

Research Article

Networked Electronic Equipments Using the IEEE 1451 Standard—VisioWay: A Case Study in the ITS Area

F. Barrero, S. L. Toral, M. Vargas, and J. Becerra

Escuela Técnica Superior de Ingeniería, Universidad de Sevilla, Avenida Camino de los Descubrimientos s/n, 41092 Sevilla, Spain

Correspondence should be addressed to F. Barrero, fbarrero@esi.us.es

Received 16 September 2011; Accepted 6 February 2012

Academic Editor: Tai Hoon Kim

Copyright © 2012 F. Barrero et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The concept of Intelligent Transportation Systems (ITSs) has been recently introduced to define modern embedded systems with enhanced digital connectivity, combining people, vehicles, and public infrastructure. The smart transducer concept, on the other hand, has been established by the IEEE 1451 standard to simplify the scalability of networked electronic equipments. The synergy of both concepts will establish a new paradigm in the near future of the ITS area. The purpose of this paper is to analyze the integration of electronic equipments into intelligent road-traffic management systems by using the smart transducer concept. An automated video processing sensor for road-traffic monitoring applications is integrated into an ITS network as a case study. The impact of the IEEE 1451 standard in the development and performance of ITS equipments is analyzed through its application to this video-based system, commercialized under the name VisioWay.

1. Introduction

Recent progress in semiconductor technology and microprocessor architectures, on the one hand, and the ubiquity of telecommunication networks, on the other, favour the growth of a novel market context for sensors and actuators [1, 2]. Sensors integrated into structures—coupled with the efficient delivery of sensed information—provide new services and benefits to society [3–5]. The actual implementation of such systems occurs on local area networks (LANs) that look like hierarchical networks with distributed computing capabilities. Using this technology, sensors and actuators provide a significant improvement in performance and effectiveness through the distribution of their resources.

It is interesting to link the development of such distributed data acquisition systems to the state of the art and future perspectives in the ITS field, which has attracted much attention in recent years [6]. ITS equipments are nowadays evolving into complex systems, allowing vehicle-to-vehicle (V2V) communication and cooperation [7, 8], or vehicle-to-infrastructure (V2I) interaction [9]. The public infrastructure (traffic lights, streetlamps or traffic signs) is adapting to these technological advances. Some examples are prompting or advisory message systems, informing the drivers on

particular traffic conditions—including alerts on imminent dangerous situations—or automated traffic-data acquisition systems transmitting relevant data to traffic control centers—traffic flow, per-lane vehicle counting, queue-length estimation, congestion level, estimation of journey times, red light runner detection, and so forth [7, 10, 11].

However, there are still some handicaps to this widespread use of electronic equipments in the ITS field: risk of breakage or connector failures in today wired systems, limited scalability in current networked systems bound to vendor-specific protocols, and so forth.

Wireless technologies are the best means to overcome these limitations, contributing for instance to a substantial ease of installation, to an increase of scalability [12] or to alleviate maintenance of physical links. Nowadays, an ideal wireless network should have some “smart” features, such as being highly scalable, software programmable, energy efficient, inexpensive, and hopefully plug-and-play compliant; on top of that, it has to be reliable and accurate, requiring a very low maintenance effort over the long term.

Recently, one of the most relevant milestones in this context is the introduction of the concept “smart transducer,” according to the IEEE 1451 standard [13, 14]. IEEE 1451 provides an open platform for the development of scalable

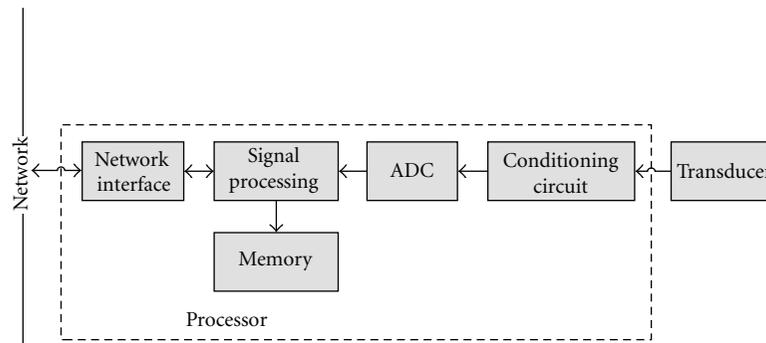


FIGURE 1: Smart transducer architecture.

networked electronic modules, using different physical links. The standard offers the ability to simplify the connectivity of electronic equipments, allowing the “plug and play” capability by means of the physical-link abstraction. These, among others, are some expected, interesting features in any IEEE1451-compliant equipment, leading to increased scalability and less expensive maintenance.

This paper analyzes the utility of IEEE 1451 for the development of ITS equipments. An IEEE 1451-compliant prototype for urban road-traffic monitoring is described and analyzed as a case study. The new prototype is built upon an existing intelligent video sensor, called VisioWay, which was developed for road-traffic monitoring applications (<http://www.visioway.com/>). VisioWay has been envisaged as part of the road-traffic infrastructure, for traffic monitoring and data collection. The relevant information is transmitted to the traffic control centers, to be used for traffic-control assessment or road-traffic management.

The paper is organized as follows. First, a brief overview of the IEEE 1451 standard and the smart transducer concept is given in Section 2. Next, Section 3 describes the preexisting VisioWay system and its use in road-traffic monitoring applications. Then, the integration of the IEEE 1451 standard into the VisioWay equipment is analyzed in Section 4, along with the description of the experimental setup. Finally, the conclusions are depicted in Section 5.

2. The Smart Transducer Concept: IEEE 1451 Standard Overview

The purpose of the IEEE 1451 standard is to define a set of common interfaces, functionalities, and commands for connecting sensors and actuators to microprocessor-based systems, instruments, and field networks in a network-independent fashion [13]. The main goal of this family of standards is developing network- and vendor-independent transducer interfaces, allowing electronic transducers to be replaced and/or moved with the minimum effort. Other desirable features include suppressing manual configuration steps. The Transducer Electronic Data Sheet (TEDS) concept is developed as an electronic datasheet that remains with the transducer during normal operation, allowing the real-time access to data and calibration details. Consequently, the IEEE

1451 family of standards provides an open platform for the development of networked electronic equipments.

Transducers that are 1451-compliant are also called smart transducers. Considering the purpose of the IEEE 1451 standard, a smart transducer should include certain elements like a physical transducer, a processor/memory core, and a local network interface through a microprocessor device. The transducer senses a given physical magnitude, generating the corresponding electrical signal. This signal is converted into digital values to be processed by the microcontroller, which delivers the resulting data to a local network. Only the selected or preconfigured data are accessible through the local network. Figure 1 depicts the architecture of a conventional smart transducer.

Two major components are defined, the Network Capable Application Processor (NCAP) and the Transducer Interface Module (TIM). The NCAP represents a network node that performs application processing and network communication functions, while the TIM consists of a transducer system, used for signal conditioning and data conversion, including up to 255 sensors and actuators. Figure 2 shows the overall structure for the standard IEEE 1451.

Eight separate standards are included in the IEEE 1451 family, from the IEEE 1451.0 to the IEEE 1451.5, the IEEE P1451.6, and the IEEE 1451.7.

The IEEE 1451.0 is one of the newest members of this family of standards. It defines a common set of commands, functionalities, and the format of the exchanged messages between the TIM and NCAP. The standard also provides the structure of the Transducer Electronic Data Sheets (TEDSs). The set of commands— independent of the physical communications media— includes the basic functions for reading and writing the transducers data, reading and writing TEDS, and sending configuration, control, and operation commands to the TIM. The aim of IEEE 1451.0 standard is to guarantee compatibility across the IEEE 1451 family.

The IEEE 1451.1 standard defines an information model for the communication between different NCAPs or between NCAPs and other systems [14]. It defines the NCAP as the main interface element between transducers and the network. The IEEE 1451 family of standards presents three possible protocols that can be indistinctly used to access electronic devices from the Internet: the Hyper Text Transfer Protocol (HTTP) outlined in the IEEE 1451.0 standard,

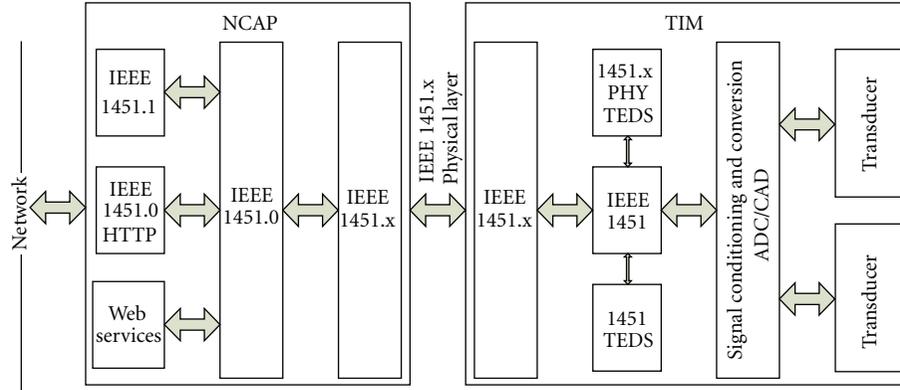


FIGURE 2: IEEE 1451 architecture overview.

TABLE 1: Generic format for TEDS.

Field	Description	Type	No. octets
0	TEDS length	UInt32	4
1 to N	Datablock	Variable	Variable
$N + 1$	Checksum	UInt16	2

an object and data model introduced in the IEEE 1451.1-1999 standard, and the proposed Smart Transducer Web Services or STWS derived from the Web Services Description Language or WSDL [15, 16]. The simplest web access is the HTTP protocol provided by the IEEE 1451.0 standard. The IEEE 1451.0 standard introduces an HTTP API that can be used to access the NCAP over TCP/IP networks. This HTTP protocol focuses on accessing transducer data and TEDS through the HTTP 1.1 protocol. Users send an HTTP request to a HTTP server running on the NCAP, receiving response from the server in Hyper Text Markup Language (HTML), eXtended Markup Language (XML), or Text format. HTTP and IEEE 1451.0 by themselves do not provide any means to discover the nodes in the network; nevertheless, there are several off-the-shelf network discovery tools that make use of different protocols that can be used for that matter.

Different physical interfaces are allowed between NCAP and TIMs, being one of the most interesting characteristics of the standard. For instance, the IEEE 1451.2 standard, first approved in 1997, specified a ten-wire point-to-point interface called Transducer Independent Interface or TII and the related protocol [17]. It defines a unique TIM named STIM or Smart Transducer Interface Module. However, a revision of this standard is expected due to the lack of manufacturer's interest and its incompatibility with the IEEE 1451.0 standard. This future revision will allow the support of more popular interfaces like USB and RS-232 [18].

The IEEE 1451.3 standard establishes a distributed multidrop interface network sharing a common pair of wires. This standard specifies multiple TIMs named TBIM or Transducer Bus Interface Modules, and it is intended to allow synchronized reading of large sensor arrays on a parallel transducer bus. Multidrop connectivity is used by defining channel identification protocols, hot-swap protocols, time

synchronization protocols, and the read and write logic functions used to access the TEDS and transducer data. The standard is now being revised to be adapted to the requirements of the IEEE 1451.0 standard.

The IEEE 1451.4 standard presents a mixed-mode interface (MMI) for analog transducers. TEDS to be used with this standard are also defined due to their specificity as they do not comply with TEDS defined in the IEEE 1451.0 standard [19]. The IEEE 1451.4 standard defines a mechanism for adding self-identification technology to traditional analog sensors and actuators.

The IEEE 1451.5 standard defines wireless communication methods between TIM and NCAP. Standards such as 802.11 (WiFi), 802.15.1 (Bluetooth), 802.15.4 (ZigBee), and 6LowPAN are adopted as viable wireless interface. Finally, other well-known interfaces, the CANopen interface and the Radio Frequency Identification (RFID) interface, meet, or will meet soon, the standard.

The proposed IEEE P1451.6 standard establishes a transducer-to-NCAP interface using the CANopen network. It also defines a mapping of the IEEE 1451 TEDS to the CANopen dictionary entries.

The recently approved IEEE 1451.7 standard describes communication methods and data formats, providing a TEDS for RFID sensors.

The TEDS is a key concept of the IEEE 1451 standard [20]. It standardizes transducers into TIMs, so any application can dynamically discover equipments, accessing their data and characteristics. TEDSs are classified into mandatory and optional TEDSs, following the IEEE 1451.0 specifications. The mandatory TEDSs include the Meta TEDSs, the Transducer Channel TEDSs, the PHY TEDSs, and the User Transducer Name TEDSs. In addition to these essential TEDSs, the IEEE 1451.0 standard supports other TEDSs: Calibration TEDS, Frequency Response TEDS, Transfer Function TEDS, Manufacturer-Defined TEDS, End User Application-specific TEDS, and Text-Based TEDS (including Meta ID TEDS, Transducer Channel ID TEDS, Calibration ID TEDS, Command TEDS, Location and Title TEDS, and Geolocation TEDS).

The IEEE 1451.0 standard specifies the format for these TEDSs, as it is shown in Table 1. This format is flexible and

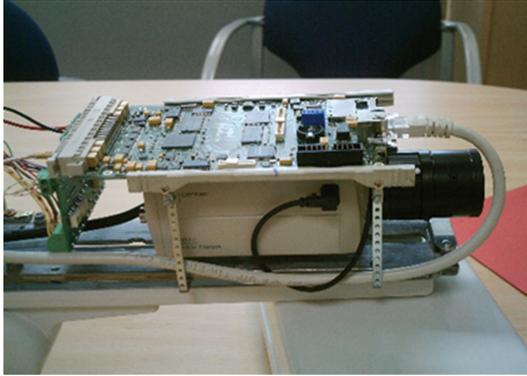


FIGURE 3: VisioWay prototype.

extensible to handle a wide variety of sensors, and it is compact to save memory space in the system. Each TEDS has a 4-octet length field, a data block with a variable size depending on the TEDS, and a 2-octet checksum field to verify the integrity of the TEDS. The data block includes different entries, and it uses a Type/Length/Value (TLV) structure to store each entry. The “Type” field is a 1-octet tag that identifies the data, the “Length” field specifies the number of octets of the data, the “Value” field includes the actual data.

The information in the data block depends on the TEDS, except in the case of the “TEDSID” field or TEDS identifier header. This field is standard in all TEDSs, including properties such as family (always 0), class (TEDS access code), version (IEEE 1451.0 standard version), and the number of octets in the length field of all TLV tuples in the TEDS except this one (normally 1).

The TEDS is normally stored in nonvolatile memories in the TIM, while the NCAP has a transferred copy. If the TEDS is not stored in the TIM, it is called virtual TEDS and the manufactures are expected to provide them. The standard allows the users to program some of these TEDSs once the smart sensor is installed, anyway. Hand assembly of TEDS is a very tedious task, even for the most simple TIMs. Hence, a TEDS compiler is highly recommended for this purpose.

Many smart sensor applications using IEEE 1451 are being developed [21, 22], including web-based sensor systems [23–26]. In this paper, the utility of the IEEE 1451 standard for the development of ubiquitous electronic equipment networks in road-traffic applications is analyzed. In particular, an existing video-based traffic sensor, VisioWay [27], is taken as a starting point to build an IEEE-1451 compliant smart traffic sensor, capable of providing new web services to the traffic control centers.

3. VisioWay in the Road-Traffic Management Field

VisioWay is a prototype of a novel electronic equipment for application in the ITS field using artificial vision [27]. Figure 3 shows the equipment. It has been designed for commercial purposes in traffic infrastructures, providing web services to the users of the infrastructure and preserving its



FIGURE 4: Detection areas in an installed prototype: details of the directional regions behavior (R3, R4, and R5) and queue detection areas (R1 and R2).

robustness as smart sensor in IP-based networks. VisioWay was designed as a road-traffic parameter estimation system, providing useful traffic information such as traffic flow, lane average speed and occupancy, or congestion levels. The data collection is structured according to a number of user-configurable polygons with arbitrary shape or size, called detection areas (Figure 4). Each one of these detection areas has interesting functionalities like presence and directional detection functionalities. The automatic vehicle detection method implemented in VisioWay is based on advanced background subtraction methods [28, 29].

VisioWay has the following features.

- (i) The Freescale i.MX21 multimedia processor is the core of the hardware system.
- (ii) Wire (Ethernet) and wireless (Bluetooth) connectivity is included.
- (iii) Interface to typical data storage media like USB or SD/MMC is provided.
- (iv) The hardware is driven by embedded Linux OS (kernel 2.4.20).
- (v) Artificial vision algorithms have been implemented for gathering information about relevant foreground objects (typically vehicles) in a traffic scene.

The i.MX21 processor is based on the advanced and power-efficient RISC processor ARM926EJ-S core, operating at speeds up to 266 MHz. On-chip modules are provided, including LCD and MMC/SD controllers, USB controllers, CMOS sensor interface, and an enhanced MultiMedia Accelerator (eMMA). The CMOS sensor interface provides the capability to acquire digital images, typically BT656 or raw data streams in the RGB or YUV components, delivering them to the media processor. The eMMA module consists of video processor units and encoder/decoder modules that support MPEG-4 and H.263 real-time encoding/decoding of images from 32×32 pixels up to CIF format at 30 fps. The processing unit is based on a preprocessor module that resizes input frames from memory or from the CMOS sensor

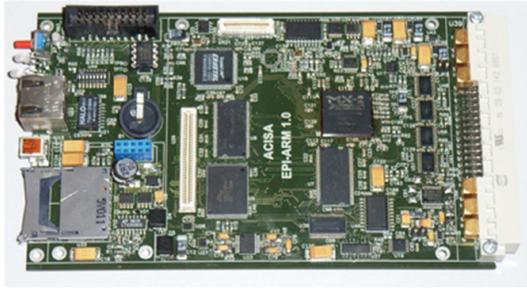


FIGURE 5: VisioWay embedded system.

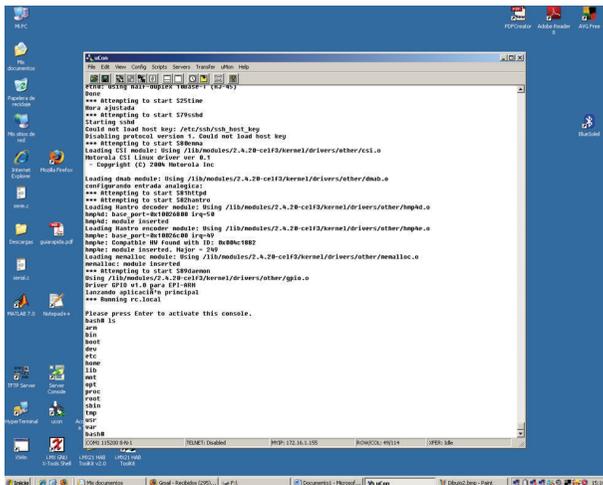


FIGURE 6: Kernel prompt of the prototype.

interface, performing color space conversion, and a postprocessor module that takes raw images from memory performing additional processing of a MPEG-4 video streaming to deblock, dering, resize, or color space conversion on decoded frames.

The prototype board (Figure 5) is also provided with a mini USB-OTG connector, a SD card connector, a RJ45 Ethernet connector, a Bluetooth expansion connector, and an analog video connector owed by a video decoder chip to allow analog video (PAL and NTSC formats) processing. On-board memory allows video processing and OS storage. A $32\text{ M} \times 16$ Flash memory is used to store the OS kernel image while two $16\text{ M} \times 16$ SDRAM modules are used for video processing and for decompressing the kernel image. An ARM Linux (kernel 2.4.20) operates the i.MX21 video processor, allowing access to a wide variety of open source software modules (Figure 6).

Several processes and services are in concurrent operation with the main process (road-traffic monitoring application). These processes are a watchdog process for rebooting the system when a periodic signal is not received from the main process, a MPEG server that operates as an interface to the i.MX21 MPEG hardware module for frame feeding from the application, and a HTTP server for configuration and supervision purposes, including an SSH server for remote

logging into the system, and an FTP server for upload and download operations.

4. VisioWay Smart Sensor and Experimental Results

The VisioWay system must be adapted to comply with the IEEE 1451 standard. The original HTTP and FTP services, adopted for reading data and configuring the VisioWay equipment, cannot provide by themselves a simple way to access all the system information, especially when plug-and-play capabilities are expected. This problem affects the development of applications that makes use of this data in two different ways. First, the application is dependent on the data structures and presentation forms and thus needs to be modified according to future changes. Second, it makes the development of new applications more difficult since a deeper knowledge of the device is required. As the devices are intended to estimate and store traffic measurements, it seems reasonable to consider them as multichannel sensors, interconnected into a sensor network. VisioWay was also intended to have different functionalities and to be installed in different locations over the urban area. Consequently, it is essential to have a common data structure and interface in order to guarantee full compatibility between different VisioWay equipments, making necessary a unique database with all the information that should be provided to upper-level systems and applications.

A network-compatible smart sensor node is implemented in VisioWay using the IEEE 1451 family of standards. The objective is to group VisioWay equipments into a single clustered network, some of them behaving as cluster heads—NCAPs in the IEEE 1451 notation. A common database (TEDS) must be designed, allowing connection regardless of the interface type. The intelligence of the device obviates unnecessary tasks when making the connection between systems, providing the interface information at connexion time, as a plug- and play-function. Other implementation details are as follows.

- (1) The system has been designed according to the IEEE 1451.0 standard specifications since this standard incorporates all the specifications except for IEEE 1451.4.
- (2) VisioWay can implement different functionalities, described in each Transducer Channel TEDS.
- (3) The role perspective: since VisioWay is intended to be installed across the city and connected through TCP/IP networks, each device can be considered as the NCAP of a particular sensor network using the HTTP API defined in the IEEE 1451.0 standard. At the same time, it is possible to connect other VisioWay devices into the network as TIMs, using the IEEE 1451.x standards and providing the NCAP a gateway function between TIMs and the Internet. The following alternatives are possible according to the developed implementation.

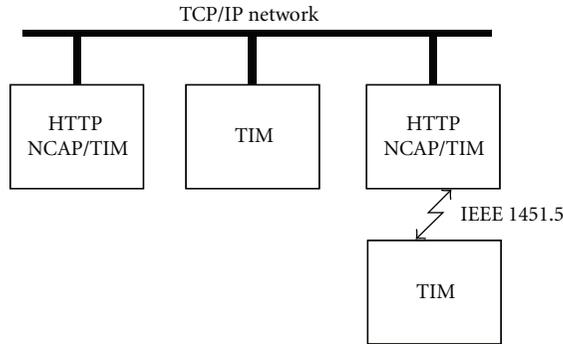


FIGURE 7: VisioWay equipments into the same TCP/IP network, playing different roles (NCAP and/or TIM roles).

- (a) A VisioWay system integrates a NCAP and a TIM into a single device. The Inter-Process Communication mechanisms available in the embedded Linux operating system link the NCAP and the TIM. As explained before, the VisioWay system is running under an ARM Linux kernel [27].
- (b) A VisioWay system serves as NCAP for other devices (TIMs using IEEE 1451.x interfaces). The current implementation supports the Bluetooth standard, as specified in the IEEE 1451.5 standard.
- (c) TCP/IP-based TIMs can be connected into the TCP/IP network. This approach has the limitation of being only standardized for the IEEE 802.11 interfaces via the IEEE 1451.5 standard. Nevertheless, this option makes sense since the communications are based on the IEEE 1451.0 message format, and the aim of the IEEE 1451.0 standard is to encourage compatibility across the IEEE 1451 family, making easier the addition of new interfaces, especially when TCP/IP is used.

According to the previous paragraphs, different VisioWay systems, playing different roles, can be connected into the same TCP/IP network, as it is shown in Figure 7. Upper-level applications need only to distinguish between NCAP and TIM access. Consequently, if a final user application is communicating with a TIM or an NCAP, IEEE 1451.0 or HTTP messages are, respectively, exchanged. Notice that every NCAP can be connected to multiple TIMs.

4.1. NCAP Implementation. An IEEE 1451.0 HTTP server has been developed in C language to implement NCAP functions over the VisioWay Linux kernel. Figure 8 depicts the block diagram of the designed NCAP application, while Table 2 shows all the implemented commands.

Although VisioWay, in its original form, runs a general purpose HTTP server, it is preferred to develop a new IEEE 1451.0 compliant HTTP server, taking into account the communication with multiple TIMs over different physical

interfaces and maintaining a minimum CPU and memory usage. The information transfer (HTTP and IEEE 1451.0) is managed using the Linux select system call, avoiding the need for multiple threads. Notice that more complex configurations are allowed running several HTTP servers within the same process. When the implemented HTTP server receives a request, it is analyzed to check if it is an implemented command. IEEE 1451.0 commands are dispatched to TIMs if required and TIMs are controlled to perform measurements or accessed from the client to read the results in the IEEE 1451.0 format. Notice that each registered TIM has a queue of requests in the NCAP implementation, and that each TIM request can be replied or not. If a reply is not expected, the NCAP uses the timing parameters included in the MetaTEDS of the TIM to determine whether the TIM is ready to receive new commands. In the case of being ready, the reply is retrieved from the queue of requests. On the contrary, if a reply is expected, the next command is sent only after its reception, taking into account timeout errors to consider failed operations that are reported to the HTTP client. It is also possible to prioritize commands, for example, reading a sensor against writing a TEDS, which requires more processing time.

4.2. TIM Design. The TIM implementation is based in the IEEE 1451.0 standard, allowing different physical interfaces. Eight standard command classes (0–7), some reserved classes (8–127), and open classes to manufacturers (128–255) are included, and the TIM is implemented as an independent process with the ability to manage and store the basic information and functionality of the sensor device using TEDS forms. After initialization, the TIM runs like a data source server. It checks the physical interface for different requests from the NCAP and decodes the command, which indicates the type of function requested and the channel address. The most common functions include reading of the channel transducer data, reading data types, or reading the channel status. If an error is detected on any of the Transducer Channels during the service, the TIM sets an appropriate flag in the channel status register.

All messages between TIM and NCAP can be classified into command messages, TIM initiated messages and reply messages, following the IEEE 1451.0 standard recommendations. All transmitted information is bundled together into a payload, which is then encoded into an octet array.

The IEEE 1451.0 and 1451.5 standards allow the implementation of a point-to-point (P2P) or general networked connection for the TIM. A P2P connection is chosen because each TIM does not normally talk with more than one NCAP, allowing for a simpler TIM implementation, as neither addressing nor discovery protocols are required.

4.3. TEDS Realization. A simple TEDS compiler and a TEDS reader have been developed to generate and test the TEDS files for the VisioWay equipment. The TEDS compiler application is a WEB-based interface that allows users to introduce the required parameter fields and generates the TEDS accordingly. The generated TEDS is displayed as

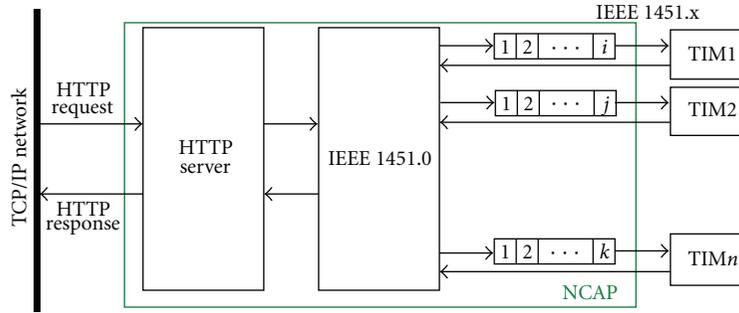


FIGURE 8: Block diagram of the implemented NCAP application (IEEE 1451.0-compatible HTTP server).

TABLE 2: Implemented commands.

Module	Operation	Task
Discovery API	TimDiscovery	List available TIMs
	TransducerDiscovery	List channels in one TIM
Transducer Access API	ReadData	Read access to measured data in a specific channel
	WriteData	Write access to a specific channel
TEDS Manager API	ReadTeds	Read a specific TEDS (Access Code)
	ReadRawTeds	Read TEDS from TIM bypassing NCAP cache
	WriteTeds	Write a specific TEDS
	WriteRawTeds	Write TEDS in TIM bypassing NCAP cache
	UpdateTedsCache	Update TEDS Cache
Transducer Manager API	Trigger	Perform a trigger on the specific channel

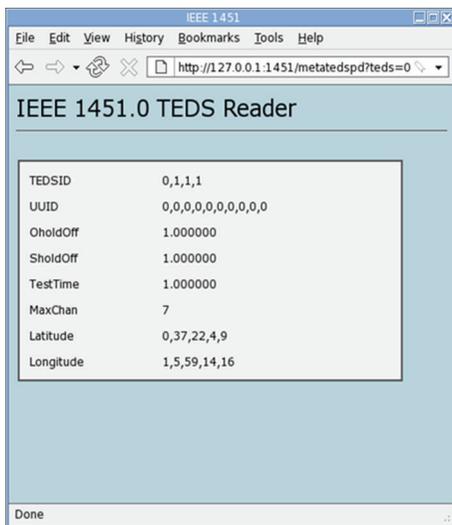


FIGURE 9: TEDS reader application.

a comma-separated list of bytes, which can be used to initialize memory buffers in the TIM program source code. The TEDS reader application is another WEB-based interface, as it is shown in Figure 9.

The implemented Meta TEDS describes the relations among the different Transducer Channels of the TIM and stores the timing parameters in the worst-case scenario. It

indicates the number of implemented Transducer Channels, being possible to group them into control, vector, geographic location, and transducer channel proxies groups. The timing parameters are used by the NCAP to establish time-out values in the communication software and determine a nonresponse event from the TIM. This TEDS also includes an “UUID” field, or globally unique identifier, which is an identification data associated with the TIM whose value is unique and it can include some manufacturing details.

Some fields are available for the manufacturers. Two of these fields have been used to save the geostationary position of the device: latitude (type 150) and longitude (type 151). Each field includes 5 octets, the first one being used to indicate the sign (i.e., north/south or east/west), while the rest specifies the degrees, minutes, seconds, and hundredths of second of the position. This format has been adopted for simplicity reasons.

Figure 10 shows the TEDS compiler application defining the Meta TEDS. The obtained file in HEX format is shown in Figure 11, including the total length in octets, the checksum, and the TLV tuples of the data block. At this stage, the Meta TEDS must be stored in the memory system of the TIM.

The *Transducer Channel TEDS* shows all the characteristics of the available sensors (each sensor has an associated Transducer Channel TEDS). VisioWay has been adapted in such a way that each functionality (vehicle-presence detection, directional vehicle detection, vehicle-queue detection, vehicle counting, occupancy, and traffic incident detection) is considered as an independent sensor, having their own

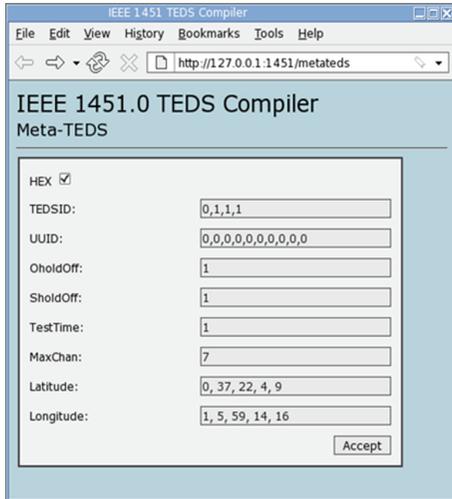


FIGURE 10: TEDS compiler application generating the Meta TEDS.

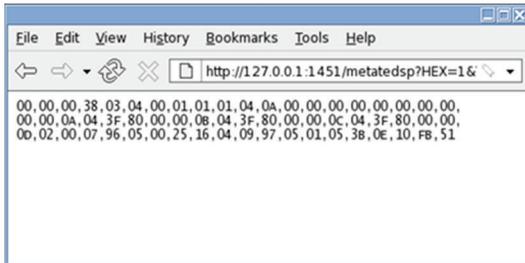


FIGURE 11: Meta TEDS "file" in HEX format.

Transducer Channel TEDS. The following information is stored in the data block of the TEDS.

- (i) The nature of the measured data (e.g., physical units and lower and upper range limits).
- (ii) The format of the data to be read from the TIM. This includes the data type (e.g., UInt8 or Float32) or the number of individual data samples in each data set, specified in fields like data model, data model length, model significant bits, and maximum data repetitions.
- (iii) The timing parameters describing the maximum required time to perform different procedures (e.g., transducer channel read setup time or self-test time), as well as the sensor sampling period.
- (iv) The available sampling modes of the sensor. Different sampling modes are supported in the standard (continuous, free-running, immediate, and trigger-initiated), as well as multiple buffers. Relevant types (i.e., fields) for these properties are sampling attribute, sampling mode capability, default sampling mode, buffered attribute, and data transmission attribute.

Figure 12 shows the TEDS compiler application creating the Transducer Channel TEDS. This TEDS is flexible enough

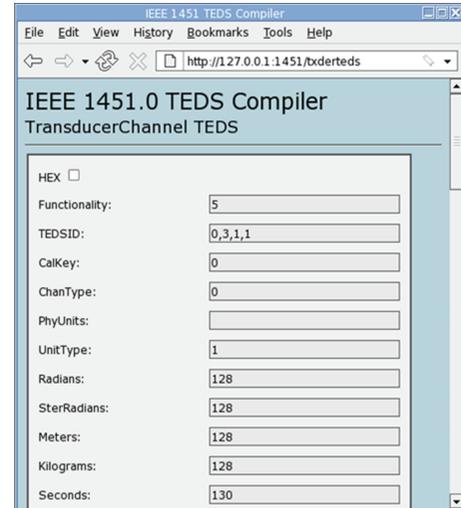


FIGURE 12: TEDS compiler application generating the Transducer Channel TEDS.

to accommodate to different transducers. The standard defines a small set of basic units that can be combined to provide most of the existing physical units. This set includes the radian and steradian and the seven international system base units. Any particular unit used in a smart sensor can be defined as a mathematical combination of these set of units. In particular, occupancy estimation, percentage of full queue, and other dimensionless quantities are obtained using VisioWay; thereby, the base units field is set to 0. Other measurements can be obtained from the ratio of two quantities, and their units can be specified using the Physical Units Interpretation field of this TEDS. Float32 data types have been chosen to be used in VisioWay channels.

Other characteristics and specifications defined with this TEDS for the smart sensor based on VisioWay are the following

- (i) Free running with pretrigger as sampling mode, although the immediate mode is also possible.
- (ii) One buffer is used.
- (iii) The system returns data when instructed. The smart sensor does not contain any streaming capability, and the data transmission attribute in this TEDS, which establishes the available transmission modes (streaming mode or request-triggered mode), is omitted. Nevertheless, streaming support could be included in future developments.
- (iv) The equipment boot time was used as the warm-up time. Notice that timing parameters are highly dependent on the hardware and optimizations of the VisioWay software. For example, fields like Transducer Channel update time, which indicates the maximum time between a trigger or a read command reception and the data available event in the data set, are determined by the video processing algorithms. Other parameters like Transducer Channel read delay time, which indicates the maximum delay between

TABLE 3: New types for the programmed Transducer Channel TEDS of VisioWay.

Functionality	Value
Vehicle-presence detection	1
Vehicle directional detection	2
Traffic incident detection	3
Vehicle counting	4
Occupancy	5

a read command reception and the corresponding start-transmission event, are also difficult to evaluate in the VisioWay equipment.

The types that the standard defines for this TEDS do not allow a complete description of VisioWay. The plug-and-play mechanism of VisioWay requires the use of one of the open-to-manufacturer types in the Transducer Channel TEDS (Table 3). Notice that more ITS-based services can be used, increasing the utility of the proposed system. For instance, if the state of a traffic light is available in real time to the VisioWay equipment, applications like red-light runner detection are feasible.

The *PHY TEDS* is dependent on the physical communication media that connects the TIM and the NCAP although the data are read-only octets after the TED is programmed. It includes detailed information of the IEEE 1451.X physical layer, establishing the minimum transmit latency, the maximum connected devices, and the maximum data throughput characteristics in the case of a wireless link. New types were defined to identify the IP and the physical addresses of the VisioWay wireless module. Notice that the information contained in this TEDS is not essential to access the TIM since most physical layers provide auto-configuration degrees completely independent of the IEEE 1451 standard.

The *User Transducer Name TEDS* stores the TIM name (the name assigned by an end-user to a specific TIM), establishing the name to identify a specific transducer in the complete system. The structure of this TEDS is recommended in the standard, and the manufacturer must provide a blank space of nonvolatile memory that can be programmed by the end-user using the standard TEDS access methods.

4.4. Traffic Monitoring Application Based on VisioWay Smart Sensor Equipments. The IEEE 1451 family of standards allows using VisioWay for providing real-time traffic data access to be used in web-based traffic control and information applications. Different prototypes are installed across the city of Seville and connected to a private Local Area Network (LAN) using its 10/100 Mbps Ethernet port and a RJ45 connector. Figure 13 shows two equipments monitoring a crossroads. The link speed and the device throughput on the private LAN have been analyzed.

A thorough study of the VisioWay capabilities and performance is required, prior to its conversion into a smart transducer. A multicast control mechanism must be applied to make an effective use of the available bandwidth [30]. An



FIGURE 13: VisioWay in two real emplacements in the city of Seville, Spain.

experiment has been performed to prove the robustness of the prototype in congested networks with multicast video delivery. One external PC runs the network traffic monitoring software, called monitoring program, based on libpcap libraries (Tcpdump/libpcap project). Libpcap utility is widely used for low-level packet monitoring and filtering. The main idea of this utility is to capture SOAP incoming and outgoing packets, while being completely independent from client coding. Using libpcap utility, each bit sent or received through the physical link can be captured, allowing the monitoring of the whole LAN traffic. Other set of PCs work as Data Servers. Two tests have been performed.

- (i) The congested network is emulated using the data servers, and the performance of VisioWay is studied. First, the throughput of the main algorithm—processed frames per second (FPS) when performing typical vehicle detection tasks—is evaluated under congested networks (up to 8000000 bps of network traffic) with a multicast control mechanism. The results show that the system throughput does not degrade in these conditions, remaining at steady values around 7 FPS. Notice that this ability depends on the hardware design. The Ethernet controller included in VisioWay analyzes network traffic at the network layer, avoiding any overload of the VisioWay CPU. The performance of VisioWay has also been tested under more adverse conditions, such as unicast control mechanism, with a 8000000 bps network traffic addressed to VisioWay. In this case, the generated network traffic degrades the system throughput till 5,57 FPS.
- (ii) Second, VisioWay performance is analyzed under constant real-time video delivery, using UDP protocol. The real-time video (MPEG-4 elementary video using CIF format, at 384 Kbps and 25 FPS) is delivered to the network monitoring PC. Table 4 and Figure 14 summarize the obtained results. The time between packets is acceptable for most MPEG-4 players in the 85% cases without necessity of incoming buffers. These buffers, usually included in

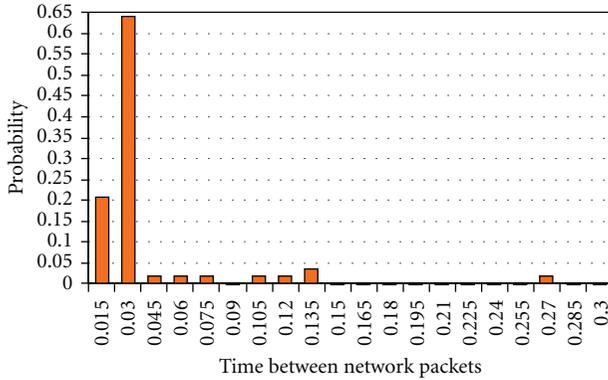


FIGURE 14: VisioWay without the smart sensor capability. Network traffic associated with video packets: time between packets probability density function.

TABLE 4: Performance of VisioWay without the smart sensor capability as a real-time video supplier.

Variables	Value
No. of packets	53
Average	0,030153 s
Standard deviation	0,043664 s
Maximum	0,260425 s
Minimum	0,000080 s

conventional MPEG-4 players, relax the random restrictions associated with the time between network packets.

As a conclusion, the device throughput on the Internet is robust. As expected, heavy network traffic is obtained when real-time video streaming is demanded. Figure 15 details the running processes on the VisioWay and their CPU time and memory requirements. The running processes correspond to the video processing algorithms (epi.out) and the SSH server for remote logging. In particular, the first one consumes the majority of the CPU time (98,8%), but just a small amount of memory (4,1%).

The developed IEEE 1451 HTTP interface allows an easy and straightforward way to acquire real-time traffic data, abstracting the implementation details and simplifying the development of web-based applications. A small web application has been developed to analyze the viability of introducing HTTP-NCAP services for monitoring road traffic in the city of Seville, Spain. The implemented application represents into Google Maps one of the specific traffic-data parameters obtained in VisioWay equipment related with traffic conditions: the region occupancy. This information is stored into designed Meta-TEDS. The application has been developed using Javascript, while the web access to the IEEE 1451.0 HTTP API is achieved using the XMLHttpRequest object. In this way, no server-side processing is required and the script supports the computational load of the response. Figure 16 details the VisioWay equipments used in the experiment, showing also the obtained occupancy values and

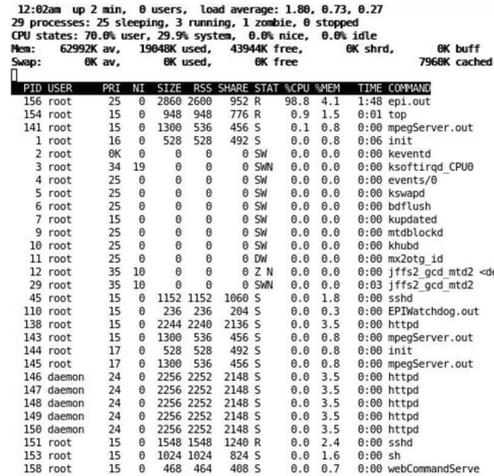


FIGURE 15: CPU time and memory requirements of concurrent processes executed in the VisioWay equipment prior its conversion into a smart transducer equipment.



FIGURE 16: Overview of the web application for identifying occupancy data around the city of Seville (Spain). Three VisioWay smart transducer equipments (TIM ID from 1 to 3) are shown on a Google Map of the analyzed area.

the Meta-TEDS of two of the equipments (TIM ID 3 and 1, resp.). Figure 17 shows the VisioWay equipments playing the “vehicle counting” task.

The device throughput on the Internet is more robust than before. The implementation of the NCAP and TIM processes corresponds to a 0% of the CPU time, being the CPU time requirements similar to the previous case (VisioWay equipment prior its conversion into a smart transducer) (Figure 15). Notice that the required amount of memory for the NCAP and TIM implementations is about 208 kbyte and 72 kbyte, respectively, including shared libraries (less than 3,17% and 1,10% of the total available memory, and about 25% and 15% of the video processing algorithms, resp.).

5. Conclusions

The development of the IEEE 1451 standard offers new opportunities for the ITS technology and embedded systems

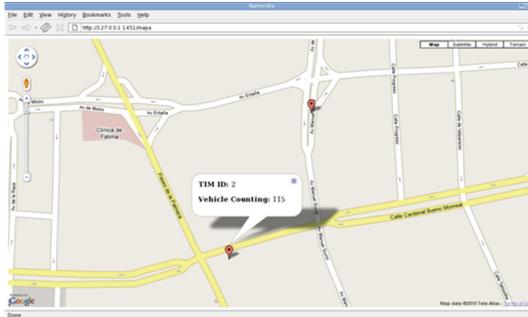


FIGURE 17: One VisioWay smart transducer implementing its vehicle-counting functionality, integrated into Google Map environment.

with application in road traffic management. The IEEE 1451 family of standards has been used to provide an open platform for the development of networked equipments, simplifying the connectivity to a network over the basis of HTTP web services and providing “plug-and-play” capabilities. In this paper, the combined use of the IEEE 1451 standard, the Internet, and embedded equipments has been analyzed in the context of web-based road-traffic monitoring applications. The IEEE 1451 standard has been integrated into an automatic video processing system, commercialized under the name of VisioWay. The proposed architecture has been tested on an experimental setup with the purpose of monitoring road-traffic data in the city of Seville, Spain, using the Internet as the communication link. The obtained results prove the effectiveness of the proposed system in the development of future ITS equipments.

Acknowledgments

VisioWay is a trademark of ACISA. This work has been supported by the Spanish Firm ACISA, the Spanish Ministry of Education and Science (Research Project with reference DPI2009-07955), and the Consejería de Innovación, Ciencia y Empresa (Research Project with reference P07-TIC-02621).

References

- [1] C. Y. Chong and S. P. Kumar, “Sensor networks: evolution, opportunities, and challenges,” *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1247–1256, 2003.
- [2] R. Aguilar-Ponce, A. Kumar, J. L. Tecpanecat-Xihuitl, and M. Bayoumi, “A network of sensor-based framework for automated visual surveillance,” *Journal of Network and Computer Applications*, vol. 30, no. 3, pp. 1244–1271, 2007.
- [3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “A survey on sensor networks,” *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002.
- [4] M. Can Filibeli, O. Ozkasap, and M. Reha Civanlar, “Embedded web server-based home appliance networks,” *Journal of Network and Computer Applications*, vol. 30, no. 2, pp. 499–514, 2007.
- [5] P. Kulkarni and Y. Ozturk, “MPHASiS: mobile patient healthcare and sensor information system,” *Journal of Network and Computer Applications*, vol. 34, no. 1, pp. 402–417, 2011.
- [6] S. L. Toral, M. R. M. Torres, F. J. Barrero, and M. R. Arahal, “Current paradigms in intelligent transportation systems,” *IET Intelligent Transport Systems*, vol. 4, no. 3, pp. 201–211, 2010.
- [7] F. Barrero, S. Toral, M. Vargas, F. Cortés, and J. M. Milla, “Internet in the development of future road-traffic control systems,” *Internet Research*, vol. 20, no. 2, pp. 154–168, 2010.
- [8] P. Alexander, D. Haley, and A. Grant, “Cooperative intelligent transport systems: 5.9-GHz field trials,” *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1213–1235, 2011.
- [9] L. Chen, C. Jiang, and J. Li, “VGITS: ITS based on intervehicle communication networks and grid technology,” *Journal of Network and Computer Applications*, vol. 31, no. 3, pp. 285–302, 2008.
- [10] M. Esteve, C. E. Palau, J. Martínez-Nohales, and B. Molina, “A video streaming application for urban traffic management,” *Journal of Network and Computer Applications*, vol. 30, no. 2, pp. 479–498, 2007.
- [11] M. Vargas, J. M. Milla, S. L. Toral, and F. Barrero, “An enhanced background estimation algorithm for vehicle detection in urban traffic scenes,” *IEEE Transactions on Vehicular Technology*, vol. 59, no. 8, pp. 3694–3709, 2010.
- [12] R. Verdone, D. Dardari, G. Mazzini, and A. Conti, *Wireless Sensor and Actuator Networks: Technologies, Analysis and Design*, Academic Press, 2007.
- [13] E. Y. Song and K. Lee, “Understanding IEEE 1451—networked smart transducer interface standard—what is a smart transducer?” *IEEE Instrumentation and Measurement Magazine*, vol. 11, no. 2, pp. 11–17, 2008.
- [14] V. Viegas, J. M. Dias Pereira, and P. M. B. Silva Girao, “A brief tutorial on the IEEE 1451.1 standard—part 13 in a series of tutorials in instrumentation and measurement,” *IEEE Instrumentation and Measurement Magazine*, vol. 11, no. 2, pp. 38–46, 2008.
- [15] E. Y. Song and K. B. Lee, “Smart transducer web services based on the IEEE 1451.0 standard,” in *Proceedings of the IEEE Instrumentation and Measurement Technology (IMTC '07)*, May 2007.
- [16] E. Y. Song and K. B. Lee, “STWS: a unified web service for IEEE 1451 smart transducers,” *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 8, pp. 1749–1756, 2008.
- [17] Y. Wang, M. Nishikawa, R. Maeda, M. Fukunaga, and K. Watanabe, “A smart thermal environment monitor based on IEEE 1451.2 standard for global networking,” *IEEE Transactions on Instrumentation and Measurement*, vol. 54, no. 3, pp. 1321–1326, 2005.
- [18] E. Y. Song and K. B. Lee, “Sensor network based on IEEE 1451.0 and IEEE p1451.2-RS232,” in *Proceedings of the IEEE International Instrumentation and Measurement Technology Conference (I2MTC '08)*, pp. 1728–1733, May 2008.
- [19] N. Olivieri, C. Distante, T. Luca, S. Rocchi, and P. Siciliano, “IEEE1451.4: a way to standardize gas sensor,” *Sensors and Actuators B*, vol. 114, no. 1, pp. 141–151, 2006.
- [20] S. Manda and D. Gurkan, “IEEE 1451.0 compatible TEDS creation using.NET Framework,” in *Proceedings of the IEEE Sensors Applications Symposium (SAS '09)*, pp. 281–286, February 2009.
- [21] L. Bissi, P. Placidi, A. Scorzoni, I. Elmi, and S. Zampolli, “Environmental monitoring system compliant with the IEEE 1451 standard and featuring a simplified transducer interface,” *Sensors and Actuators A*, vol. 137, no. 1, pp. 175–184, 2007.

- [22] D. Wobschall and S. Mupparaju, "Low-power wireless sensor with SNAP and IEEE 1451 protocol," in *Proceedings of the 3rd IEEE Sensors Applications Symposium (SAS '08)*, pp. 225–227, February 2008.
- [23] A. Flammini, P. Ferrari, E. Sisinni, D. Marioli, and A. Taroni, "Sensor integration in industrial environment: from fieldbus to web sensors," *Computer Standards and Interfaces*, vol. 25, no. 2, pp. 183–194, 2003.
- [24] E. F. Sadok and R. Liscano, "A web-services framework for 1451 sensor networks," in *Proceedings of the IEEE Instrumentation and Measurement Technology Conference (IMTC '05)*, pp. 554–559, May 2005.
- [25] G. Bucci, F. Ciancetta, E. Fiorucci, D. Gallo, and C. Landi, "A low cost embedded web services for measurements on power system," in *Proceedings of the IEEE International Conference on Virtual Environments, Human-Computer Interfaces, and Measurement Systems (VECIMS '05)*, pp. 7–12, June 2005.
- [26] K. B. Lee and E. Y. Song, "Sensor Alert Web Service for IEEE 1451-based sensor networks," in *Proceedings of the IEEE Instrumentation and Measurement Technology Conference (IMTC '09)*, pp. 1055–1058, May 2009.
- [27] S. L. Toral, M. Vargas, and F. Barrero, "Embedded multimedia processors for road-traffic parameter estimation," *Computer*, vol. 42, no. 12, pp. 61–68, 2009.
- [28] S. Toral, M. Vargas, F. Barrero, and M. G. Ortega, "Improved sigma-delta background estimation for vehicle detection," *Electronics Letters*, vol. 45, no. 1, pp. 32–34, 2009.
- [29] J. M. Milla, S. L. Toral, M. Vargas, and F. Barrero, "Computer vision techniques for background modelling in urban traffic monitoring," in *Urban Transport and Hybrid Vehicles*, S. Sciyo, Ed., 2010.
- [30] W. Q. Sun, J. S. Li, and P. L. Hong, "A stateful multicast access control mechanism for future metro-area-networks," *Internet Research*, vol. 13, no. 2, pp. 134–138, 2003.

