigatórios do PHY TEDS

| Tipo | Nome | Descrição | Dados | Octetos |
|------|----------|--|---------|---------|
| — | Length | Comprimentos do TEDS | Uint32 | 4 |
| 3 | TEDSID | Cabeçalho de identificação do TEDS | Uint8 | 4 |
| 10 | Radio | Tipo do rádio | Uint8 | 1 |
| 11 | MaxBPS | Throughput máximo de dados | Uint32 | 4 |
| 12 | MaxCDev | Número máximo de dispositivos conectados permitidos | Uint16 | 2 |
| 13 | MaxRDev | Número máximo de dispositivos registrados permitidos | Uint16 | 2 |
| 14 | Encrypt | Tipo de criptografia e tamanho da chave | Uint16 | 2 |
| 15 | Authent | Suporte a autenticação | Boolean | 1 |
| 16 | MinKeyL | Tamanho mínimo da chave de autenticação ou criptografia | Uint16 | 2 |
| 17 | MaxKeyL | Tamanho máximo da chave de autenticação ou criptografia | Uint16 | 2 |
| 18 | MaxSDU | Tamanho máximo da mensagem para transferência | Uint16 | 2 |
| 19 | MinALat | Período mínimo para iniciar transmissão com dispositivo desconectado | Uint32 | 4 |
| 20 | MinTLat | Tempo mínimo para completar transferência de uma mensagem | Uint32 | 4 |
| 21 | MaxXact | Número máximo de transações simultâneas | Uint8 | 1 |
| 22 | Battery | Dispositivo é alimentado por bateria | Uint8 | 1 |
| 23 | RadioVer | Versão do rádio | Uint16 | 2 |
| 24 | MaxRetry | Número máximo de tentativas antes de desconectar | Uint16 | 2 |
| — | Checksum | Checksum | Uint16 | 2 |

Tabela 4.9: Octetos do PHY TEDS do Crossbow MICA

| Tipo | Comprimento | Valor |
|------|-------------|---------------------|
| — | — | 0x00 0x00 0x00 0x27 |
| 0x03 | 0x04 | 0x05 0x0d 0x01 0x01 |
| 0x0a | 0x01 | 0xff |
| 0x0b | 0x02 | 0x96 0x00 |
| 0x0c | 0x02 | 0x00 0x01 |
| 0x0d | 0x02 | 0x00 0x01 |
| 0x0e | 0x02 | 0x00 0x00 |
| 0x0f | 0x01 | 0x00 |
| 0x10 | 0x02 | 0x00 0x00 |
| 0x11 | 0x02 | 0x00 0x00 |
| 0x12 | 0x02 | 0x00 0xff |
| 0x13 | 0x04 | 0x1d 0xcd 0x65 0x00 |
| 0x14 | 0x04 | 0x00 0x00 0x03 0xe8 |
| 0x15 | 0x01 | 0x01 |
| 0x16 | 0x01 | 0x01 |
| 0x17 | 0x02 | 0x00 0x00 |
| 0x18 | 0x02 | 0x00 0x05 |
| — | — | 0x00 0x00 |

Ambas as implementações foram feitas não só em plataformas diferentes, mas também em linguagens de programação diferentes. No caso dos Crossbow MICA, utilizando o sistema operacional EPOS, o desenvolvimento foi feito em C++(ELLIS; STROUSTRUP, 1990). Para o Sun SPOT, o mesmo desenvolvimento foi feito utilizando a linguagem Java. Entretanto, as duas empregam o paradigma de orientação a objetos, sendo as classes criadas em cada uma das plataformas muito similar, como pode ser visto nos diagramas UMLs apresentados nas figuras 4.3 e 4.2.

A classe *IEEE1451dot5* é a parte correspondente ao módulo de comunicação, similar ao presente no NCAP. Em ambas as implementações do WTIM, existe um Observador (GAMMA et al., 1994)¹ atrelado à classe *IEEE1451dot5*, que é notificado de mensagens que chegam ao módulo de comunicação através da interface física. Esse ouvinte é a camada *IEEE1451dot0*, que tem a lógica responsável por receber as mensagens destinadas ao próprio TIM ou encaminhar as mensagens aos seus respectivos TransducerChannels de destino. Os TEDS de descrição geral do sistema, tais como o Meta-TEDS e o PHY TEDS, estão armazenados na classe *IEEE1451dot0*. Além disso, estão nesta classe, as referências

¹O Observador é um objeto de um programa de computador, pertencente ao padrão de projetos Observer. O padrão Observer permite que objetos interessados, os observadores, sejam avisados da mudança de estado ou outros eventos ocorrendo num outro objeto observável.

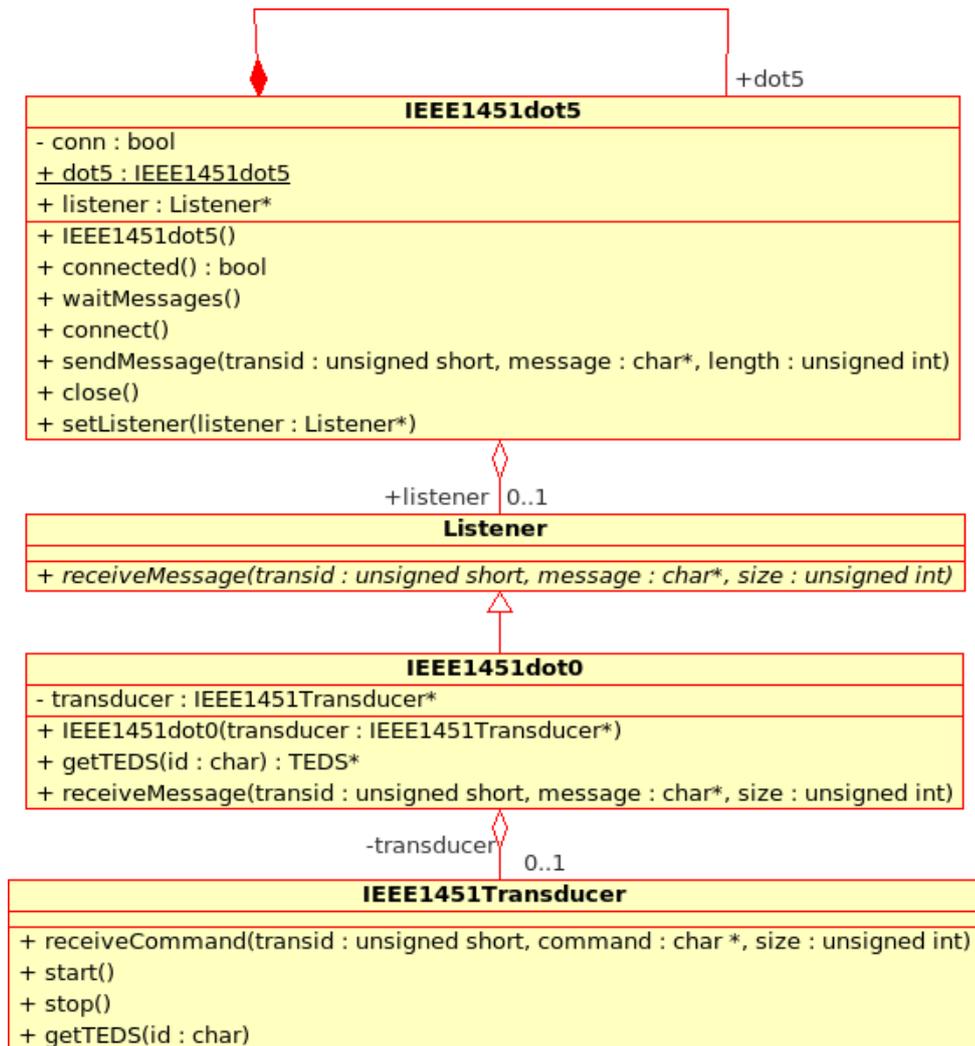


Figura 4.2: Diagrama UML do WTIM implementado no Crossbow MICA

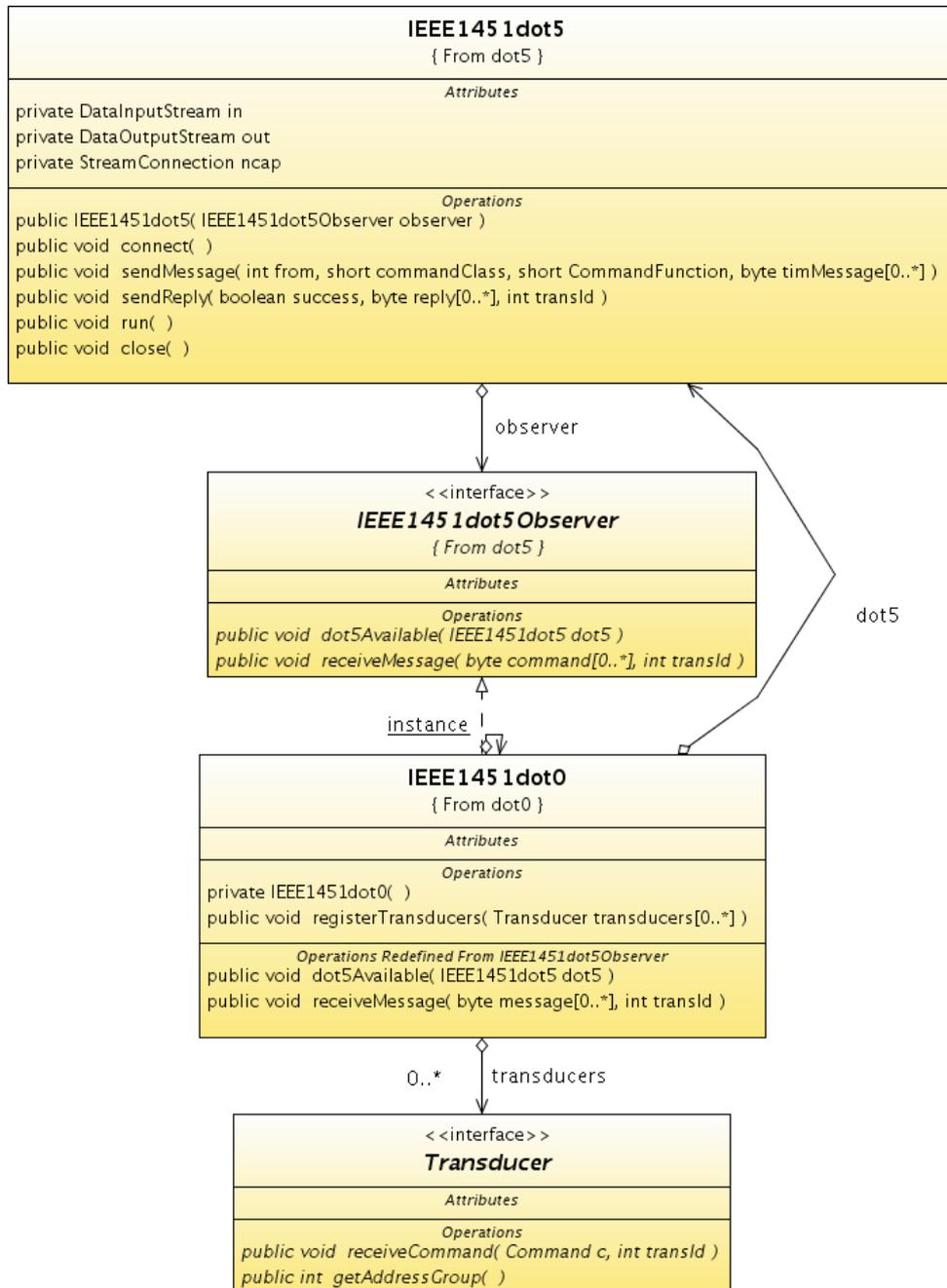


Figura 4.3: Diagrama UML do WTIM implementado no Sun SPOT

das implementações dos *TransducerChannels*, abstraídos através da interface *Transducer*. As implementações da interface *Transducer* armazenam os TEDS de cada transdutor que representam, além de possuírem a lógica necessária para interpretar e responder os comandos destinados a estes canais. Em poucas palavras, cada *TransducerChannel* existente no WTIM é uma implementação da interface *Transducer*.

No mote SPOT, uma MIDlet, ponto de entrada em programas Java ME CLDC, é responsável por colocar o nó em funcionamento, apresentando as implementações de *TransducerChannel* e do módulo de comunicação para a classe *IEEE1451dot0*. Semelhantemente, uma aplicação EPOS faz o mesmo papel no mote MICA. Uma vez que a aplicação é iniciada, o módulo de comunicação se registra ao primeiro NCAP que encontrar, podendo a partir daí, aceitar comandos e, caso venha ser solicitado, iniciar seu modo de operação.

Uma vez que as ferramentas EPOS geram software para o MICA em formato little endian, optou-se por armazenar os TEDS com a ordem dos bytes invertida, aproveitando a organização das próprias variáveis da linguagem de programação. Também, devido as restrições de hardware, optou-se pelo uso de uma única thread, diminuindo o tamanho do código gerado.

4.4 Implementação NCAP

Para implementar o NCAP, escolheu-se a utilização de plataformas e ferramentas que fossem acessíveis. Por tanto, o NCAP é formado por um computador pessoal, rodando sistema operacional Linux, e com software implementado usando a linguagem Java e rodando sobre uma plataforma Java. Cada norma da família IEEE 1451 implementada neste trabalho, foi representada por uma classe Java, tornando o sistema bastante modular. O diagrama UML, apresentado na figura 4.4, apresenta as principais classes que formam o NCAP.

A interface *IEEE1451dot5* é responsável por abstrair o módulo de comunicação estipulado pela norma IEEE 1451.5, existindo duas implementações no sistema, cada uma referente as duas redes de sensores sem fio utilizadas. A classe *EposIEEE1451dot5* é responsável por abstrair a rede formada pelos nós sensores Crossbow MICA rodando o sistema EPOS e a classe *SpotIEEE1451dot5* pela rede formada pelos motes Sun SPOT. Essas implementações dos módulos de comunicação se comunicam com os gateways das respectivas RSSF através da porta USB do computador. Este acesso, na rede formadas

pelos nós SPOT é feita pelas bibliotecas providas pelo próprio mote, enquanto no caso do Crossbow MICA é feito a partir de bibliotecas para acesso às portas seriais do computador. O gateway é responsável em fazer com que as mensagens cheguem aos nós destinatários, sem a necessidade de mais intervenções.

Uma vez que os módulos de comunicação são abstraídos pela interface *IEEE1451dot5*, é possível criar e incorporar implementações para outros meios de comunicação físico. Apesar de essa interface levar o nome da norma IEEE 1451.5, também é possível que o mesmo seja implementado seguindo outros padrões da família 1451 que tratam da parte de comunicação, como por exemplo, a IEEE 1451.2. Dessa maneira, o NCAP pode abranger vários de módulos de comunicação, seguindo as diferentes normas pertencentes à família IEEE 1451 ou tratando-se de diferentes implementações de uma mesma norma.

A *IEEE1451dot0* é a principal classe do sistema, encarregando-se da implementação do padrão IEEE 1451.0. Cabe a esta classe, manter e organizar todas as implementações *IEEE1451dot5* em funcionamento, assim como conhecer a aplicação NCAP que esta sendo utilizada. *Application* é uma interface que tem o objetivo de abstrair as aplicações NCAP. Essas aplicações são responsáveis pelas ações que o sistema deve tomar, tendo como meio de comunicação com os TIMs a classe *IEEE1451dot0*. Diferentemente das interfaces de software propostas pela norma, que são baseadas em operações de polling, este trabalho utiliza uma interface de software baseada em eventos. Toda mensagem recebida pelo NCAP, seja uma resposta a um comando ou uma mensagem iniciada pelo TIM, é entregue para a aplicação NCAP através da chamada de um método, descartando a necessidade que esta mesma aplicação continuamente consulte a classe *IEEE1451dot0*. Sempre que uma mensagem é enviada por uma aplicação NCAP para um TIM, cabe à *IEEE1451dot0* determinar qual implementação do módulo de comunicação deve ser corretamente utilizada, através do endereço do destinatário, para que a mensagem seja corretamente entregue. Toda mensagem enviada através da *IEEE1451dot0* recebe um número identificador (ID) que é informado ao remetente, de forma que a aplicação NCAP consiga distinguir respostas de diferentes comandos que tenham sido enviados. Além disso, a implementação do padrão IEEE 1451.0 tem o papel de agregar todas as mensagens destinadas ao NCAP e entrega-las à aplicação NCAP que está fixada para no sistema.

Afora as classes que abstraem as normas IEEE 1451, existem as chamadas classes utilitárias, que permitem, por exemplo, montar e interpretar os comandos de forma mais fácil.

Uma das características exigidas pela norma IEEE 1451.0 é que deve ser possível

desligar ou ligar um novo TIM ao sistema, sem que nenhum dano seja sofrido pelo NCAP. Esta característica, referenciada pela norma como Hot-Swap, faz parte do NCAP aqui implementado. Toda vez que um WTIM é ligado, ele é automaticamente reconhecido e incorporado ao sistema e ao seu desligamento, nenhum sofrimento é causado ao NCAP. Do mesmo modo, outras características impostas pela norma são seguidas, de forma a tornar este um NCAP compatível com o padrão:

- O NCAP utiliza as estruturas de mensagens especificadas;
- O NCAP utiliza os comandos especificados;
- O NCAP faz uso de um protocolo de comunicação e mídia física definido por um outro membro da família de padrões IEEE 1451;
- O NCAP suporta as especificações técnicas exigidas.

4.4.1 Aplicações NCAP

As aplicações NCAP são componentes que funcionam acima em conjunto da classe IEEE1451dot0, sendo responsáveis por efetuar as tarefas determinadas para o NCAP. Neste trabalho foram criadas duas aplicações NCAP, chamadas Cache e Main, conforme mostra o diagrama UML na figura 4.5. Sempre que um evento ocorre no NCAP, ele é notificado às aplicações através dos métodos da interface *Application*, implementada pelas aplicações NCAP. Os métodos providos pela interface são:

reportTIMConnected: este método é chamado toda vez que um TIM se registra no NCAP, seja a primeira vez ou n-ésima vez que o registro é efetuado. Para identificar o TIM que causou o evento, é passado como parâmetro o endereço do dispositivo, que é determinado conforme a rede a que ele pertence. Cabe à aplicação o controle de quantas vezes o TIM se conectou ao sistema.

reportTIMDisconnected: sempre que um TIM se desconectar do sistema, seja por que ele tenha sido desligado ou perdido comunicação, a aplicação é informada através deste método, que tem como parâmetro o endereço do TIM.

reportCommandReply: chamado sempre que é recebida uma resposta a um comando enviado pela aplicação, informando o identificador da mensagem original, o endereço do remetente e os dados contidos na resposta.

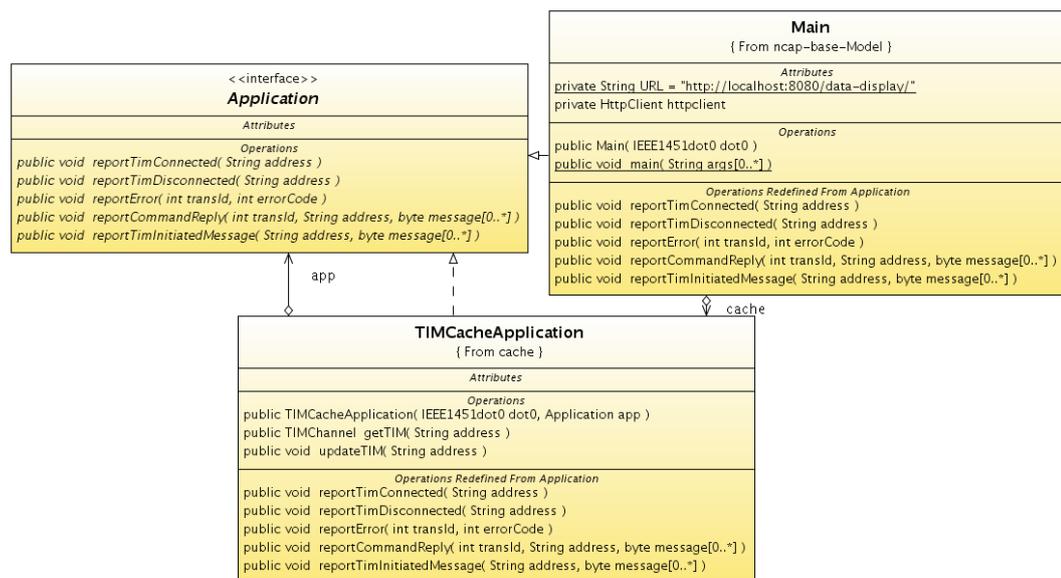


Figura 4.5: Diagrama UML das aplicações NCAP

reportTimInitiatedMessage: as mensagens iniciadas pelos TIMs e destinadas ao NCAP são entregues à aplicação através deste método. Nele, é informado através dos parâmetros o endereço do remetente, assim como o conteúdo da mensagem recebida.

reportError: os erros ocorridos internamente, nas camadas inferiores, são notificados à aplicação através deste método. Os erros podem ser os mais diversos, como por exemplo, a falha na entrega de uma mensagem a um TIM. Caso o erro ocorrido seja de comunicação, é passado também o identificador da mensagem que ocasionou o erro, além do número do erro ocorrido.

A aplicação chamada Cache tem a responsabilidade de enviar comandos para obter todos os TEDS obrigatórios do WTIM e dos TransducerChannels existentes. Uma vez que todos os TEDS tenham sido obtidos e armazenados, a aplicação notifica a próxima aplicação, Main, da presença de um novo WTIM. Então, Main envia um comando para que o TransducerChannel do WTIM que acabou de se registrar entre em modo de operação. Atribui-se também à Main a tarefa de interpretar os TEDS e os dados recebidos, enviando essas informações para o servidor de dados.

Quando um WTIM é ligado e se registra ao NCAP, são funções das aplicações, em ordem de acontecimento, as seguintes:

1. Obter e armazenar os TEDS obrigatórios do WTIM (Cache);

2. Interpretar os TEDS (Main);
3. Notificar a aplicação web de um novo WTIM (Main);
4. Enviar o comando para colocar os TransducerChannels do WTIM em funcionamento (Main);
5. Receber as mensagens iniciadas pelo WTIM contendo as leituras dos sensores (Main);
6. Interpretar as leituras e enviá-las para o servidor de dados (Main).

4.5 Módulos de Comunicação sem fio

Este trabalho utiliza uma das implementações dos módulos de comunicação mais simples previstas pela norma, onde um ou mais TIMs se comunicam somente, e diretamente, com o NCAP em que se registraram. Os módulos de comunicação implementados neste trabalho são guiados pela norma IEEE 1451.5, entretanto não utilizam as APIs propostas pela norma, uma vez que estas são opcionais. Visto que os aspectos físicos da interface não estão descritos na norma, utilizaram-se os suportes de comunicação providos pela plataforma de cada um dos nós sensores utilizados.

O WTIM contém uma Dot5AR para comunicação sem fio, similar ao apresentado pelo NCAP, completando logicamente a ligação sem fio entre WTIM e NCAP. O Dot5AR do NCAP suporta múltiplas conexões a WTIMs com o mesmo tipo de rádio. Pode haver múltiplas Dot5ARs sobre um NCAP, como é o caso deste trabalho, onde o NCAP possui implementações Dot5AR para a rede formada pelos nós Sun SPOT e pelos nós Crossbow MICA. Já o Dot5AR dos WTIMs é dedicado, tendo um único canal de comunicação física.

Com a utilização de mensagens iniciadas pelo TIM, surge o problema de uma possível confusão que poderia ser causada entre essas mensagens e simples respostas de comandos enviados pelo NCAP. Além disso, opcionalmente a camada de comunicação pode suportar o envio de múltiplas mensagens ao mesmo tempo, cada uma com um identificador de mensagem diferente. Portanto, convencionou-se enviar juntamente com a mensagem um número identificador. Quando uma mensagem do tipo comando é enviada do NCAP ao TIM através das interfaces de software providas, o identificador atribuído é um valor fornecido pelo NCAP. Ao responder os comandos, o TIM usa esse identificador que para enviar a resposta. Já para mensagens iniciadas pelo TIM, define-se que esse valor é sempre igual a zero, permitindo assim ao NCAP diferenciar comandos NCAP de comandos TIM através deste valor.

4.6 Servidor de Dados

As leituras dos sensores, recebidas pelo NCAP, são dirigidas a um servidor Web, para que sejam armazenadas e melhor visualizadas através da geração de gráficos. Através de requisições, utilizando o protocolo de comunicação HTTP, o servidor é informado da conexão ao NCAP de novos TIMs e dos dados coletados e enviados pelos mesmos. Esse servidor não tem relação com a utilização do protocolo HTTP apresentada na norma IEEE 1451.0, mas permite que os dados que tenham sido enviados para o NCAP sejam acessados através do protocolo HTTP em navegadores Web.

Para programação do servidor web foi utilizada a tecnologia de Servlets que, quando executados dentro de um container Web², insere novos recursos ao servidor, permitindo ao desenvolvedor de software adicionar conteúdo dinâmico utilizando a linguagem Java. O Jetty (CONSULTING,), é um container web baseado em padrões, totalmente implementado em Java e que tem como filosofia ser uma ferramenta de fácil compreensão e utilização. É distribuído sob a licença Apache 2.0 e, portanto, é livre para uso comercial e distribuição. Essas características do Jetty o tornaram uma ótima opção para uso neste trabalho.

Através de duas URIs, cada uma mapeada para uma Servlet diferente, é possível tanto que as aplicações NCAP notifiquem os eventos ocorridos, quanto que os clientes façam o acesso aos dados contidos no servidor. Essas URIs são acessadas através de requisições HTTP feitas à porta 8080 e para endereço IP ou hostname da máquina em que o servidor web reside. As URIs são baseadas na arquitetura REST (FIELDING; TAYLOR, 2000), sendo cada recurso endereçável unicamente, utilizando uma sintaxe universal para uso nos links hipermídia. Os Servlets, e as URIs correspondentes de cada um, são:

TIMServlet: (*/data-display/tim*) Quando a aplicação NCAP é notificada do registro de um novo TIM, uma requisição HTTP utilizando o método POST é feita para o TIMServlet, juntamente com os parâmetros descrevendo informações do TIM, tais como nome, endereço e número de transdutores. Também é passado como parâmetro o nome e as unidades de leitura de cada TransducerChannel presente no novo TIM. Como resposta, um identificador único daquele TIM é informado para que as informações do mesmo possam ser acessadas posteriormente. Ao executar uma requisição para TIMServlet, utilizando o método GET, a lista dos TIMs disponíveis

²Container Web é um servidor web especializado com a capacidade de executar Servlets. Ele é responsável por mapear URLs para uma Servlet particular, administrar o ciclo de vida dos Servlets e assegurar que quem tenha feito a requisição para uma dada URL tenha permissão para tanto.

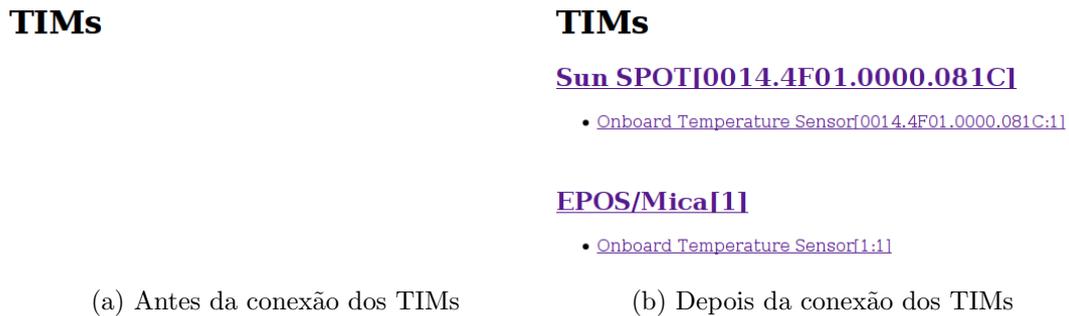


Figura 4.6: Página exibindo TIMs conectados ao Servidor

no servidor de dados é exibida em um documento HTML, conforme mostrado na figura 4.6. Caso o identificador do TIM seja informado como último elemento do URI, apenas as informações sobre o TIM específico serão retornadas como resposta à requisição.

TransducerServlet: (*/data-display/transducer*) As leituras de cada sensor, recebidas pela aplicação NCAP, são repassadas, através de um método POST, para a URL mapeada para o TransducerServlet. Os dois últimos elementos do URI requisitada devem ser, respectivamente, o identificador do TIM e o número do Transducer-Channel responsável pelo envio dos dados. Quando um método GET é feito para o TransducerServlet, a resposta é uma página HTML contendo o gráfico com todas as leituras já enviadas por aquele sensor, mostrada na figura 4.7.

Assim, os Servlets podem ser acessados com URLs no formato *http://localhost:8080/data-display/tim*, que exibe os TIMs conectados ao NCAP.

Onboard Temperature Sensor [0014.4F01.0000.081C:1]

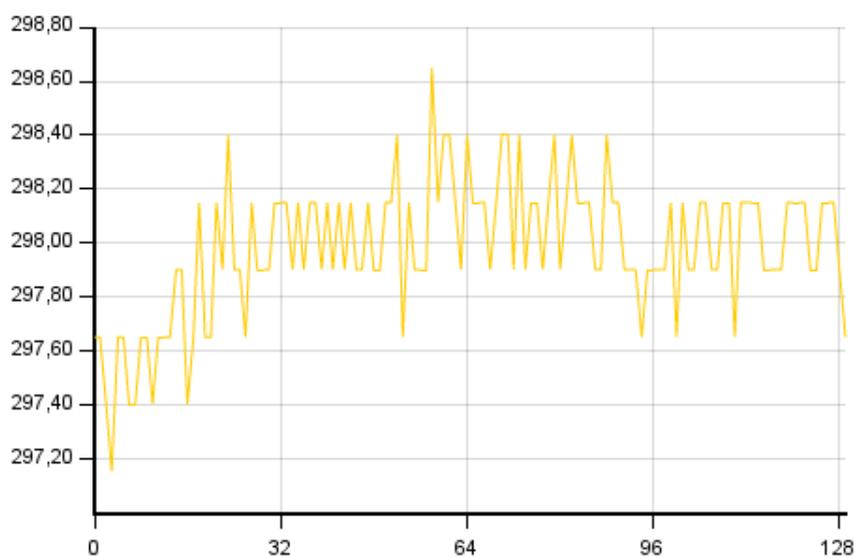


Figura 4.7: Gráfico gerado pelo servidor Web com as leituras de um sensor de temperatura

5 *Conclusão*

Neste trabalho foram estudadas as normas pertencentes à família IEEE 1451, além de empregá-las, quando possível, em redes de sensores sem fio. As RSSF são atualmente um dos principais temas da área tecnológica e apesar de haverem distintas soluções e plataformas para tais, através da padronização introduzida pela IEEE 1451, mais especificamente os padrões 1451.0 e 1451.5, foi possível integrar diferentes redes, formadas por diferentes motes e tecnologias de comunicação sem fio, permitindo a configuração automática e compreensão das leituras coletadas uma vez que estes nós tenham se conectado ao sistema.

Mesmo se tratando de um padrão extremamente complexo, as características oferecidas pelas normas pertencentes a 1451 trazem diversas vantagens e melhorias quando utilizadas juntamente com as RSSF. Como este trabalho demonstrou, a adição de novos nós sensores torna-se um processo fácil e automático, uma vez que um novo nó pode ser colocado em funcionamento apenas ligando-o à rede, permitindo assim, uma expansão rápida e eficiente de redes existentes. Além disso, a existência de um conjunto padronizado de interfaces, estruturas e formatos de comunicação, tornou possível manter interoperabilidade entre os diferentes motes utilizados e que o sistema pudesse reconhecer e trabalhar com novos sensores de maneira ágil.

Este trabalho também mostrou que a implementação dos padrões 1451 pode ser executada em diferentes tipos de sistemas, não só a adoção com recursos limitados de hardware, mas também com diferentes linguagens de programação. Sem dúvida nenhuma, um dos principais adendos da IEEE 1451 são os TEDS. Os TEDS simplificaram muitos dos aspectos da criação de redes de sensores sem fio, não sendo mais necessário o sistema adivinhar os novos sensores que se conectam ao sistema, já que aos mesmos foi acrescentada a capacidade de se auto-descreverem. Também, mostra-se muito importante que os fabricantes adotem descrição dos sensores comercializados, diminuindo o esforço necessário para incorporação de diferentes e novos sensores ao sistemas existentes.

Apesar de a IEEE 1451 não ser de aplicação exclusiva das RSSF, fazendo com que muitas vezes ela não contemple as limitações impostas por esta tecnologia, a utilização dos conceitos existentes na família permitiu que, através da estrutura de “TIM Initiated Messages” e utilizando o modo de operação correto, os WTIMs conectados ao NCAP enviassem as informações coletadas de forma autônoma, não havendo a necessidade de sucessivas interrogações aos WTIMs registrados no NCAP.

Em poucas palavras, as vantagens apresentadas pela família IEEE 1451 tornam-na uma ótima opção para utilização em conjunto com as redes de sensores sem fio.

6 Trabalhos Futuros

Pelo fato de as normas utilizadas neste trabalho, IEEE 1451.0 e IEEE 1451.5, serem muito recentes, existem ainda poucas referências sobre o assunto. Por tanto, não se descarta a possibilidade de haverem equívocos de implementação referentes ao mau entendimento das normas. Isto talvez possa tornar esta implementação não robusta o suficiente para uso em aplicações comerciais. Mas de qualquer forma, isto não inibe a possibilidade de ampliar o desenvolvimento dos componentes aqui construídos.

Ainda que os TEDS sejam uma das principais características e vantagens da IEEE 1451, sua construção é um processo repetitivo e demorado, cabendo, quando não presentes em sensores que atendem a norma, serem montadas pelo programador ou usuário do sistema. Dessa forma, torna-se importante a necessidade de uma ferramenta específica para a montagem dos TEDS, facilitando e guiando o processo de montagem dos mesmos.

Também não foi discutido neste trabalho como agregar e enviar por mensagens iniciadas pelo TIM informações coletadas quando na existência de mais de um sensor no mesmo WTIM. Entretanto, isto pode ser efetuado através da lógica de combinação de conjunto de dados em um TransducerChannel proxy, apresentada na norma IEEE 1451.0.

Referências Bibliográficas

- ABOELAZE, M.; ALOUL, F. Current and future trends in sensor networks: a survey. *Wireless and Optical Communications Networks, 2005. WOCN 2005. Second IFIP International Conference on*, p. 551–555, 2005.
- AKYILDIZ, I. F. et al. A survey on sensor networks. *Communications Magazine, IEEE*, v. 40, n. 8, p. 102–114, 2002.
- AKYILDIZ, I. F. et al. Wireless sensor networks: a survey. *Computer Networks*, Elsevier, v. 38, n. 4, p. 393–422, 2002.
- ARSENEAU, E. et al. Simplifying the development of sensor applications.
- CONSULTING, M. *Jetty Java HTTP Servlet Server*. Disponível em: <<http://www.mortbay.org/jetty/>>.
- COULOURIS, G.; DOLLIMORE, J.; KINDBERG, T. *Distributed Systems: Concepts and Design (4th Edition)*. 4. ed. [S.l.]: Addison Wesley, 2005. 944 p. ISBN 0321263545.
- COURSES, E.; SURVEYS, T. Networked sensor monitoring using the universal IEEE 1451 Standard. *Instrumentation & Measurement Magazine, IEEE*, v. 11, n. 2, p. 18–22, 2008.
- CULLER, D.; ESTRIN, D.; SRIVASTAVA, M. Guest editors' introduction: Overview of sensor networks. *Computer*, v. 37, n. 8, p. 41–49, 2004.
- ELLIS, M. A.; STROUSTRUP, B. *The annotated C++ reference manual*. [S.l.]: Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1990. 480 p. ISBN 0201514591.
- FIELDING, R.; TAYLOR, R. Principled design of the modern Web architecture. p. 407–416, 2000.
- FROHLICH, A. A. Application-oriented operating systems. *Sankt Augustin: GMD-Forschungszentrum Informationstechnik*, v. 1, 2001.
- GAMMA, E. et al. *Design Patterns: Elements of Reusable Object-Oriented Software*. [S.l.]: Addison-Wesley Reading, MA, 1994. 416 p. ISBN 0201633612.
- GARCÍA-HERNÁNDEZ, C. F. et al. Wireless sensor networks and applications: a survey. *IJCSNS*, v. 7, n. 3, p. 264, 2007.
- GOSLING, J. et al. *Java Language Specification: The Java Series*. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 2000.

HADIM, S.; MOHAMED, N. Middleware: middleware challenges and approaches for wireless sensor networks. *Distributed Systems Online, IEEE*, v. 7, n. 3, 2006. ISSN 1541-4922.

HILL, J.; CULLER, D. Mica: A Wireless Platform for Deeply Embedded Networks. *IEEE MICRO*, IEEE Computer Society, p. 12–24, 2002.

IEEE Standard for a Smart Transducer Interface for Sensors and Actuators - Common Functions, Communication Protocols, and Transducer Electronic Data Sheet (TEDS) Formats. *IEEE Std 1451.0-2007*, p. 1–335, Sep 2007. Revision of 1451.1-1999.

IEEE Standard for A Smart Transducer Interface for Sensors and Actuators - Mixed-Mode Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats. *IEEE Std 1451.4-2004*, p. 1–430, Mar 2004.

IEEE standard for a smart transducer interface for sensors and actuators - transducer to microprocessor communication protocols and Transducer Electronic Data Sheet (TEDS) formats. *IEEE Std 1451.2-1997*, Sep 1998.

IEEE Standard for a Smart Transducer Interface for Sensors and Actuators-Digital Communication and Transducer Electronic Data Sheet (TEDS) Formats for Distributed Multidrop Systems. *IEEE Std 1451.3-2003*, p. 1–175, 2004.

IEEE Standard for a Smart Transducer Interface for Sensors and Actuators-Network Capable Application Processor (NCAP) Information Model. *IEEE Std 1451.1-1999*, 2000. Superseded by 1451.0-2007.

IEEE Standard for a Smart Transducer Interface for Sensors and Actuators Wireless Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats. *IEEE Std 1451.5-2007*, p. C1–236, Oct 2007.

IEEE standard for binary floating-point arithmetic. *ANSI/IEEE Std 754-1985*, Aug 1985. Superseded by 754-2008.

KHEMAPECH, I.; DUNCAN, I.; MILLER, A. A survey of wireless sensor networks technology. In: *Proceedings of the 6th Annual PostGraduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting: PGNET 2005*. [S.l.: s.n.], 2005.

KIM, S. et al. Health monitoring of civil infrastructures using wireless sensor networks. *Proceedings of the 6th international conference on Information processing in sensor networks*, ACM Press New York, NY, USA, p. 254–263, 2007.

KULARATNA, N.; SUDANTHA, B. An Environmental Air Pollution Monitoring System Based on the IEEE 1451 Standard for Low Cost Requirements. *Sensors Journal, IEEE*, v. 8, n. 4, p. 415–422, 2008.

LEE, K. Ieee 1451: A standard in support of smart transducer networking. In: *Instrumentation and Measurement Technology Conference, 2000. IMTC 2000. Proceedings of the 17th IEEE*. [S.l.: s.n.], 2000. v. 2, p. 525–528 vol.2.

- LEE, K.; SONG, E. Wireless sensor network based on iee 1451.0 and iee 1451.5-802.11. *Electronic Measurement and Instruments, 2007. ICEMI'07. 8th International Conference on*, p. 4–7, 2007.
- LEVIS, P. et al. Tinyos: An operating system for sensor networks. *Ambient Intelligence*, Springer, p. 115–148, 2005.
- MAINWARING, A. et al. Wireless sensor networks for habitat monitoring. *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, ACM New York, NY, USA, p. 88–97, 2002.
- MARCONDES, H. et al. Epos: Um sistema operacional portavel para sistemas profundamente embarcados. *Proceedings of the Fourth Brazilian Workshop on Operating Systems*, p. 796–804, 2007.
- MICROSYSTEMS, S. *Connected, Limited Device Configuration. Java 2 Platform Micro Edition. Version 1.0*. Specification, Sun Microsystems, 2000. Disponível em: <<http://jcp.org/en/jsr/detail?id=30>>.
- NEMETH-JOHANNES, J.; SWEETSER, V.; SWEETSER, D. Implementation of an iee-1451.0/1451.5 compliant wireless sensor module. *Autotestcon, 2007 IEEE*, p. 364–371, 2007.
- OOSTDYK, R. et al. A Kennedy Space Center implementation of IEEE 1451. In: SPIE. *Proceedings of SPIE*. [S.l.], 2006. v. 6222, p. 62220N.
- RAMOS, H.; RAMOS, P.; PACES, P. Development of a IEEE 1451 Standard Compliant Smart Transducer Network with Time Synchronization Protocol. In: *Instrumentation and Measurement Technology Conference Proceedings, 2007 IEEE*. [S.l.: s.n.], 2007. p. 1–6.
- REV, A. *MTS/MDA Sensor and Data Acquisition Board User's Manual*. [S.l.], 2005.
- ROMER, K.; MATTERN, F. The design space of wireless sensor networks. *Wireless Communications, IEEE [see also IEEE Personal Communications]*, v. 11, n. 6, p. 54–61, 2004.
- SCHMID, T.; DUBOIS-FERRIERE, H.; VETTERLI, M. Sensorscope: Experiences with a wireless building monitoring sensor network. In: *Workshop on Real-World Wireless Sensor Networks (REALWSN'05)*. [S.l.: s.n.], 2005.
- SIMON, D.; CIFUENTES, C. The squawk virtual machine: Java on the bare metal. *Conference on Object Oriented Programming Systems Languages and Applications*, ACM New York, NY, USA, p. 150–151, 2005.
- SIMON, D. et al. Java on the bare metal of wireless sensor devices: the squawk java virtual machine. *Proceedings of the 2nd international conference on Virtual execution environments*, ACM Press New York, NY, USA, p. 78–88, 2006.
- SONG, E.; LEE, K. An implementation of the proposed IEEE 1451.0 and 1451.5 standards. In: *Sensors Applications Symposium, 2006. Proceedings of the 2006 IEEE*. [S.l.: s.n.], 2006. p. 72–77.

- SONG, E. Y.; LEE, K. Understanding iee 1451-networked smart transducer interface standard-what is a smart transducer? *Instrumentation & Measurement Magazine, IEEE*, v. 11, n. 2, p. 11–17, 2008.
- SZEWCZYK, R. et al. An analysis of a large scale habitat monitoring application. *Proceedings of the 2nd international conference on Embedded networked sensor systems*, ACM New York, NY, USA, p. 214–226, 2004.
- TILAK, S.; ABU-GHAZALEH, N. B.; HEINZELMAN, W. A taxonomy of wireless micro-sensor network models. *ACM SIGMOBILE Mobile Computing and Communications Review*, ACM New York, NY, USA, v. 6, n. 2, p. 28–36, 2002.
- WADE, R.; MITCHELL, W. M.; PETTER, F. Ten emerging technologies that will change the world. *Technology Review*, v. 106, n. 1, p. 22–49, 2003. Disponível em: <<http://www.technologyreview.com/Infotech/13060/page2/>>.
- WANNER, L. F. The epos system supporting wireless sensor networks applications. p. 53, 2003.
- WANNER, L. F. et al. Operating system support for data acquisition in sensor networks. *Emerging Technologies and Factory Automation, 2006. ETFA'06. IEEE Conference on*, p. 582–585, 2006.
- WARNEKE, B. et al. Smart Dust: Communicating with a Cubic-Millimeter Computer. *COMPUTER*, IEEE Computer Society, p. 44–51, 2001.
- WARNEKE, B. A.; PISTER, K. S. J. Mems for distributed wireless sensor networks. *International Conference on Electronics, Circuits and Systems*, p. 15–18, 2002.
- WEIGHTS, I. B. of et al. *The International System of Units (SI)*. Nist special publication 330 2008 edition. [S.l.]: US Dept. of Commerce, Technology Administration, National Institute of Standards and Technology, 2008. 96 p. United States version of the English text of the eighth edition (2006) of the International Bureau of Weights and Measures publication - Le Système International d' Unités (SI).
- WEISER, M. The computer for the 21 st century. *ACM SIGMOBILE Mobile Computing and Communications Review*, ACM New York, NY, USA, v. 3, n. 3, p. 3–11, 1999.
- WERNER-ALLEN, G. et al. Monitoring volcanic eruptions with a wireless sensor network. *Wireless Sensor Networks, 2005. Proceedings of the Second European Workshop on*, p. 108–120.
- WOBSCHELL, D. Wireless gas monitor with IEEE 1451 protocol. In: *Sensors Applications Symposium, 2006. Proceedings of the 2006 IEEE*. [S.l.: s.n.], 2006. p. 162–164.
- WOBSCHELL, D.; MUPPARAJU, S. Low-power wireless sensor with SNAP and IEEE 1451 protocol. In: *Sensors Applications Symposium, 2008. SAS 2008. IEEE*. [S.l.: s.n.], 2008. p. 225–227.
- XU, N. A survey of sensor network applications. *University of Southern California*, <http://enl.usc.edu/ningxu/papers/survey.pdf>, 2003.

YERGEAU, F. *RFC2279: UTF-8, a transformation format of ISO 10646*. RFC Editor, United States, 1998. Disponível em: <<http://portal.acm.org/citation.cfm?id=RFC2279>>.

YURISH, S. Y. Ieee 1451 standard and frequency output sensors: How to obtain a broad-based industry adoption? *Sensors & Transducers Magazine*, v. 59, n. 9, p. 412–418, 2005.

APÊNDICE A -- Artigo

A norma IEEE 1451 aplicada a redes heterogêneas de sensores sem fio

Lucas B. Torri¹

1

lucastorri@gmail.com

Abstract. *This paper presents a study about the IEEE 1451 family and its application in wireless sensor networks, addressing the constraints presented by WSNs and enabling motes, known in the 1451 family as WTIMs, to take the initiative of communicating with the network gateway. Key-words: Wireless Sensor Networks, WSN, IEEE 1451, WTIM*

Resumo. *Esse trabalho apresenta um estudo feito sobre a família IEEE 1451 e a sua aplicação a redes de sensores sem fio, de forma a contemplar as restrições apresentadas pelas RSSF e permitir que os nós sensores, conhecidos na família IEEE 1451 como WTIMs, sejam os responsáveis por tomar a iniciativa nas comunicações com o gateway de rede. Palavras-chave: Redes de Sensores sem Fio, RSSF, IEEE 1451, WTIM*

1. Introdução

As redes de sensores sem fio, ou RSSF, são sistemas autônomos formados por dispositivos micro-processados, conhecidos como nós sensores, dotados de sensores e da capacidade de comunicação sem fio [Tilak et al. 2002]. Estes dispositivos podem ser distribuídos em determinado ambiente, monitorando cooperativamente os fenômenos que ocorrem nesse meio, e, permitindo assim, utilizar as informações coletadas para entender e tomar decisões adequadas sobre o ambiente estudado.

Com a evolução da tecnologia, foram criadas diversas plataformas para facilitar o desenvolvimento e criação das RSSF, cada uma com diferentes conceitos, abstrações, formas de manipulação de dados, algoritmos, e combinações hardware [Romer and Mattern 2004, Wanner 2003].

2. IEEE 1451

A fim de atenuar o problema causado pela heterogeneidade em sistemas dotados de transdutores, foi criada a família de padrões IEEE 1451 [Lee 2000]. Os padrões têm como intuito definir um conjunto de interfaces para conectar transdutores a sistemas e instrumentos de uma maneira independente de redes de comunicação, simplificando o desenvolvimento de dispositivos inteligentes e de controle de múltiplos transdutores.

Dentre as normas pertencentes à família, podemos destacar a IEEE 1451.0 e a IEEE 1451.5. A IEEE 1451.0 serve como base para as demais normas, introduzindo a arquitetura geral do sistema, composta de dois tipos de dispositivos: NCAPs e TIMs [Song and Lee 2008]. O NCAP é responsável por executar aplicações de processamento e comunicação com a rede externa, enquanto o TIM é um dispositivo consistindo de sensores e atuadores, condicionadores de sinais e conversores de dados, e TEDS. Os TEDS,

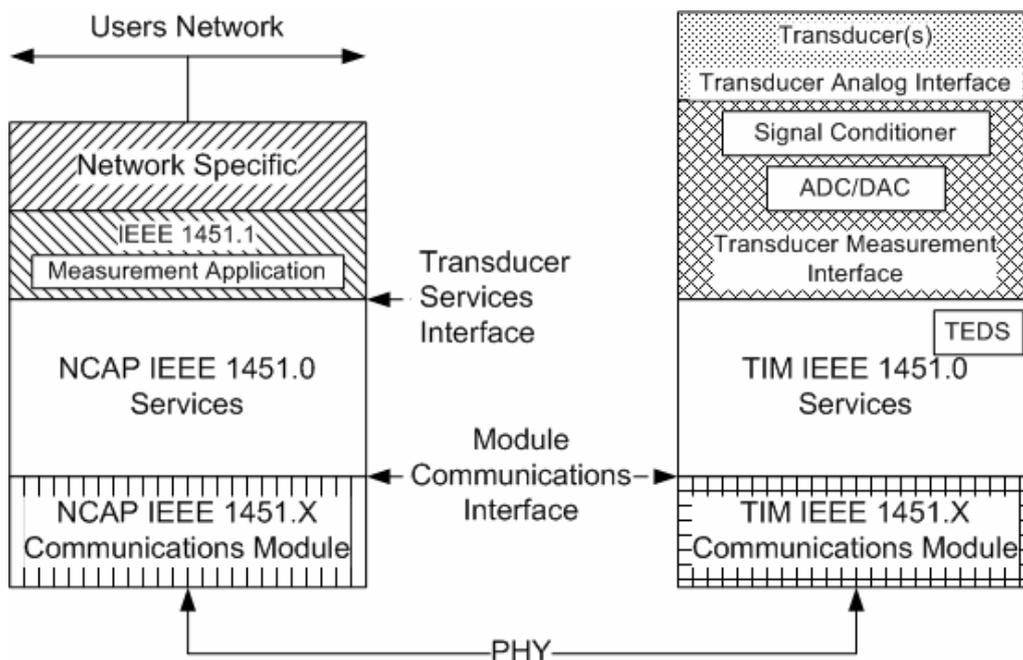


Figure 1. Modelo de referência proposto pela IEEE 1451.0[iee 2007]

um dos principais conceitos da IEEE 1451, são estruturas com a função de identificar e descrever as características do TIM e dos transdutores nele contidos. Além disso, a IEEE 1451.0 define modos de operação, que descrevem com será o funcionamento dos transdutores e a transmissão dos seus dados com o NCAP. A comunicação entre o NCAP e os TIMs é definida pelas outras normas da família, como por exemplo a IEEE 1451.5, que define canais de comunicação física através de sinais de rádio-frequência, além de descrever TEDS específica para este canal. O modelo proposto pela IEEE 1451.0 pode ser visto na figura 1.

3. Desenvolvimento da IEEE 1451 em RSSF

A família de padrões IEEE 1451 foi criada com o intuito de ser aplicável a qualquer sistema dotado de transdutores, portanto, podendo ser aplicada a redes de sensores sem fio. Comparando o modelo de objetos proposto pela norma IEEE 1451.0 com o contexto de RSSF, poderíamos igualar o NCAP a um gateway, e o TIM aos nós sensores.

Seguindo as informações contidas nas normas IEEE 1451.0 e IEEE 1451.5, definiu-se para implementação a arquitetura apresentada na figura 2. Nela, são apresentadas duas RSSF, formadas pelos nós Sun SPOT e o Crossbow MICA rodando o sistema EPOS. Esses nós são na verdade TIMs, contendo em funcionamento apenas um sensor de temperatura em cada um deles. O NCAP é um software escrito em linguagem Java e funcionando em computador pessoal e executando duas aplicações NCAP. Essas aplicações têm respectivamente a função de obter e armazenar os TEDS dos TIMs conectados, e colocar em funcionamento, processar e enviar ao servidor Web as informações obtidas pelos sensores. O servidor Web armazena os dados coletados pelos sensores e pode exibí-los através de páginas HTML.

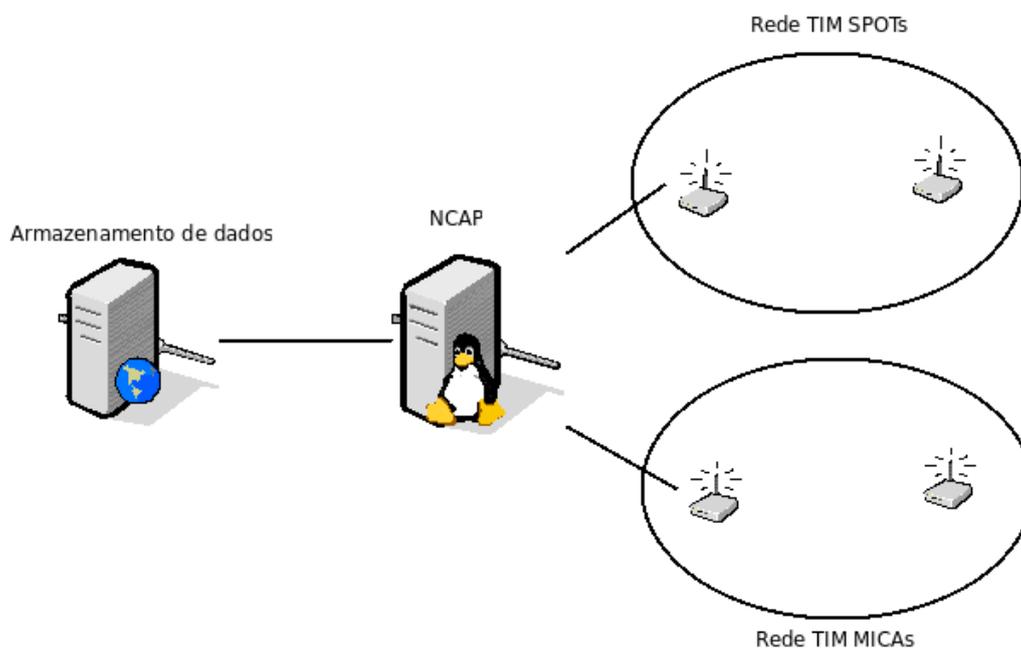


Figure 2. Arquitetura definida para a implementação das normas IEEE 1451

3.1. Modo de operação

Um dos principais gargalos das RSSF, é a restrição de energia imposta. Portanto, certas aplicações optam por manter os nós sensores em estado de dormência, religando-os em determinados intervalos de tempo e efetuando leituras, processamento e comunicação com os outros nós [Mainwaring et al. 2002, Szewczyk et al. 2004]. Essa forma de operação mostra uma tendência onde os motes são responsáveis por tomarem a iniciativa de comunicação com o gateway da rede a qual eles pertencem. Entretanto, em muitas implementações existentes da IEEE 1451, essas ações costumam ser tomada pelo NCAP [Lee and Song 2007].

Para contemplar este modo de operação, utilizou-se o modo “streaming operation“, que determina que um sensor deve adquirir dados e transmiti-los ao NCAP assim que o conjunto de leituras tenha sido preenchido, sem a necessidade de receber comandos adicionais. Isso permite que as leituras feitas por um sensor sejam enviadas ao NCAP de forma autônoma e automática. Infelizmente, a norma não descreve qual estrutura deve ser utilizada para enviar os dados coletados. Para solucionar essa questão, utilizaram-se neste trabalho as estruturas de “TIM Initiated Message“ descritas pela IEEE 1451.0. Assim, uma vez que o sensor entre em modo de operação e complete as leituras de um conjunto de dados, uma ”TIM Initiated Message“ é enviada ao NCAP, preenchida conforme a tabela 1, contendo todas as leituras efetuadas.

Os dois primeiros octetos da estrutura são formados pelo endereço do TransducerChannel que está enviando a mensagem. Como nessa implementação há apenas um sensor, o valor desse será igual a 1, indicando o primeiro sensor, o de temperatura. Em seguida, há dois octetos especificando qual o tipo de comando desta mensagem. O byte mais significativo, com valor três, indica a classe de comandos a serem usados quando um transdutor está no modo de operação (transducer operating state commands) e o menos

significativo a função do comando. O valor um indica a função “read TransducerChannel data-set segment“, usada para ler um conjunto de dados do TransducerChannel. A terceira parte, também formada por dois octetos, indica o número de octetos dependentes da mensagem, ou seja, quantos octetos são esperados no campo de dados. O TEDS do sensor de temperatura descreve que os conjuntos de dados deste são formados por 10 campos, sendo cada um deles um número de ponto flutuante de 32 bits. Dessa forma, o campo de comprimento recebe o valor 40. Por último estão os 40 octetos contendo em seqüência de leituras, as informações obtidas pelo sensor de temperatura.

Table 1. Estrutura da mensagem iniciada pelo TIM para envio das leituras

| | | |
|-----------------------------|-----|------|
| Número do TransducerChannel | msb | 0x00 |
| | lsb | 0x01 |
| Comando | msb | 0x03 |
| | lsb | 0x01 |
| Comprimento | msb | 0x00 |
| | lsb | 0x28 |
| Dados | 1 | 0x?? |
| | ... | 0x?? |
| | 40 | 0x?? |

Resumidamente, uma vez que um TIM tenha se registrado ao NCAP, os seguintes passos são seguidos:

1. O NCAP envia um comando do tipo “operate” para colocar o TransducerChannel escolhido do TIM em operação;
2. O TransducerChannel entra em modo de operação, desconectando se do NCAP.
3. O TransducerChannel começa a fazer leituras no sensor em intervalos regulares até que o conjunto de dados esteja completo, colocando o TIM em modo de dormência entre esses intervalos;
4. O TIM se conecta novamente e o conjunto de dados é enviado ao NCAP, utilizando a estrutura de uma mensagem iniciada pelo TIM e os campos seguindo o formato de um comando de leitura do conjunto de dados, conforme a tabela 1;
5. O TIM se desconecta novamente, retomando as operações a partir do segundo passo.

3.2. Servidor de dados

As leituras dos sensores, recebidas pelo NCAP, são dirigidas a um servidor Web, para que sejam armazenadas e melhor visualizadas através da geração de gráficos. Através de requisições, utilizando o protocolo de comunicação HTTP, o servidor é informado da conexão ao NCAP de novos TIMs e dos dados coletados e enviados pelos mesmos. Esse servidor não tem relação com a utilização do protocolo HTTP apresentada na norma IEEE 1451.0, mas permite que os dados que tenham sido enviados para o NCAP sejam acessados através do protocolo HTTP em navegadores Web.

A figura 3 mostra uma página HTML gerada dinamicamente uma vez que cada um dos TIMs se conectou ao sistema, enquanto a figura 4 mostra o gráfico resultante das leituras de temperatura de um dos TIMs conectados ao NCAP.

TIMs

Sun SPOT[0014.4F01.0000.081C]

- [Onboard Temperature Sensor\[0014.4F01.0000.081C:1\]](#)

EPOS/Mica[1]

- [Onboard Temperature Sensor\[1:1\]](#)

Figure 3. Página exibindo TIMs conectados ao Servidor

Onboard Temperature Sensor [0014.4F01.0000.081C:1]

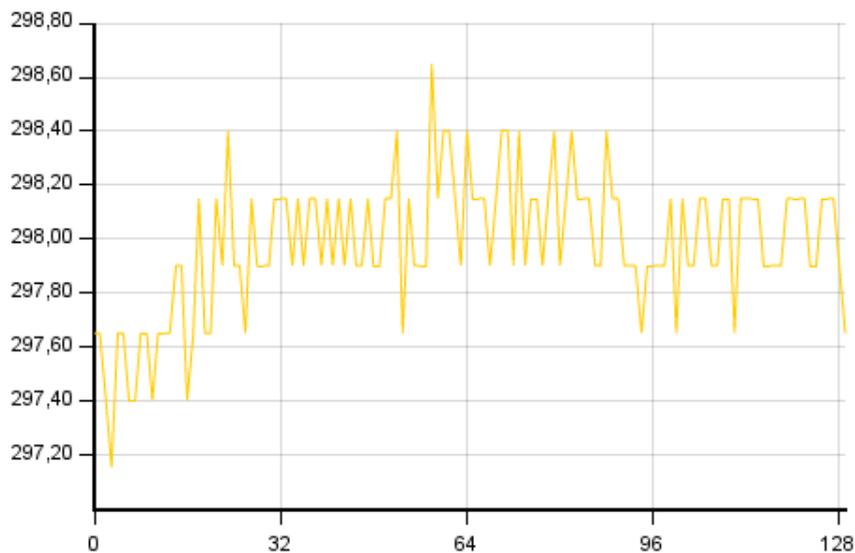


Figure 4. Gráfico gerado pelo servidor Web com as leituras de um sensor de temperatura

4. Conclusão

A padronização introduzida pela família IEEE 1451 traz diversas vantagens às RSSF. A existência de um conjunto padronizado de interfaces, estruturas e formatos de comunicação permitiu a integração de diferentes redes, formadas por diferentes nós e tecnologias de comunicação sem fio, possibilitando a configuração automática e compreensão das leituras dos nós conectados ao sistema. Além disso, tornou possível manter interoperabilidade entre os diferentes nós utilizados e que o sistema pudesse reconhecer e trabalhar com novos sensores de maneira ágil, uma vez que um novo nó pode ser colocado em funcionamento apenas ligando-o à rede, permitindo assim, uma expansão rápida e eficiente de redes existentes.

Por último, através da estrutura de “TIM Initiated Messages” e sua utilização introduzida neste trabalho, juntamente com um modo de operação condizente, permitiu-se que os TIMs pudessem manter-se constantemente em estado de dormência. Dessa forma, os TIMs conectados ao NCAP puderam trabalhar de forma mais autônoma, enviando as informações coletadas sem haver a necessidade de sucessivas interrogações pelo NCAP aos TIMs registrados no sistema.

References

- (2007). Ieee standard for a smart transducer interface for sensors and actuators - common functions, communication protocols, and transducer electronic data sheet (teds) formats. *IEEE Std 1451.0-2007*, pages 1–335. Revision of 1451.1-1999.
- Lee, K. (2000). Ieee 1451: A standard in support of smart transducer networking. In *Instrumentation and Measurement Technology Conference, 2000. IMTC 2000. Proceedings of the 17th IEEE*, volume 2, pages 525–528 vol.2.
- Lee, K. and Song, E. (2007). Wireless sensor network based on ieee 1451.0 and ieee 1451.5-802.11. *Electronic Measurement and Instruments, 2007. ICEMI'07. 8th International Conference on*, pages 4–7.
- Mainwaring, A., Culler, D., Polastre, J., Szewczyk, R., and Anderson, J. (2002). Wireless sensor networks for habitat monitoring. *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 88–97.
- Romer, K. and Mattern, F. (2004). The design space of wireless sensor networks. *Wireless Communications, IEEE [see also IEEE Personal Communications]*, 11(6):54–61.
- Song, E. Y. and Lee, K. (2008). Understanding ieee 1451-networked smart transducer interface standard-what is a smart transducer? *Instrumentation & Measurement Magazine, IEEE*, 11(2):11–17.
- Szewczyk, R., Mainwaring, A., Polastre, J., Anderson, J., and Culler, D. (2004). An analysis of a large scale habitat monitoring application. *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 214–226.
- Tilak, S., Abu-Ghazaleh, N. B., and Heinzelman, W. (2002). A taxonomy of wireless micro-sensor network models. *ACM SIGMOBILE Mobile Computing and Communications Review*, 6(2):28–36.
- Wanner, L. F. (2003). The epos system supporting wireless sensor networks applications. page 53.