

Understanding the IEEE 1451 standard in 6LoWPAN Sensor Networks

Jorge Higuera, Jose Polo

Instrumentation, Sensors and Interfaces (ISI) Group. Castelldefels School of Technology (EPSC)
Departament d'Enginyeria Electrònica (EEL), Universitat Politècnica de Catalunya (UPC)
Esteve Terrades 7, Building C4, 08860 Castelldefels (Barcelona) Spain
jhiguera@eel.upc.edu, jose.polo@upc.edu

Abstract— The adoption of the IPv6 networking in IEEE 802.15.4 sensor networks, using IEEE 1451 standardization, increases the interoperability of low-power smart sensor devices in IP networks. This paper explains the design and implementation of IPv6 over Low power Wireless Personal Area Networks (6LoWPAN) and it uses the IEEE 1451 standard over IEEE 802.15.4 sensor networks. We propose an application of API IEEE 1451 that uses web services to transmit IEEE 1451 commands between Network Capable Application Processor (NCAP) and Wireless Transducer Interface Module (WTIM) using the client-server architecture. In addition the design of 6LoWPAN Physical Transducer Electronic Datasheet (PHY-TEDS) IEEE 1451.5 is explained. All TEDS were analyzed in terms of memory size. Each IEEE 1451 command is encapsulated in an UDP socket to establish NCAP-to-WTIM communication in an environmental monitoring application

Keywords: IEEE 1451, 6LoWPAN, NCAP, WTIM, TEDS, UDP.

I. INTRODUCTION

In last decade, the development of heterogeneous wireless technologies such as IEEE 802.11 (WiFi), Bluetooth, WiMAX or IEEE 802.15.4 has been very fast. In the past, some of these technologies were tested to deploy smart sensors in wireless networks using IEEE 1451 [1] but they are not fully end-to-end IP based architecture. For example in [2] is used Bluetooth or in [3] IEEE 802.15.4 interface and ZigBee, both use the IEEE 1451 standard. Currently, the development of IPv6 over Low power Wireless Personal Area Networks (6LoWPAN) [4] using on top the IEEE 1451.5 standard [5] allows the design of smart sensors fully end-to-end IPv6 based architecture using the physical layer IEEE 802.15.4 with low power options and limited computation. These features enable them to operate in scalable and efficient WSN, increasing the interoperability and the open standardization. A large number of these 6LoWPAN smart sensors work detecting events or physical phenomena and they could be deployed in smart ambient forwarding the information over air, using a multihop routing and collecting all information with a sink node based on native IP with LAN network access [6]. Examples of 6LoWPAN sensor networks are shown in [7]. Nowadays, IEEE 1451 is a family of high level standard organized around a set of common architectures and protocols to permit the interoperability of multiple smart sensors. The main objective of this standard is the adoption of manufacturer independent common communication protocol.

This standard can be used in wired or wireless sensor networks, for example, IEEE 1451.5 [9] to use in Bluetooth, IEEE 802.15.4, ZigBee and 6LoWPAN networks. The main advantage of IEEE 1451 is the concept of the Transducer Electronic Datasheet (TEDS) embedded in a memory of the sensor node, with options to self diagnosis, multisensing and to store this information in engineering units [10].

In this paper we explain the design of a 6LoWPAN sensor network architecture that uses the IEEE 1451 standard with two types of devices: a sink node, namely Network Capable Application Processor (NCAP) and a sensor node, namely Wireless Transducer Interface Module (WTIM) [11]. Our testbed uses the 6LoWPAN Berkeley stack Blip [12]. We define the mandatory Meta-TEDS (IEEE 1451.0), Transducer Channel TEDS (IEEE 1451.0), User's Transducer Name TEDS (IEEE 1451.0) and Physical 6LoWPAN TEDS (IEEE 1451.5). Finally we define the structure to send common commands IEEE 1451.0 using minimal UDP/6LoWPAN packets with header compression defined in RFC4944 [4] in a real 6LoWPAN sensor network deployment.

II. PROPOSED APPROACH

A. IEEE standard 802.15.4 into the IPv6 networks

The IPv6 stack can be used in IEEE 802.15.4 sensor networks using the 6LoWPAN adaptation layer. In this case, using compression and fragmentation of TCP or UDP datagram reduce the IP overhead in each message. The IPv6 stack uses the next layers shown in figure 1:

- Application layer includes HTTP protocol to communicate in the Web with other nodes.
- Transport layer that is responsible for transporting application layer messages, between the client and server sides of an application using TCP or UDP messages
- Network layer is responsible for routing datagram from one host to another, including fragmentation to support the IPv6. ICMPv6 performs error reporting as well as some other diagnostic functions in the network.
- 6LoWPAN adaptation layer includes header compression and fragmentation to reduce transmission overhead, over multiple hops.

- The Medium Access Control (MAC) layer manages access to the physical channel and network beaoning.
- Physical layer manages the physical RF transceiver and performs channel selection and energy management functions.

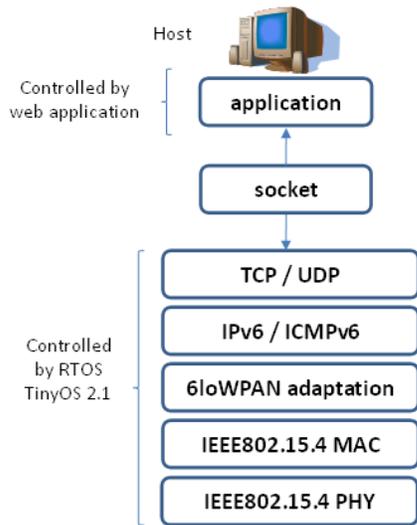


Figure 1. 6LoWPAN stack

The socket is a door between the web application and lower layers controlled by Real Time Operative System (RTOS) TinyOS 2.1 [13].

B. IEEE 1451 reference Model in 6LoWPAN sensor networks

The IEEE 1451 reference model is included in a 6LoWPAN sensor network. Figure 2 shows an example to connect a web client in an Internet Protocol Version 6 (IPv6) cloud or other web clients in an Internet Protocol version 4 (IPv4) cloud with our IEEE 1451 6LoWPAN sensor network that uses UDP packets. In this architecture, the IPv4 web client uses the commands IEEE 1451 in the payload data frame structure using a tunnel IPv6 over IPv4. This is done using a technique known as tunneling, which consists of encapsulating IPv6 packets within IPv4 ones, in effect, using IPv4 as a link layer for IPv6. Other users in an IPV6 network need a router to create the tunnel connectivity with the 6LoWPAN sensor network. Figure 2 shows the WTIM sensor nodes that use UDP packets to send the information. NCAP or coordinator node is a bridge of sensor network with LAN IPv4 network. In order to reach the IPv6 Internet, an isolated host or network must use the existing IPv4 infrastructure to carry IPv6 packets. The web client sends commands using an UDP socket to communicate with each NCAP and WTIM associate compliant common commands IEEE 1451.0. This approach uses the model client-server architecture. Today, this technique is very common to connect the web clients with 6LoWPAN sensor networks using IPv4 and could coexist for many years during the transition of IPv4 to IPv6, but it needs more complexity for both end-user, and the networks that provide the tunneling service. In our case we obtained the tunnel service by the Internet Service Provider (ISP) Hurricane Electric IPv6 Tunnel Broker [15] that permits manage and configure until four free IPv6 over IPv4 tunnels. For example Our NCAP uses the IPv6 direction

2001:470:1f12:89b::2/64 and the first WTIM uses the IPv6 direction 2001:470:1f12:89b:: 40

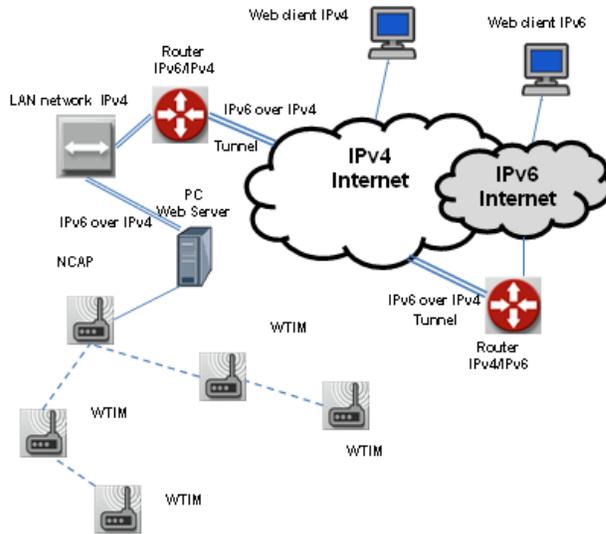


Figure 2. 6LoWPAN network compliant the IEEE 1451 standard

C. 6LoWPAN data frame structure using IEEE 1451 standard

The frame format compliant with the 6LoWPAN and IEEE 1451 commands in IPv6 packets on IEEE 802.15.4 networks are shown in figure 3. Since IPv6 requires support of packet sizes much larger than the IEEE 802.15.4 frame size, an adaptation to layer 6LoWPAN is required. The IEEE 802.15.4 frame includes the PHY and MAC IEEE 802.15.4, payload and MAC frame control sequence. In the IEEE 802.15.4 payload is included the 6LoWPAN UDP datagram with the header compression options: HC1 to compress IPv6 headers and HC2 to compress UDP headers. When more than one LoWPAN header is used in the same packet, there is an order established: Mesh addressing header (M), Broadcast header (B), fragment header (F), Header compression (HC1) and finally the 6LoWPAN payload.

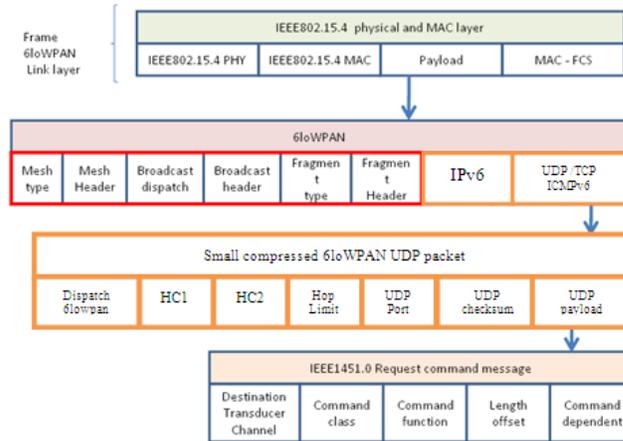


Figure 3. 6LoWPAN data frame structure IEEE 1451.0

IEEE 1451.0 commands, in our testbed, are encapsulated in 6LoWPAN payload. Initially, each full 6LoWPAN/UDP packet is composed of three main parts: IPv6 header (40 bytes), UDP header (8 bytes) and the payload (53 bytes). At present, the 6LoWPAN format allows a very heavy compression using HC1 header to compress IPv6 header, and HC2 to compress UDP header, reducing the full 6LoWPAN/UDP packet to the small compressed 6LoWPAN UDP packet. In practice, we use only the small UDP packet with header compression as it is shown in figure 3. Each small UDP packet uses only a dispatch value (1 byte) to indicate IPv6 over IEEE 802.15.4, the header compression dispatch (HC1) to compress IPv6 header (1 byte), HC2 header (1 byte) to compress the UDP source and destination port in a local 6LoWPAN network, IP Hop Limit is not compressed and it indicates the hop limit (1 byte), UDP checksum header (2 bytes) and finally the IEEE1451 command encapsulated in the UDP payload (up 108 available bytes in our WTIM).

III. RESULTS

A. Discover and registration phase

6LoWPAN sensor network compliant the IEEE 1451 standard uses a simple service discovery network protocol, shown in figure 4, to discover and register each NCAP and WTIM. In the first step, the Web client opens a socket with the Server. Next, the Server sends the web client request to NCAP. The NCAP discovers all WTIMs by sending an IEEE 1451 command to discover and register each transducer channel of WTIM. Finally, the WTIM returns the information stored in all TEDS and uses the response command to NCAP. This schema is shown in figure 4.

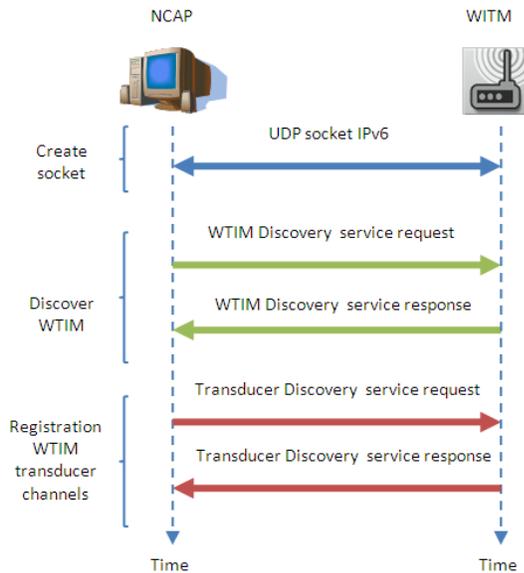


Figure 4. Discover sequence between NCAP and WTIM

The figure 5 shows the sequence for NCAP and WTIM to send the IEEE 1451 command encapsulated in UDP datagram

to create the socket to open the port 7 with destination address of WTIM 2001:470:1f12:9d1:41.

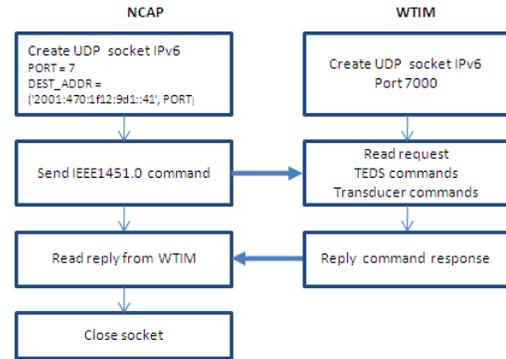


Figure 5. Command sequence between NCAP and WTIM

B. Transducer Electronic Data Sheets (TEDS)

Transducer Electronic Data Sheets (TEDS) is a submodule of WTIM node and it stores the information using non volatile memory. The mandatory TEDSs used in each WTIM node are: Meta-TEDS, TransducerChannel TEDS, User's Transducer Name TEDS, and Physical TEDS. Table 1 shows an example of Physical TEDS.

TABLE 1. 6LoWPAN TEDS PHY IEEE 1451.5

#	Field name	Description	type	# octets	Content
		TEDS length	UInt32	4	00 00 00 2C
3	TEDSID	TEDS Identification Header	UInt8	4	05 13 01 01
10	Radio	Radio Type	UInt8	1	03
11	MaxBPS	Max data throughput	UInt32	4	00 03 D0 90
12	MaxCDev	Max Connected Devices	UInt16	2	FF FF
13	MaxRDev	Max Registered Devices	UInt16	2	FF FF
14	Encrypt	Encryption	UInt16	2	00 00
15	Authent	Authentication	Boolean	1	False = 0 0
18	MaxSDU	Max SDU Size	UInt16	2	59
19	MinALat	Min Access Latency	UInt32	4	1ms (1024 uS)
20	MinTLat	Min Transmit Latency	UInt32	4	1ms (1024 uS)
21	MxXact	Max Simultaneous Transactions	UInt8	1	1
22	Battery	Device is battery powered	UInt8	1	1
23	RadioVer	Radio Version #	UInt16	2	01 00
24	MaxRetry	Maximun Retries Before discon	UInt16	2	00 10
96	Address	IPv6 address	UInt16	5	2001:470:1f12:89b::40
97	Header	Header datagram	UInt8	1	HC1 = 1
98	Data_transfer	Data transfer	UInt8	1	UDP = 1
99	Phy_Ch	Physical Channel IEEE802.15.4	UInt8	1	16
100	phyFrec	Frecuency	UInt16	1	2.480 GHz
101	Topology	6LoWPAN topology	UInt8	1	Mesh = 2
102	Fragmentation	Fragmentation	UInt8	1	4 bytes
103	App Scenary	Appplication scenary	UInt8	1	Agricultural Monitoring = 06
	Checksum		UInt16		AA AC

Physical TEDS is used in all WTIM nodes of our 6lowPAN network. This information is important because it helps the NCAP to know the configuration of each node, including its network address, communication protocol, operating frequency, physical channel, the use of fragmentation in UDP packets and other configuration options shown in table 1. TEDS that are used in this application:

- Meta-TEDS: stores the common information of all WTIMs.
- TransducerChannel TEDS: stores the information of each Transducer Channel of the WTIM.
- User's Transducer Name TEDS: stores the name by which the end user will know this WTIM.
- Physical TEDS: IEEE 1451.5 TEDS for 6lowPAN networks.
- Calibration TEDS: calibration parameters and time interval of each calibration. It includes constants to convert raw data into engineering units, for example, convert degrees Celsius in Kelvin Temperature.

We have added 8 new registers for 6lowPAN support (registers 96-103) not included in the IEEE 1451.5 standard. These registers provide useful information about the WTIM and could be included in a future revision of the IEEE 1451.5 standard. Table 2 shows the memory used in each mandatory TEDS to include in each WTIM sensor node in the 6lowPAN sensor network.

TABLE 2. TEDS MEMORY IN EACH WTIM NODE

TEDS	Memory used	Function
MetaTEDS (IEEE 1451.0)	40 bytes	Number of physical channels
Transducer Channel TEDS (IEEE 1451.0)	99 bytes	Physical Configuration parameters
User's Transducer Name TEDS (IEEE 1451.0)	25 bytes	Information name in ASCII
PHY TEDS (IEEE 1451.5)	98 bytes	Physical Configuration interface
6lowPAN		
Calibration TEDS (IEEE 1451.0)	60 bytes	Calibration parameters

C. TESTBED SPECIFICATION

Table 3 shows the parameters of the 6lowPAN sensor network.

TABLE 3. TEDS MEMORY IN EACH WTIM NODE

Parameter	Description
Number of NCAPs	1
Number of WTIMs	3
Radio Range outdoor	Up 200 m
Radio Range indoor	Up 100 m
Running Time	30 days
IEEE 1451.5 interface	6lowPAN
RTOS	TinyOS 2.1
Transport protocol	UDP
Phy standard	IEEE 802.15.4
Frequency band	2,4GHz
Channel	21

The NCAP uses the IEEE 1451 commands API to discover, register, configure and read the data from each onboard transducer channel after power-on and initialization of the WTIM node.

D. WIRESHARK CAPTURE IPV6 STATISTICS

To observe how the packets arrive to NCAP, Wireshark [14] can be used. Wireshark is a protocol analyzer used in data networks, software development, communication protocols and training in computer networking, available under the General Public License (GNU) to use in Windows, Linux, OS X, Solaris, FreeBSD, NetBSD and others operative systems. It is originally started in 1998 with the name Ethereal, but in 2006 was renamed Wireshark. This application is very useful to solve problems related with protocols and to develop data network analysis in real time. In our tests, the details of each data packet passing through an IPv6 network have been examined and analyzed. Wireshark has some limitations in Windows, supporting mainly Ethernet, point-to-point Protocol (PPP), TokenRing and WLAN networks but other networks could be supported in the future. Under Linux, additional network support includes: Bluetooth, capture on synchronous links using Cisco HDLC encapsulation (CiscoHDLC), capture FrameRelay traffic (FrameRelay), capture IrDA traffic (IrDA), USB, capture traffic from a machine to itself (LoopBack), and Virtual Bridged LAN (VLAN). Figure 6 shows information sorted and filtered that allows the user to see all traffic being passed over the IPv6 network.

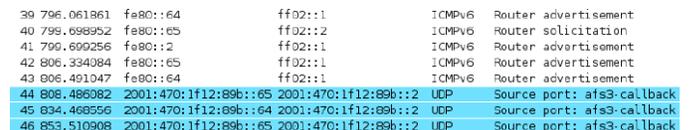


Figure 6. IPv6 network traffic

Note that in figure 6 the Internet Control Message Protocol, ICMPv6, is used by the 6lowPAN network to communicate network layer information. For example Router Advertisements are sent by each router announcing the IPv6 address. The NCAP discovers the address of their neighboring routers simply by listening for advertisements. The NCAP may send a Router Solicitation message on its local network. Figure 7 shows the 32 bytes payload of a UDP packet.

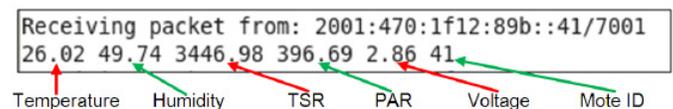


Figure 7. UDP report of the WTIM node

Each field of the UDP payload is read and collected in a web server and will be displayed through the web page.

ACKNOWLEDGMENT

This work is partially supported by the Spanish Ministry of Education and Science under contract TEC2007-66331 and by the European Regional Development Fund. Jorge Higuera is supported by the Programme Alban, the European Union Programme of High Level Scholarships for Latin America, scholarship No. E07D400433CO.

REFERENCES

- [1] IEEE 1451.0 Standard for a Smart Transducer Interface for Sensors and Actuators. Common Functions, Communication Protocols, and Transducer Electronic Data Sheet (TEDS) Formats. IEEE Instrumentation and Measurement Society. ISBN: 0-7381-5597-7. Sept 2007.
- [2] D. Sweetser, V. Sweetser, J. Nemeth-Johannes. "A Modular Approach to IEEE-1451.5 wireless sensor development". Proceedings of the IEEE Sensors Applications Symposium 2006, pp. 82-87, 2006.
- [3] J. Higuera, J. Polo, M. Gasulla. "A ZigBee wireless sensor network compliant with the IEEE 1451 standard". Sensors Applications Symposium, 2009. IEEE SAS 2009. pp. 309-313, 17-19 Feb. 2009
- [4] G. Montenegro, N. Kushalnagar, J. Hui, D. Culler. "Transmission of IPv6 Packets over IEEE 802.15.4 Networks". IETF RFC 4944, Sept. 2007; <http://tools.ietf.org/html/rfc4944>
- [5] IEEE Instrumentation and Measurement Society, TC-9, The Institute of Electrical and Electronics Engineers, Inc., "IEEE STD 1451.5-2007, Standard for a Smart Transducer Interface for Sensors and Actuators Wireless Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats". IEEE Instrumentation and Measurement Society. ISBN: 978-0-7381-5599-9.
- [6] J. Hui and D. Culler. "Extending IP to Low-Power, Wireless Personal Area Networks". IEEE Internet Computing, July-Aug. 2008. Vol 12, pp. 37 – 45.
- [7] H. Shimada, T. Sato, K. Fujikawa, H. Sunahara. "Implementation and Evaluation of a Memory Management System for IPv6 Wireless Sensor Networks". International Symposium on Applications and the Internet, 2008. pp. 401-404, 2008.
- [8] K. Lee, E. Song. "Understanding IEEE 1451-Networked smart transducer interface standard - What is a smart transducer?" Instrumentation & Measurement Magazine, IEEE. Vol 11, pp. 11-17, 2008.
- [9] NIST National Institute of Standards and Technology, IEEE 1451. <http://ieee1451.nist.gov/>
- [10] M. Dunbar. "Plug-and-play sensors in wireless networks". IEEE Instrumentation & Measurement Magazine, Vol 4, pp. 19 – 23, 2001.
- [11] D. Wobschall. "Networked sensor monitoring using the universal IEEE 1451 standard". IEEE Instrumentation and measurement magazine, pp.18 – 22, 2008
- [12] Berkeley IP project <http://smote.cs.berkeley.edu:8000/tracenv/wiki/blip>
- [13] TinyOS project homepage, <http://www.tinyos.net>.
- [14] Wireshark homepage, <http://www.wireshark.org>
- [15] Hurricane Electric Internet services homepage, <http://tunnelbroker.net/>