

# Improvements in Wireless Communication and Support for Wireless Multimedia Sensor Networks in IEEE 1451

Leonardo Maccari Rufino, Rodrigo Vieira Steiner, Rodrigo Valceli Raimundo, Antônio Augusto Fröhlich  
Laboratory for Software and Hardware Integration  
Federal University of Santa Catarina  
P.O.Box 476, 88040900 - Florianópolis - SC - Brazil  
Email: {leonardo,rodrigo,rodrigovr,guto}@lisha.ufsc.br

**Abstract**—The IEEE 1451 smart transducer interface standard aims at establishing a common interface to promote interoperability among the myriad of transducers currently in the market. Nonetheless, the acceptance of the standard in the realm of Wireless Sensor Networks (WSN) seems to be constrained by apparently incompatible operational semantics: while ordinary transducers deployed in control networks usually operate at fixed periods, or even as slaves of master controllers, wireless sensors mostly avoid such regimens due to its inherent energy and bandwidth utilization inefficiency. In the first part of this paper, we propose a reconciling strategy based on Transducer Interface Module Initiated Message (TIM-IM) that enables Wireless Transducer Interface Modules (WTIM) to communicate efficiently with a Network Capable Application Processor (NCAP). The proposed solution has been validated through a prototype implementation with expressive efficiency gains, increasing the sensor's battery lifetime, compared with the polling method. The second part of this work is related to the new and emerging technology called Wireless Multimedia Sensor Network (WMSN), which is not addressed in IEEE 1451. Thus, we also propose an adjustment to the IEEE 1451 project in order to enable its use in WMSN.

**Keywords**—IEEE 1451, Wireless Sensor Network, Wireless Multimedia Sensor Network, TIM Initiated Message

## I. INTRODUCTION

Wireless sensor networks (WSN) are formed by devices called sensor nodes capable of monitoring some phenomena around them and communicate via radio frequency. Factors such as mobility among nodes, heterogeneity in communication, and diversity of sensors make WSN vulnerable to failure. Aiming to solve these problems, the IEEE 1451 family was created.

Past efforts to bring IEEE 1451 to the realm of WSN focused mainly on interoperability, without major concerns about efficiency and conformance to the traditional semantics of such networks [1] [2]. The reconciling strategy proposed in this paper is built around the fact that Network Capable Application Processor (NCAP) can potentially close the gap between Wireless Transducer Interface Modules (WTIM) and control applications by deploying the concept of information freshness. Instead of polling WTIMs for fresh information about sensed data, NCAPs can rely on the Transducer Interface Module Initiated Messages (TIM-IM) feature of IEEE 1451

and let WTIMs report new information whenever there are new data to be reported, as shown in Figure 1.

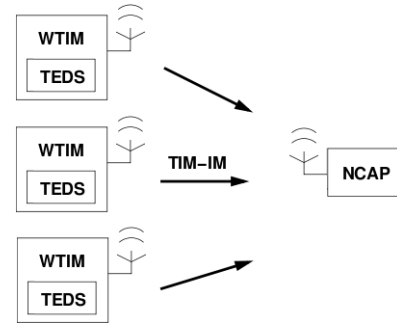


Figure 1. Overview of proposed IEEE 1451 WSN strategy.

In this context, WTIMs can decide whether an observed variation in sensed data is to be reported to NCAP via a TIM-IM, or simply ignored. In the worst-case, that is, in a constantly changing environment, this strategy spares NCAP request messages, thus improving on network bandwidth and also on mote's energy, since the sensor no longer needs to listen constantly the channel for NCAP request messages. In case of relatively stable environments, efficiency can be largely improved.

While WSNs are composed of sensors which monitor some scalar physical phenomena, such as temperature, pressure, and humidity, a new technology called Wireless Multimedia Sensor Network (WMSN) appeared during the last few years. It was leveraged due to the production of cheap CMOS (Complementary Metal Oxide Semiconductor) cameras and microphones, which can acquire rich media content from the environment. WMSN can be defined as networks of wirelessly interconnected sensor nodes equipped with multimedia devices that are capable of retrieving video and audio streams, images, and scalar sensor data [3].

Although WMSNs have the potential to enable many new applications, for instance, multimedia surveillance sensor networks, traffic avoidance and control systems, advanced health care delivery, automated parking advice, smart homes, environmental monitoring, and so on [4], IEEE 1451 standard does

not deal with this type of sensors. This paper proposes an adaptation to the IEEE 1451 project in order to enable its utilization in wireless multimedia sensor networks. To achieve this, we have included information in TEDS (Transducer Electronic Data Sheet), such as the description of data stream (audio or video). We have also added two new commands to the set of messages described by the standard, in order to notify the receiving side that a multimedia stream is starting/ending.

The rest of this paper is organized as follows: Section II presents the IEEE 1451 family. Section III shows the related work. The proposed TIM-IM is described in Section IV, and the adjustment in the standard to support WMSN is presented in Section V. Section VI shows the results. Finally, Section VII concludes the paper.

## II. IEEE 1451 FAMILY

The Instrumentation and Measurement Society's Sensor Technology Technical Committee TC-9 in the Institute of Electrical and Electronics Engineers (IEEE) has been working to establish a set of smart sensor interface standards called IEEE 1451. The goal of the standards family is to define a set of common communication interfaces for connecting transducers to microprocessor-based systems, instruments, and field networks in a network-independent environment [5] [6].

The specifications developed place no restrictions on the use of signal conditioning and processing schemes, analog-to-digital converters, microprocessors, network protocols, and network communication media. Therefore, the IEEE 1451 project's aim is to reduce the industry's effort to develop and migrate to networked smart transducers.

A smart transducer is the integration of an analog or digital sensor or actuator element, a processing unit, a communication interface, and the associated software from signal conditioning, calibration, diagnostics, and communication [2]. Based on this premise, a smart transducer model is shown in Figure 2 (a). Figure 2 (b) shows the IEEE 1451 smart transducer architecture, which is quite similar to the previous. The main differences are the division of the system into two major components, called TIM and NCAP, and the introduction of the TEDS.

### A. TIM and NCAP

To achieve transducers-to-network interchangeability and transducer-to-networks interoperability, the IEEE 1451 standards family divides the parts of a system into two general categories of devices, Transducer Interface Module (TIM) and Network Capable Application Processor (NCAP):

- **TIM** is a module that contains the communication interface, signal conditioning, analog-to-digital and/or digital-to-analog conversion and, in many cases, the transducer. A TIM may range in complexity from a single sensor or actuator to units containing many transducers (sensors and actuators).
- **NCAP** is the hardware and software that provides the gateway function between TIMs and user network or host processor.

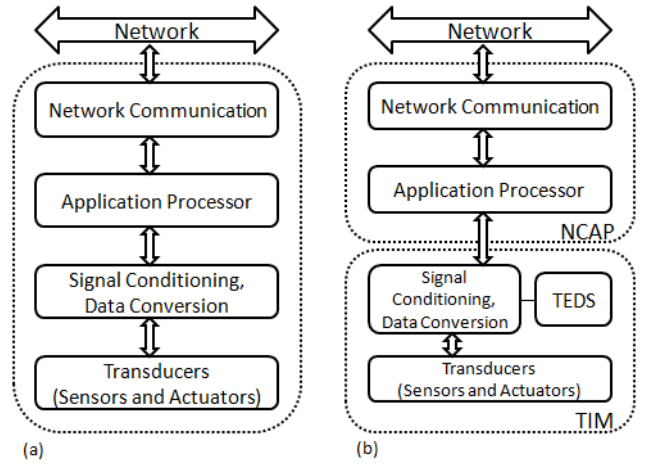


Figure 2. (a) A smart transducer model; (b) The IEEE 1451 smart transducer model.

Some members of the standards family provide the communication interfaces between an NCAP and one or more TIMs.

### B. TEDS

A key feature of an IEEE 1451 smart transducer is the specification of the standardized Transducer Electronic Data Sheet (TEDS) and their formats. TEDS contains manufacturer-related information about the sensor, such as manufacturer name, sensor types, serial number, measurement range, accuracy, and calibration data.

TEDS provides many benefits, for instance, it enables self-identification of sensors or actuators and provides long-term self-documentation. TEDS reduces human error and eases field installation, upgrade, and maintenance of sensors. And it also provides plug-and-play capability.

There are four mandatory TEDS required for TIMs, all others are optional. The mandatory TEDS are: *Meta-TEDS*, *TransducerChannel TEDS*, *User's Transducer Name TEDS*, and *PHY TEDS*. TEDS can be stored in some form of electrically erasable programmable ROM, if the contents never change, and the changeable portions can be in TIM's RAM [2].

### C. Family Members

Currently the family consists of six active standards and one in development. This work is related to two of them: IEEE 1451.0 and IEEE 1451.5:

- **IEEE 1451.0** introduces the concept of TIM and NCAP connected by a media specified by another member of the family. It also develops a set of common functionality for the whole family, which is independent of physical communication media and includes the basic functions required to control and manage smart transducers, common communication protocols, and media-independent TEDS formats [7].
- **IEEE 1451.5** introduces the concept of Wireless Transducer Interface Module (WTIM), connected wirelessly over an approved radio communication module to a NCAP. The IEEE 1451.5 approved radios (Dot5AR) are

IEEE 802.11, IEEE 802.15.4, IEEE Bluetooth, and IEEE ZigBee technologies. A WTIM is a TIM which contains a Dot5AR for wireless communication [8].

#### D. TransducerChannel

TransducerChannel is the name given by the standard to the combination of transducer and the components of signal conditioning and conversion associated with it, such as ADC (Analog-to-Digital Converter). Each TransducerChannel is associated with an identification number of 16 bits, which works as destination address of commands sent by NCAP, beyond distinguishing the TransducerChannels in TIM. TIM receives the identification number 0, and any command to this address is intended for the TIM itself and not for an individual TransducerChannel.

When NCAP wants to send a request to a specific TransducerChannel, it must send the message to the TIM's physical address, such as an IP (Internet Protocol) address, and the IEEE 1451 command must contain the TransducerChannel ID in a specific field.

#### E. Operation Modes

Each TransducerChannel can support different sampling and transmission modes, both described in IEEE 1451.0. The combination of them form the operating mode of a TransducerChannel which describes, in general, how is the operation of the sensor or actuator. The first, sampling mode, describes how the TransducerChannel should proceed to perform data retrieval, if it is a sensor, or how the incoming data should be used, when the transducer is an actuator. The sampling mode can also set the interval in which data should be retrieved/used. The second, transmission mode, determines how the information is transferred between NCAP and TIM. The mode or modes of operation that can operate a TransducerChannel are indicated in *TransducerChannel TEDS's* attributes in each transducer.

The standard specifies some operating modes for the operation of transducers. The choice of the sampling mode *continuous sampling* in combination with either the transmission mode *streaming when a buffer is full* or *streaming at a fixed interval* forms an operating mode known as *streaming operation*. A sensor operating in this mode must acquire data and transmit them to NCAP when the reading set is completed, without the need to receive additional commands from NCAP. Thus, the readings taken by a sensor can be sent autonomously and automatically.

### III. RELATED WORK

Song and Lee [9] [10] describe an implementation of the proposed IEEE 1451.0 and 1451.5 standards using the Java programming language. This system consists of two wireless nodes, NCAP and WTIM, which communicate with each other using the standard interfaces through 802.11 wireless communication modules. Three examples are discussed in these papers. The first example focuses on wireless node announcement and discovery using the publisher-subscriber

model. The second example focuses on request-response of sensor data using the client-server model. Finally, the third example focuses on request-response of sensor TEDS using the client-server model. In these works, when NCAP needs to collect data from WTIM, it must take the initiative by sending a request, causing the mote to send a response back to it. Our work differentiates from these because WTIM is able to start the transmission of information without a previous request sent by NCAP, saving the network bandwidth.

Gilsinn and Lee [11] discuss the IEEE 1451 interface for smart sensors, and possible solutions for creating a wireless interface for the standard. Most sensors require an energy source, such as an electric power supply or battery, to power the sensing element and associated electronics. Thus, the authors comment that low-power wireless sensors would be useful in locations where maintenance is performed regularly and batteries could be changed as part of that maintenance. In addition, another way to energize the sensors would be getting energy from their surrounding, thus a power conversion device would be designed to take energy from heat, light, sound, vibration, or any other source of energy available from the environment. Our TIM-IM method aims to minimize energy consumption, keeping the sensor in a sleep state most of its time.

Mangharam [12] describes a real-time voice stream-capability in wireless sensor networks. The paper presents FireFly, a time-synchronized multimedia sensor network platform using Nano-RK real-time sensor operating system [13]. In order to achieve high end-to-end throughput, bounded latency, and predictable lifetime, hardware-based time synchronization was employed. The author also proposed to use RT-Link [14], a TDMA-based (Time Division Multiple Access) link layer protocol for message exchange on well-defined time slots and pipelining along multiple hops.

Efficient use of bandwidth in WSNs is of interest due to the constraints imposed by their size and power. In applications where an audio signal must be transmitted from sensors to a base station through a wireless medium, it is interesting to stop transmission when speech is not present. As a pre-processor to such a system, a voice activity detector (VAD) is required to make a decision as to whether or not the current frame should be transmitted. Berisha [15] proposes a low-complexity VAD and a simple gender classifier for use in a distributed acoustic sensing system. A local decision is made by the sensor, based on the measurements, and then, the data can be transmitted to the base station. At the base station, a data fusion algorithm is implemented that combines the sensor's individual decisions located in the network to form a final one.

Data compression is also an important aspect in WMSN due to its limited amount of bandwidth available for communication between wireless sensors. Gosztolya [16] proposes to use Huffman coding, a well-known lossless compression method for any kind of data, pre-calculating the Huffman codes of each possible sample offline. Therefore, the encoder must only read the appropriate code from a pre-calculated table for each sample value. By using Huffman combined with other audio

compression methods, they obtained a significant reduction in the required bandwidth, such as 30% using  $\mu$ -law codec, with practically no information loss, and 50%, with a small information loss, using Truncate method.

#### IV. TRANSDUCER INTERFACE MODULE INITIATED MESSAGE

Due to problems in WSN like the amount of energy available, some applications choose to keep the sensor nodes asleep during most of its operation. Nodes can be awakened from time to time, in order to make the proper measurements, process data, and communicate with other nodes. Among the operation modes available on IEEE 1451.0, the one that best fits this scenario is called *streaming operation*, as discussed in Section II-E.

The IEEE 1451.0 standard also describes a structure called “TIM Initiated Message”. However, it only demonstrates its use to send the status from TIMs to NCAP, such as to indicate that some TEDS have changed or if a command was rejected. The standard does not describe how we should proceed to send data collected by the sensor to NCAP. We propose a way for this transfer to be made, enabling a TIM to take the initiative and send collected data by its TransducerChannels to the NCAP on which it is registered.

We adopt the “TIM Initiated Message” structure, filling the existing fields with the values as instructed by the “read TransducerChannel data-set segment” command, described by the standard. This way, the application must wait for samples obtained by the TransducerChannel, which is working on *streaming operation* mode.

The message’s structure is shown in Figure 3. The first two octets are formed by the TransducerChannel address which is sending the message. The next two octets specify the message’s command type, where the most significant byte indicates the command class to be used and the least significant represents the command function. This work proposes to fill these fields with the values 3 and 1, indicating a “transducer operating state” command and “read TransducerChannel data-set segment” function, respectively. The third part also consists of two bytes and indicates the number of octets transmitted in the message, in other words, how many octets are expected in the data field. Finally, there are octets containing the reading set obtained by the sensor.

Channel (2 bytes)	Command (2 bytes)	Length (2 bytes)	Data (X bytes)
----------------------	----------------------	---------------------	-------------------

Figure 3. TIM Initiated Message format.

The various TEDS contain information related to the TIM and they represent the best place to describe whether a sensor is working on polling or TIM-IM mode. An important TEDS is the *TransducerChannel TEDS*, which contains information such as the physical unit used to define the information being measured (such as meter, kilogram, second, kelvin, etc.), the number of data samples transmitted by the sensor, the

sampling and transmission modes supported by the TransducerChannel, and so on.

In order to allow the use of messages initiated by TIM correctly, we propose to extend the table 58 of IEEE 1451.0 [7], which describes the data transmission modes supported by a TransducerChannel. This information is contained in *TransducerChannel TEDS* and the three new entries to the table are shown in Table I. According to the amendments, a sensor can describe that its TransducerChannel works exclusively in TIM Initiated Message mode, allowing the sensor to save energy, keeping it in a sleep state most of its time. Then, the value 1 of the original table represents a TransducerChannel working on polling mode, values 2, 3, and 4, running on polling and TIM-IM modes, and the new values 5, 6, and 7, only TIM-IM mode.

Table I  
Data transmission attribute.

Value	Description
5	This TransducerChannel is capable of being operated in the Streaming when a buffer is full mode.
6	This TransducerChannel is capable of being operated in the Streaming at a fixed interval mode.
7	This TransducerChannel is capable of being operated in the Streaming when a buffer is full or Streaming at a fixed interval modes.

Figure 4 presents the sequence diagram of IEEE 1451 working in polling mode, and Figure 5 shows the diagram when using TIM Initiated Message method. Briefly, once a WTIM has registered to NCAP and the NCAP has ordered WTIM’s TEDS as needed, the following steps of TIM-IM are:

- 1) NCAP sends an “operate” command to put the TIM’s TransducerChannel in operation;
- 2) TransducerChannel comes into operation mode, disconnecting from NCAP, if possible;
- 3) TransducerChannel begins to make sensor readings at regular intervals until the data set is complete, putting the WTIM in sleep mode between these intervals;
- 4) WTIM connects again, waits an “operate” command, and the data set is sent to NCAP using the structure of a message initiated by TIM and the fields filled according to a reading command format as described in this paper;
- 5) WTIM disconnects again, resuming the operations from step 3.

As can be seen in Figure 5, some exchanges of messages between NCAP and TIM can be removed to optimize the operation, reducing energy and network bandwidth consumption. Then, we propose an optimized TIM-IM approach, shown in Figure 6. Since NCAP knows that TIM works in TIM-IM mode, NCAP will not send any data request. Thus, the messages connect, operate, and disconnect have been removed.

Although TIM will be asleep during the sensing time and will not be able to respond to messages from NCAP, this optimized approach can be used in situations where it is useful to save the network bandwidth. Still, NCAP and TIM must agree in advance to use this mode.

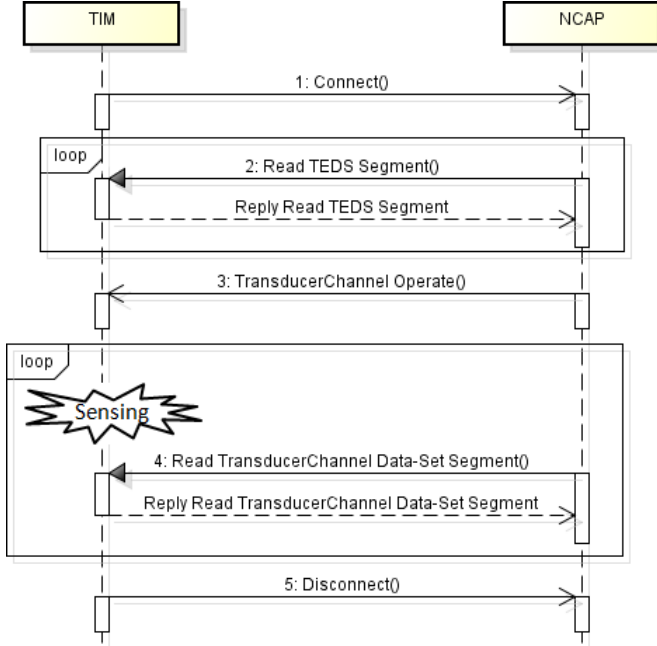


Figure 4. Polling sequence diagram.

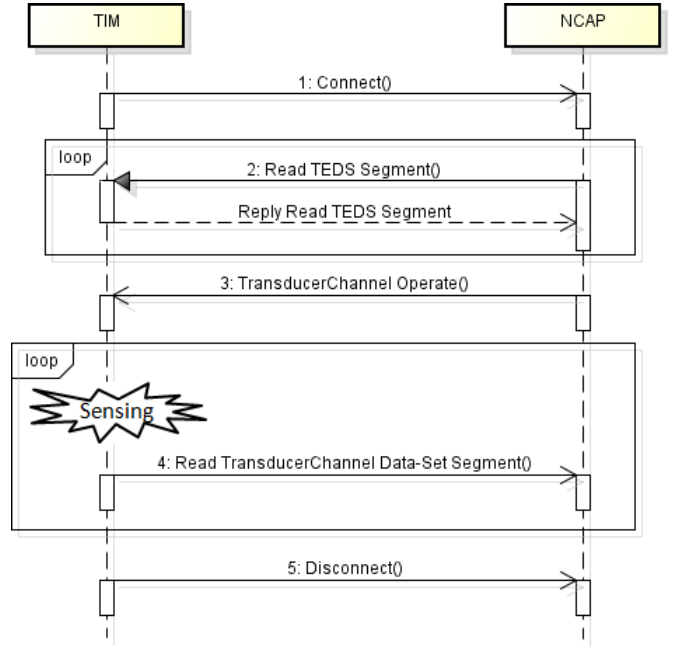


Figure 6. Optimized TIM Initiated Message sequence diagram.

## V. SUPPORT FOR WIRELESS MULTIMEDIA SENSOR NETWORKS IN IEEE 1451

For the sake of the increasing use of multimedia sensors and their potential use in several areas, this paper proposes the support for WMSN in IEEE 1451 standard. In order to allow the sending of multimedia information, first, we propose to extend the table 48 of IEEE 1451.0 [7], which describes the *TransducerChannel TEDS* data block structure. The new table will not change its information, except by adding two new entries, which are shown in Table II. According to the amendments, a sensor can describe that its *TransducerChannel* sends multimedia data (audio or video stream). These new options behave as the already existing non-multimedia alternatives (radians, steradians, meters, kilograms, seconds, amperes, kelvins, moles, and candelas), representing two new entries to the field that describes the physical units.

There are cases where it is not possible to completely express the physical unit with only one field. To solve this issue, an optional text-based TEDS, called *Units Extension TEDS*, is provided by the standard to give a place to include a text that would extend the unit field [7]. *TransducerChannel TEDS* has a field which provides the TEDS access code for the extension TEDS. We propose to use *Units Extension TEDS* to report some additional information about the audio or video stream sensed. For example, the text-based TEDS may indicate that the audio collected is sent using G.711 A-law codec.

In order to notify the opposite side that a data stream is starting or ending, we also propose to extend the table 31 of IEEE 1451.0 [7], which describes the commands used when the *TransducerChannel* is in operational state. Table III shows the two new transducer operating state commands, represented by the command lines with identification numbers 5 and 6.

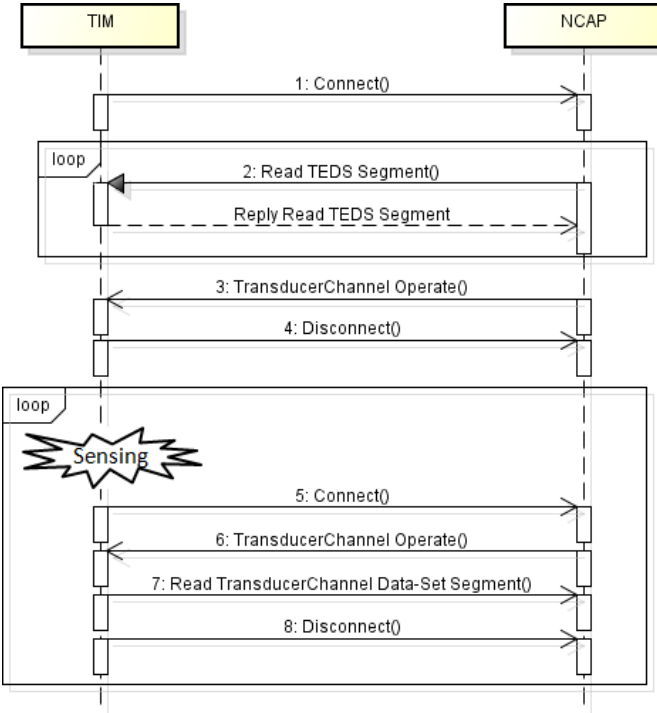


Figure 5. TIM Initiated Message sequence diagram.

Table II  
Structure of the TransducerChannel TEDS data block.

Field	Field Name	Description	Type	# octets
61	AudioStream	The exponent for Audio Stream	UInt8	1
62	VideoStream	The exponent for Video Stream	UInt8	1

They can be used in polling method, where NCAP requests sensed data to the sensors, and in TIM-IM mode, where the motes are responsible for taking the initiative to communicate with NCAP which they belong.

Some sequence diagrams are shown below, presenting the exchange of messages between NCAP and TIM. Figure 7 presents the traditional approach when using polling mode, where NCAP need to send a request message for each response from TIM. As multimedia sensors send a stream of data, with possibly a lot of packets, NCAP also has to send a huge number of requests. In order to optimize this situation, Figure 8 shows an alternative polling method, where NCAP must only send one command to start the stream (“Start Read TransducerChannel data-set segment”, Table III) and another message to stop it (“Stop Read TransducerChannel data-set segment”, Table III). In the meantime, TIM is responsible for sending the multimedia data.

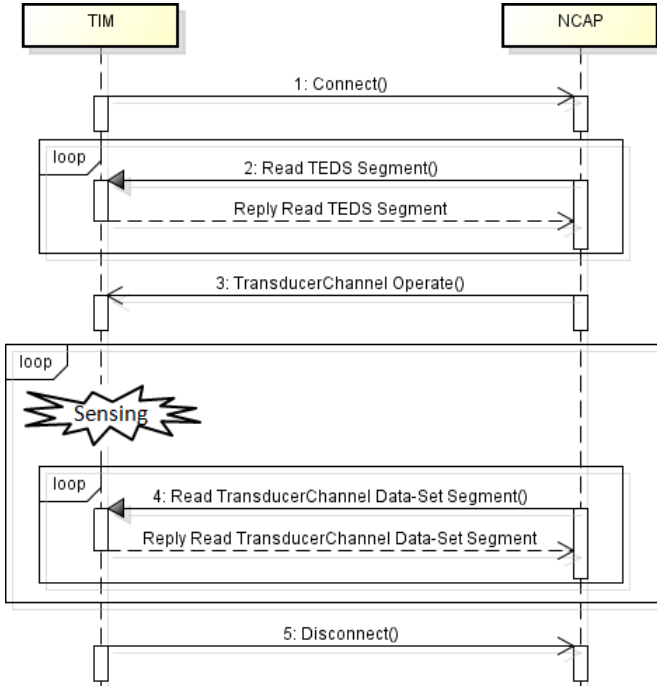


Figure 7. Polling sequence diagram for WMSN.

A sensor operating in TIM-IM mode must acquire data and transmit them to NCAP when desired, without the need to receive additional commands from NCAP. Figure 9 presents the sequence diagram when using TIM Initiated Message method. When the TransducerChannel is active and it needs to send multimedia information to NCAP, the sensor starts sending a command to prepare the other side (“Start Read Transduc-

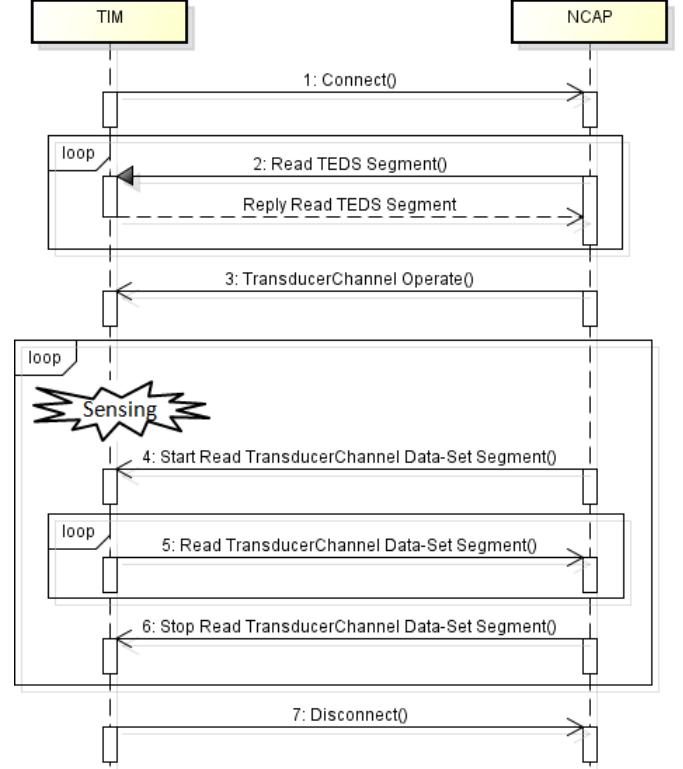


Figure 8. Optimized polling sequence diagram for WMSN.

erChannel data-set segment”), then it sends the sensed data, ending with a stop message (“Stop Read TransducerChannel data-set segment”). When working in TIM-IM mode, the start, read, and stop commands are sent using the “TIM Initiated Message” structure, as suggested in Section IV. Finally, the same optimization in TIM-IM presented in the previous Section can be performed here, resulting in Figure 10.

## VI. RESULTS

In order to evaluate our proposals we have implemented and tested the strategies in real sensor nodes, EPOSMoteII. The EPOSMoteII platform focus on modularization, and thus is composed by interchangeable modules for each function (processing/communication, sensing and power supply). Figure 11 shows the development kit which is slightly larger than a R\$1 coin. The main module is responsible for processing, storage, and communication. The model used in this research features a 32-bit ARM7 processor, 128kB of flash, 96kB of RAM, and an IEEE 802.15.4-compliant radio transceiver. The default sensing module contains a temperature sensor and an accelerometer, leds, switches, and a micro USB (that can also be used as power supply). In order to use a multimedia



Table III  
Transducer operating state commands.

cmdFunctionId	Command	Address class			Reply expected	Required/optional
		TransducerChannel	Proxy	Group/global		
5	Start Read TransducerChannel data-set segment	Yes	Yes	No	No	See NOTE
6	Stop Read TransducerChannel data-set segment	Yes	Yes	No	No	See NOTE

NOTE—A Start Read TransducerChannel data-set segment and a Stop Read TransducerChannel data-set segment commands are required for multimedia sensors.

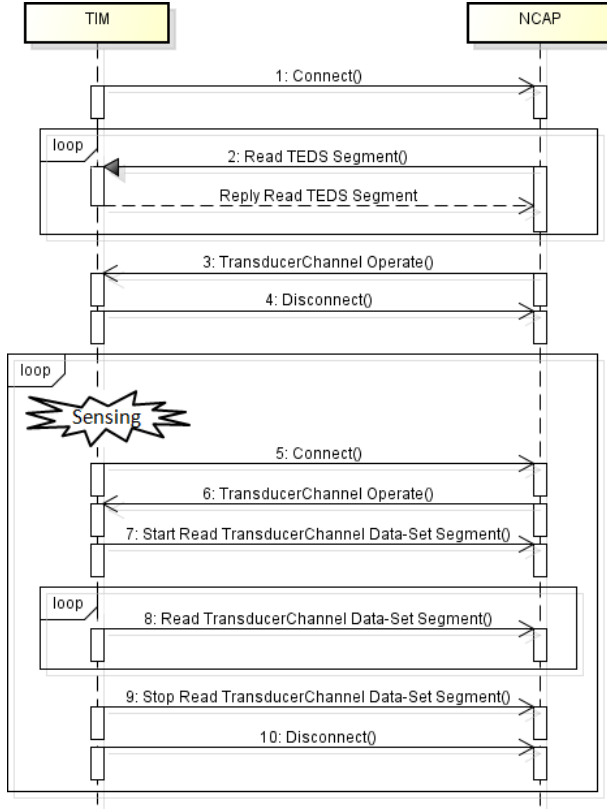


Figure 9. TIM Initiated Message sequence diagram for WMSN.

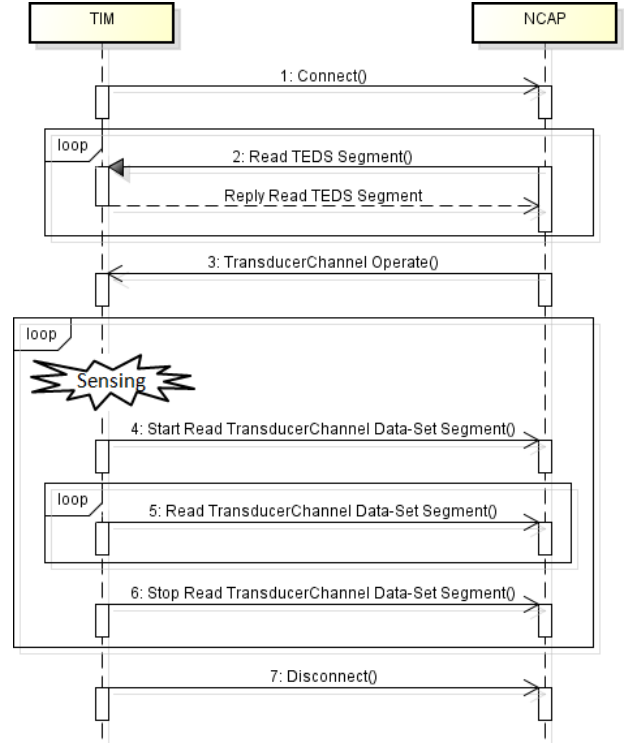


Figure 10. Optimized TIM Initiated Message sequence diagram for WMSN.

sensor, we have implemented an external audio board, which contains a microphone to capture surrounding audio.

We used EPOS (Embedded Parallel Operating System) [17] as our software platform, which is a component-based framework for generating dedicated runtime support environments, allowing programmers to develop platform-independent applications. The EPOS Project aims at automating the development of embedded systems, allowing developers to concentrate on what really matters: their applications [18].

EPOS has a small memory footprint, a complete communication stack, and support for sensor networks. The EPOS's runtime support environment for sensor network applications includes, besides the basic functions of an operating system running with limited resources, mechanisms for efficient power management, field reprogramming, uniform abstraction of heterogeneous sensor devices and configurable communica-

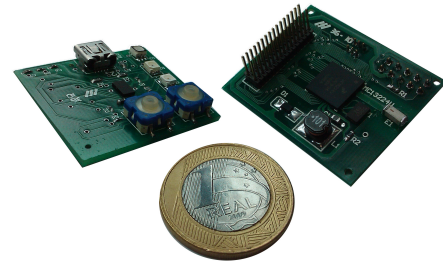


Figure 11. EPOS Mote II SDK side-by-side with a R\$1 coin. On the left the expansion module. On the right the processing module.

tion service through C-MAC (Configurable Media Access Control) protocol, thus fulfilling the specific requirements found in applications for WSN [19]. In addition, EPOS implements UDP/IP (User Datagram Protocol/Internet Protocol) and TCP/IP (Transmission Control Protocol/Internet Protocol), which are used in this work for communication between NCAP and TIM. UDP/IP is used to transmit audio stream and TCP/IP to ensure the reliability of the other messages. Finally, C-MAC can be configured to avoid idle listening through low power listening, where it periodically turns on the radio to check for channel activity, and if activity is detected, the radio is kept on to receive a packet, otherwise it is immediately turned off for some time [20]. For all these reasons, EPOS was chosen for use in this work.

The purpose of this evaluation is to determine the overhead of traditional operating mode (i.e. polling) versus TIM-IM, in accordance with the standard. We have evaluated the strategies in terms of energy and memory consumption. Figure 12 presents our test scenario.

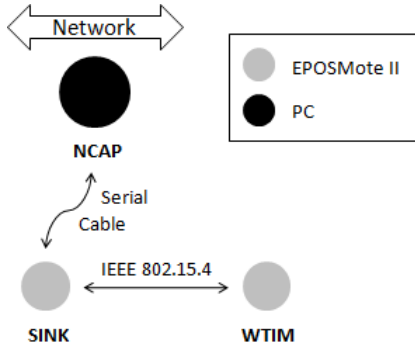


Figure 12. IEEE 1451 test scenario.

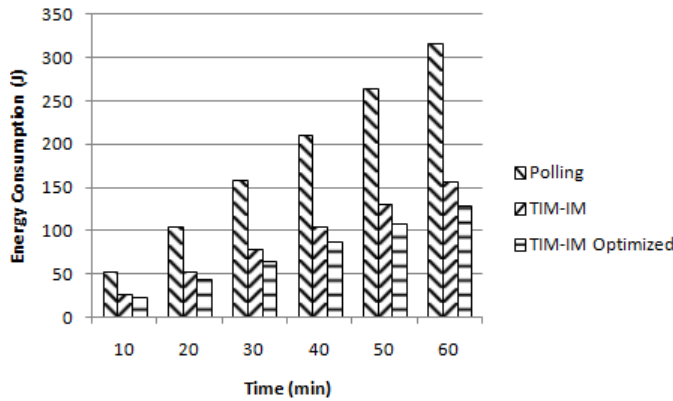


Figure 13. Energy consumption for temperature sensor.

In the first seconds, polling and TIM-IM have the same behavior, because of the time required for the sensor to self-identify and answer to the TEDS requests initiated by NCAP. After that, the polling method, which implements low power listening with a timeout of 500 ms and a sleep period of 1000 ms (EPOS' default values), continues to consume a high amount of energy because it keeps the radio on during part

of its time, waiting for messages. In both TIM-IM method (optimized and not), the sensor can turn off the radio while collecting data, turning it on from time to time to send the information. The optimized TIM-IM mode saves a little more energy because it does not send packets to connect and disconnect from TCP each time it will send the sensed data.

Figure 13 shows the energy consumption for a temperature sensor. In order not to favor any configuration, Transducer-Channel collects a temperature sample every 10 seconds in all cases and, in TIM-IM mode, it sends autonomously when its buffer with 10 entries is full, i.e., every 100 seconds. The polling method works similarly, but NCAP is responsible for sending a request every 100 seconds to TIM.

The results show that after 1 hour running, TIM-IM method consumed less than half the amount of sensor's energy (129,83 J in optimized TIM-IM, 156,13 J in TIM-IM, and 316,96 J in polling). This implies in a significant increase in the sensor battery lifetime. Using more sophisticated techniques, such as sending data only when a temperature sample has changed compared to its previous value, easily done in software, would result in an even higher gain in situations where there is little variation in the sensed data.

Figure 14 presents the energy consumption for a multimedia sensor. In this experiment the sensor collects an audio sample every 127 microseconds, resulting in 7884 samples per second. The node sends 27 packets per second, and each packet has 292 samples. We used G.726 codec at 16 kbit/s, which represents an Adaptive Differential Pulse-Code Modulation (ADPCM) technique [21]. Samples are sent for 5 seconds, following a pause of 5 seconds, forming a loop. The four operation modes were implemented as proposed in this paper. The consumption is higher here when compared to the temperature sensor due to the amount of data being handled. The audio sensor also performs a large number of ADC readings, and has to run the codec. The difference between polling and TIM-IM modes is smaller here because the mote sends packets for 5 seconds at intervals of 5 seconds, while the temperature sensor sends only one packet at intervals of 100 seconds. This way, the time the radio can stay off is smaller, which reduces the space for the optimization.

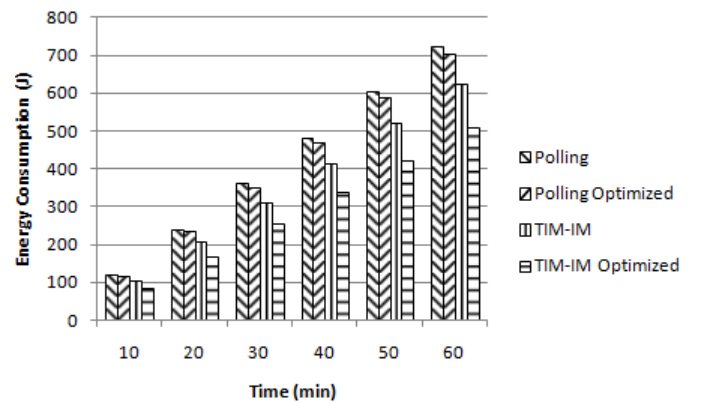


Figure 14. Energy consumption for multimedia sensor.



In order to analyze the memory footprint of our implementation, we have used the arm-size tool, from GNU Binutils version 2.20. The results are shown in Table IV. The first line, *Temperature WTIM + EPOS*, shows the size of the EPOS operating system along with the temperature WTIM's implementation of IEEE 1451. The second, *Temperature WTIM*, presents the size of temperature WTIM only, without EPOS. The next two lines show the results for audio WTIM, in a similar way.

Table IV  
Memory footprint (in bytes).

Section	.text	.data	.bss	TOTAL
Temperature WTIM + EPOS	56492	213	5280	61985
Temperature WTIM	8320	0	4	8324
Audio WTIM + EPOS	62812	213	5280	68305
Audio WTIM	9332	0	4	9336

## VII. CONCLUSION

This paper presented a solution for using IEEE 1451 standard in WSNs where sensor nodes start the communication through the use of messages initiated by TIM, while maintaining the ability to add different types of motes due to the existence of standardization.

As a result, we adopted the structure "TIM Initiated Message" described by the standard (IEEE 1451.0 [7]), filling the existing fields with the values as instructed by the command "read TransducerChannel data-set segment". Consequently, once placed the TransducerChannel into operation, the application must wait for samples obtained by the sensor and sent by WTIM to NCAP autonomously.

We have also presented an adaptation to IEEE 1451 aiming to allow its use in WMSN. First, we have included some information in the *TransducerChannel TEDS* data block structure, where a sensor can describe that its TransducerChannel sends multimedia data (audio or video stream). Second, the optional text-based *Units Extension TEDS* was used to report some information about the audio or video stream sensed, such as the codec used by the sensor. Finally, in order to optimize the exchange of messages between NCAP and TIM, we have extended the table with the commands used when the TransducerChannel is in operational state.

We validated the proposed solutions by building a WSN and WMSN IEEE 1451-compliant, using the EPOSMoteII platform. We have measured the energy and memory consumption, comparing polling and TIM-IM modes. The TIM-IM approach significantly increased the sensor battery's lifetime. The results show that after 1 hour running, the TIM-IM method consumed less than half the amount of sensor's energy compared with the polling method for the temperature sensor. The difference was smaller for the multimedia sensor, but still the TIM-IM mode presented better results.

## REFERENCES

[1] R. L. Oostdyk, C. T. Mata, and J. M. Perotti, "A Kennedy Space Center implementation of IEEE 1451 networked smart sensors and lessons learned," in *IEEE Aerospace Conference*, vol. 2, 2006, p. 20.

[2] E. Y. Song and K. Lee, "Understanding IEEE 1451-Networked Smart Transducer Interface Standard - What Is a Smart Transducer?" in *IEEE Instrumentation & Measurement Magazine*, vol. 11, no. 2, April 2008, pp. 11–17.

[3] M. AlNuaimi, F. Sallabi, and K. Shuaib, "A survey of wireless multimedia sensor networks challenges and solutions," in *Innovations in Information Technology (IIT), 2011 International Conference on*, April 2011, pp. 191–196.

[4] I. Akyildiz, T. Melodia, and K. Chowdury, "Wireless multimedia sensor networks: A survey," *Wireless Communications, IEEE*, vol. 14, no. 6, pp. 32–39, December 2007.

[5] K. Lee, "A Synopsis of the IEEE P1451- Standards for Smart Transducer Communication." [Online]. Available: [ieeep1451.nist.gov/1451synopsis-599F.pdf](http://ieeep1451.nist.gov/1451synopsis-599F.pdf)

[6] —, "IEEE 1451: A Standard in Support of Smart Transducer Networking," in *17th IEEE Instrumentation and Measurement Technology Conference*, vol. 2, Baltimore, MD, USA, May 2000, pp. 525–528.

[7] IEEE 1451.0, *IEEE Standard for a Smart Transducer Interface for Sensors and Actuators - Common Functions, Communication Protocols, and Transducer Electronic Data Sheet (TEDS) Formats*, September 2007.

[8] IEEE 1451.5, *IEEE Standard for a Smart Transducer Interface for Sensors and Actuators - Wireless Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats*, October 2007.

[9] E. Y. Song and K. Lee, "An implementation of the proposed IEEE 1451.0 and 1451.5 standards," in *Proceedings of the 2006 IEEE Sensors Applications Symposium*, 2006, pp. 72–77.

[10] K. Lee and E. Song, "Wireless Sensor Network Based on IEEE 1451.0 and IEEE 1451.5-802.11," in *8th International Conference on Electronic Measurement and Instruments*.

[11] J. D. Gilsinn and K. Lee, "Wireless interfaces for IEEE 1451 sensor networks," in *Proceedings of the First ISA/IEEE Conference Sensor for Industry*, 2001, pp. 45–50.

[12] R. Mangharam, A. Rowe, R. Rajkumar, and R. Suzuki, "Voice over sensor networks," in *Real-Time Systems Symposium, 2006. RTSS '06. 27th IEEE International*, December 2006, pp. 291–302.

[13] A. Eswaran, A. Rowe, and R. Rajkumar, "Nano-rk: an energy-aware resource-centric rtos for sensor networks," in *Real-Time Systems Symposium, 2005. RTSS 2005. 26th IEEE International*, December 2005, pp. 10 pp. –265.

[14] A. Rowe, R. Mangharam, and R. Rajkumar, "Rt-link: A time-synchronized link protocol for energy-constrained multi-hop wireless networks," in *Sensor and Ad Hoc Communications and Networks, 2006. SECON '06. 2006 3rd Annual IEEE Communications Society on*, vol. 2, September 2006, pp. 402–411.

[15] V. Berisha, H. Kwon, and A. Spanias, "Real-time acoustic monitoring using wireless sensor motes," in *Circuits and Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on*, 2006, pp. 847–850.

[16] G. Gosztolya, D. Paczolay, and L. Tó andth, "Low-complexity audio compression methods for wireless sensors," in *Intelligent Systems and Informatics (SISY), 2010 8th International Symposium on*, September 2010, pp. 77–81.

[17] A. A. Fröhlich, *Application-Oriented Operating Systems*. Sankt Augustin: GMD - Forschungszentrum Informationstechnik, 2001. [Online]. Available: <http://www.lisha.ufsc.br/guto/publications/aos.pdf>

[18] H. Marcondes, A. S. H. Junior, L. F. Wanner, R. L. Cancian, D. M. Santos, and A. A. M. Fröhlich, "EPOS: Um Sistema Operacional Portável para Sistemas Profundamente Embarcados," in *III Workshop de Sistemas Operacionais*, Campo Grande, MS, Brasil, Julho 2006, pp. 31–45.

[19] A. A. Fröhlich and L. F. Wanner, "Operating System Support for Wireless Sensor Networks," *Journal of Computer Science*, vol. 4, no. 4, pp. 272–281, 2008.

[20] R. Steiner, T. R. Mück, and A. A. Fröhlich, "A Configurable Medium Access Control Protocol for IEEE 802.15.4 Networks," in *International Congress on Ultra Modern Telecommunications and Control Systems*, Moscow, Russia, October 2010, pp. 1–8. [Online]. Available: <http://www.lisha.ufsc.br/pub/icumt2010.pdf>

[21] The International Telegraph And Telephone Consultative Committee, 40, 32, 24, 16 kbit/s Adaptive Differential Pulse Code Modulation (ADPCM), 1990.