A Review on IEEE 1451 Family of Standards and IoT Stack Layers

Syed Khaja Ahmeduddin Zakir[1], Dr.Kaleem Fatima[2]

[1]*Assistant Professor in ECE Dept, MuffakhamJah College of Engineering & Technology, Hydereabad, India*

[2]*Professor in ECE Dept, MuffakhamJah College of Engineering & Technology, Hydereabad, India*

ABSTRACT

In this paper we explore the state of the art IEEE 1451 family of standards and the IoT stack layers. The IEEE 1451 is a family of standards for Interfacing Smart Transducers. These standards were created by Institute of Electrical and Electronic Engineers (IEEE) Instrumentation and Measurement Society's Technical Committee on Sensor Technology. The Internet of Things (IoT) is a global network of communicating Things, which is defined in a broad sense. The "Thing" has got sensing, actuating and computation capabilities. It can be uniquely addressed and can respond to stimuli from the environment.

***KEYWORDS**-IEEE 1451, IoT, STIM, TEDS, TII, NCAP*

I. INTRODUCTION

We find the usage of Transducers globally in almost all the sectors of life including Home, Medical, Transport, Energy and Industrial automation. A Transducer is a device which converts an electrical parameter to non-electrical parameter and vice versa. A Transducer can be a Sensor or Actuator. A Sensor is a Transducer which can generate an electrical signal in proportion to a physical, biological or chemical parameter where as an Actuator takes an electrical signal and responds by taking a physical action.A Smart Transducer is a collection of analog or digital transducer element, a processing unit, a communication controller and associated software for signal conditioning, calibration, diagnostics and communication[1].

The Internet of Things (IoT) was proposed a decade ago for connecting physical devices at the network level. The IoT will be an integrated part of the future internet.With IoT, we can establish a connection to anything, access it from anywhere and at any moment of time, and can efficiently access any type of service and information about any object. The purpose of IoT is to extend the use of internet with remote control ability, sharing of data, constant continuous connectivity and so on. All the devices can be connected to the local and global networks by using an ON and data collecting embedded sensor.

The following sections of the paper are organized as: Section II covers the IEEE 1451 architecture and the family of standards, Section III covers the IoT Stack Layers, protocols and applications.

II. IEEE 1451

A. Architecture

The IEEE 1451 standard based smart system is composed of Smart Transducer Interface Module (STIM), Transducer Independent Interface (TII) and Network capable application processor (NCAP). The conceptual diagram is represented in Figure 1.

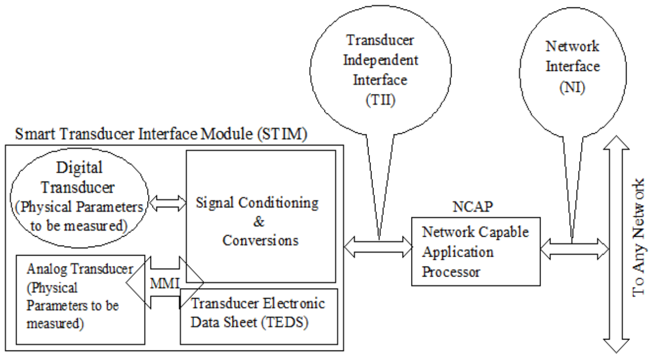

Figure 1. IEEE 1451 based conceptual Architectural diagram

The NCAP is a network node that can perform application processing and network communication functions. It is the smart sensor's window to the outside world. It consists of a processor with an embedded operating system. Any transducer or group of transducers can connect to any control network through a properly configured NCAP.Figure 2. Represents an example of NCAP being used through Ethernet.

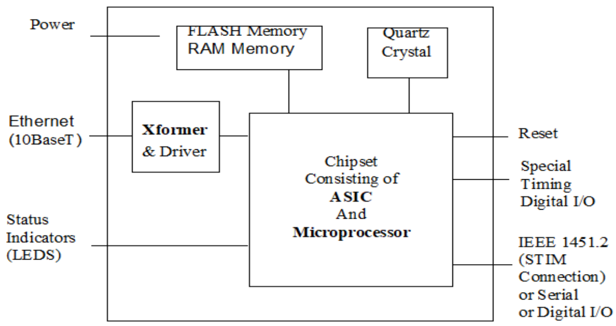

Figure 2. An example of NCAP [13]

The communication between NCAP and the sensor modules (STIMs) can be a serial communication like SPI, UART and I2C or wireless mediums like RFID, ZigBee, Bluetooth, etc.[2]

B. STIM

The STIM is the vital part of the measuring system.It is comprised of signal conditioning and data conversion units and a number of sensors and actuators with a combination of up to 255 devices [1]. It is responsible for providing the software as well as hardware to drive, collect, and manage the transducers associated with it. STIM also contains the TEDS. TEDS is meant for analog sensors for whom it stores the information like model number, serial number, sensitivity and calibration parameters. TEDS could be stored in EEPROM, if the contents never change or in the changeable portions of the RAM of the STIM.Figure 3 represents an example of STIM.

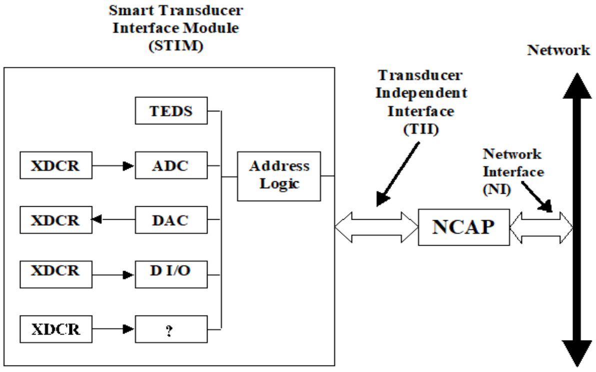

Figure 3. Overview of a STIM and how it associates through the TII to an NCAP [13]

The Transducer Independent Interface (TII) states a medium for communication and a protocol for transmitting sensor data. This interface provides a set of operations, such as read, write, read and write message, read and write responses, etc.

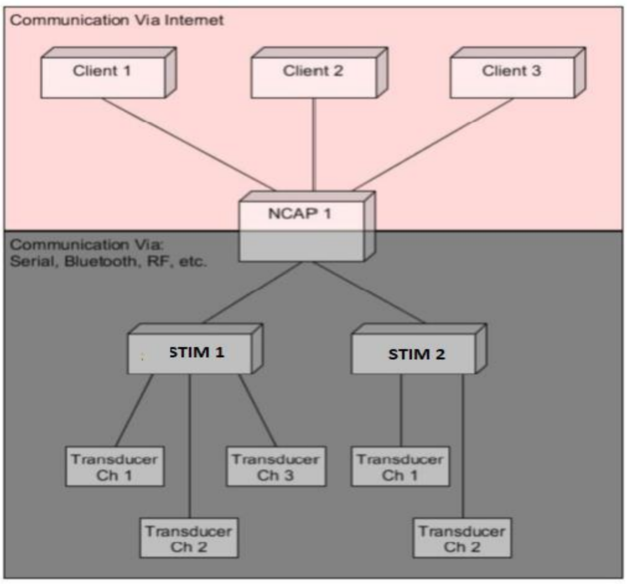

Figure 4. Basic Architecture of a P21451 Network with multiple Clients talking to a single NCAP [2]

The Network Interface (NI) describes a network communication protocol for communication of NCAPs to the network.Here comes the IoT into the picture and plays a very vital role.

Josh Velez et al [2] represented in their paper that each STIM can establish connection with only one NCAP, but the NCAP can be connected to as many Clients (End-Users) and STIMS as needed [2].The same is represented in Figure 4.

C. IEEE 1451 Family of Standards [1], [12]

A set of network-independent, common, open communication interface are defined by the IEEE 1451 for connecting transducers to microprocessors and control/field networks[1]. A set of protocols are given by the IEEE 1451 standard family for wireless and wired control and distributed monitoring applications.

IEEE 1451.0 –This standard lays out a set of common functionality, commands and TEDS for the family of IEEE 1451 smart transducer interface standards. It includes the basic functions of reading and writing to the transducers and to the TEDS and also includes the functions for sending configuration, control and operation commands to the STIM [1].

IEEE 1451.1 –This standard states not only a common object model but also the interface specification for the networked smart transducer. Its main focus is on the communication between NCAPs and other nodes in the system [1].

IEEE 1451.2 –This standard lays out a Transducers-to-NCAP interface and TEDS for point-to-point configurations. This describes a communication layer based on the SPI with additional hardware lines for flow control and timing, which results in a total of 10 lines for the interface [1].

IEEE 1451.3–This standard lays out a Transducers-to-NCAP interface and TEDS which uses a multi-drop communication protocol. It allows the connection of the transducers in the form of an array of nodes on a multi-drop transducer network through a common pair of wires [1].

IEEE 1451.4 – A Mixed-Mode Interface (MMI) is layed out by this standard for analog transducers operating in analog and digital modes. It main purpose is to add the TEDS feature to the legacy analog sensors [1].

IEEE 1451.5–This standard lays out a Transducers-to-NCAP interface and TEDS for wireless transducers.Standards like 802.11 (Wi-Fi), 802.15.1 (Bluetooth), 805.15.4 (ZigBee) and 6LowPAN can be used as the IEEE 1451.5 wireless interface[1],[2], [11].

IEEE 1451.6–This standard lays out a Transducers-to-NCAP interface and TEDS using the high-speed CAN open network interface. It correlate the IEEE 1451 TEDS to the CAN open directory[1].

IEEE 1451.7 – This standard lays out an interface and communication protocol between transducers and RFID systems. By providing sensor information in supply-chain reporting i.e identifying products and tracking of their condition, the standard opens new opportunities for sensor and RFID system manufacturers [1].

Figure 5 represents the position of the various IEEE 1451 standards

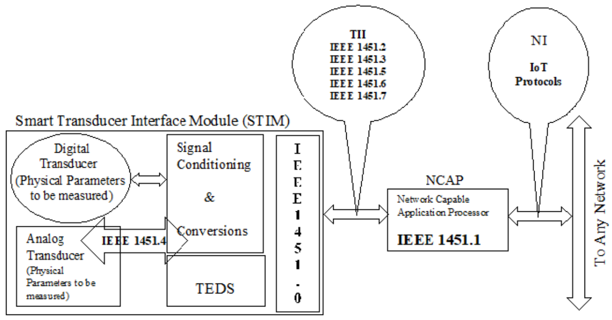

Figure 5. Depicting the position of IEEE 1451 standards

III. IoT Stack Layers [5] – [10]

IoT is not a just single technology but it is a collection of different technologies. IoT is collection of Embedded Devices, Internet and the Cloud.Figure 6 illustrates the functional diagram of IoT.

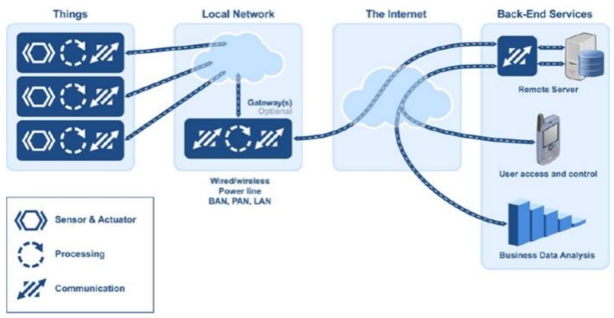

Figure 6. Functional Diagram of IoT [19]

There is no single agreement on the architecture of IoT which is agreed globally. Different architectures have been suggested by various researchers. Three and Five-Layer architecture and Cloud and Fog based Architectures are the examples of different architectures [3]. The Figure 7 represents the IoT protocols that have been standardised for each layer of the TCP/IP model.

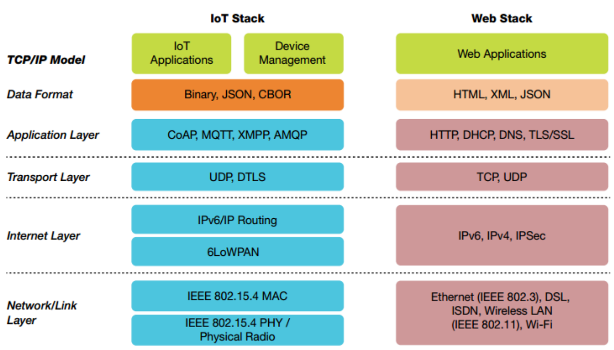

Figure 7. IoT protocol Stack [18]

Network / Link Layer –IEEE 802.15.4 is a standard for wireless communication that provides specification for the Physical Layer (PHY) and Media Access Control (MAC) layers. It is maintained by IEEE 802.15 working group. It is the base for the Wireless HART, ZigBee, ISA100.11a and MiWi specifications. These further enhance the standard by upgrading the upper layers. It can be put into use with standard Internet Protocol and 6LoWPAN to build a wireless embedded internet.

Internet Layer –The IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN) is an adaptation layer which permits IPv6 packets transmission over 802.15.4 links. This protocol operates only in the frequency range of 2.4GHz with a transfer rate of 250 kbps.An IPv6 packet (1280 bytes) is so large that it cannot fit into a single 802.15.4 frame (127 bytes).

So 6LoWPAN performs

1. Fragmentation and Reassembly: It performs the fragmentation of the IPv6 packet and then transmits the resulting multiple smaller size packets that can fit in an 802.15.4 frame. It also reassembles the fragmented packets to recreate the IPv6 packet.
2. Header Compression: It squeezes the IPv6 packet header to reduce the packet size.

Transport Layer – User Datagram Protocol (UDP) is the replacement for TCP, a connection oriented protocol. UDP is preferred because it is a connectionless protocol and has low overhead.UDP is not only much faster but also the UDP header size is much smaller than TCP and as a result, UDP is preferred for the constrained environment of devices and sensors. CoAP uses UDP rather than TCP.

Communications privacy is provided by Datagram Transport Layer Security (DTLS) for datagram protocols. This protocol permits the client/server applications to have communication in such a way that it can prevent tampering, message forgery and eavesdropping. The Transport Layer Security (TLS) is the basis of the DTLS protocol and the DTLS provides the same security features as that of TLS.

Application Layer –

1. Constrained Application Protocol (**CoAP**) is to be used in resource-constrained internet devices. The devices, sensors and actuators operate in an environment that is constrained i.e. there is low memory, low power, low bandwidth and high rate of packet failure. HTTP face difficulties to work in the constrained environment. CoAP is the substitute for HTTP. CoAP can be mapped to HTTP. CoAP follows the same pattern of request-response as adopted by HTTPCoAP uses similiar HTTP features like Methods (Post, Get, Put and Delete), URLs, Status Codes and content-type. CoAP has some differences as well. CoAP runs on UDP and HTTP runs on TCP. CoAP uses reduced and small headers (header size limited to 4 bytes).

CoAP uses EXI (Efficient XML Interchange) data format which is an efficient binary data format. Other supported features are built in header compression, resource discovery, auto configuration, asynchronous message exchange, congestion control and support for multicast messages.

There are four types of messages in CoAP: nonconfirmable, confirmable, reset and acknowledgement. For reliable transmission over UDP, confirmable messages are used. The response can be piggybacked in the acknowledgement itself. For security it uses DTLS.CoAP libraries exists in all programming languages like C, C#, Java, Python, JavaScript, etc. There are libraries for iOS and Android as well.

2. Message Queue Telemetry Transport (**MQTT**) is a publish/subscribe based "light weight" messaging protocol for M2M (Machine-to-Machine Communication) and IoT. It collects device data and communicate it to the servers. Its main areas of application are large networks of small devices that need to be controlled or monitored from the cloud. In MQTT there is

a. MQTT Publisher – A sensor or a device in the world of IoT that publishes a piece of information

b. MQTT Subscriber – Anyone who is interested in subscribing and receiving a piece of information one is interested in.

c. MQTT Broker – An intermediary that receives information from publisher and forwards them to the subscriber.

Since MQTT runs over TCP, it cannot be used with all types of IoT applications. Also it uses text for topic names, which increases its overhead.

3. Extensible Messaging and Presence Protocol (**XMPP**) is best suitable for connecting devices to people, since people are hooked up to the servers. It is an open technology that can be used for real-time exchange of information since it can power up a wide range of apps like instant messaging, presence, voice and video calls, multi-party chat.

4. Advanced Message Queuing Protocol (**AMQP**)'s main intention is that it should be applicable for message-oriented middleware. The AMQP is characterised by the important features like queuing, message orientation, routing (including point-to-point and publish-and-subscribe), security and reliability.Its a queuing system, that is designed to have connections among the servers.

Data Format – JavaScript Object Notation (**JSON**) is a lightweight syntax for data-interchange using which the humans feel comfortable for reading and writingUsing this the machines can parse and generate easily. JSON is text in nature that is written with Object Notation JavaScript.

Concise Binary Object Representation (**CBOR**), a data format designed with the goal of providing tiny messages. The super successful JSON data model is the basis for CBOR. JSON uses text encoding. As CBOR uses binary encoding it results in a message of compact size.

IoTApplications[3],[5] – [8] –The technological infrastructure provided by the IoT allows for creating and deploying many novel applications that will improve the quality of our life. Some uses of IoT applications are in Home Automation, Smart Cities – Smart Transport, Smart Water Systems, Social Life and Entertainment, Health and Fitness, Smart Environment and Agriculture, Supply Chain and Logistics, Energy Conservation and Security.

Conclusion

In this paper, a review of the IEEE 1451 family of standards and the IoT technology was performed. We described the IEEE 1451 Smart Transducers concepts and the IEEE 1451 architecture. The NCAP and STIM can be improved by developing a GPS link. A comprehensive study of the IoT protocol stack was carried out. With the rapid development in the emerging state of the art tools and technologies of IoT, we could find a lot more applications in almost all the domains of the society.

References

[1] E. Y. Song and K. Lee, "Understanding IEEE 1451-networked smart transducer interface standard," *IEEE Instrumentation and Measurement Magazine*, Vol 11, no. 2, pp. 11-17, Apr 2008.

[2] Josh Velez, Russell Trafford, Andrew DeMartino, Kevin Tomkins, IrynaTopchiy, "Developing IEEE P21451 Compatible IoT Networks Utilizing Readily Available Development Platforms," *IEEE Sensors Application Symposium*, pp 1-5, March 2017.

[3] PallaviSethi and Smruti R. Sarangi, "Internet of Things: Architectures, Protocols and Applications," *Journal of Electrical and Computer Engineering*, 25 pages, Jan. 2017.

[4] Anuj Kumar, Vishal Srivastava, Manoj Kumar Singh and Gerhard P. Hancke, "Current Status of the IEEE 1451 Standard-Based Sensor Applications," *IEEE Sensors Journal*, Vol 15, issue 5, pages 2505-2513, May 2015.

[5] KaleemUllah, Munam Ali Shah and Sijing Zhang, "Effective Ways to Use Internet of Things in the Field of Medical and Smart Health Care," *IEEE International Conference on Intelligent Systems Engineering*, pages 372-379, January 2016.

[6] MoeenHassanaliragh, Alex Page, TolgaSoyata, Gaurav Sharma, Mehmet Aktas, Gonzalo Mateos, BurakKantarci, SilvanaAndreescu, "Health Monitoring and Management Using Internet-of-Things (IoT) Sensing with Cloud-based Processing: Opportunities and Challenges,"*IEEE International Conference on Services Computing*, Pages: 285-292, July 2015.

[7] S. M. Riazul Islam, DaehanKwak, Md. HumaunKabir, Mahmud Hossain and Kyung-Sup Kwak,"The Internet of Things for Health Care: A comprehensive Survey," *IEEE Access*, Vol. 3, Pages: 678-708, June 2015.

[8] BoyiXu, Li Da Xu, HongmingCai, Cheng Xie, Jingyuan Hu and FenglinBu,"Ubiquitous Data Accessing Method in IoT-Based Information System for Emergency Medical Services," *IEEE Transactions on Industrial Informatics*, Vol. 10, issue 2, Pages: 1578 – 1586, May 2014.

[9] JIongJin, JayavardhanaGubbi, SlavenMarusic and MarimuthuPalaniswami, "An Information Framework of Creating a Smart City through Internet of Things," *IEEE Internet of Things Journal*, Vol. 1, issue 2, Pages: 112 – 121, 2014.

[10] Mohammad Mostafizur Rahman Mozumdar, ArunGanesan and AlirezaAmeri, "Synthesizing Sensor Networks Backbone Architecture for Smart Buildings," *IEEE Sensors Journal*, Vol. 14, No. 12, December 2014.

[11] MiroslavSveda and Roman Trchalik, "ZigBee-to-Internet Interconnection Architectures,"*IEEE Conference on Systems*, April 2007.

[12] Kang Lee, "IEEE 1451: A Standard in Support of Smart Transducer Networking," *IEEE Instrumentation and Measurement Technology Conference*, Vol. 2, Pages: 525-528, 2000.

[13] Bill Wright and Mike Dillon, "Application of IEEE P1451 'Smart Transducer Interface Standard' in Condition Based Maintenance," available on www.utdallas.edu.

[14] Dr.CuongDuc Truong, "Routing and Sensor Search in the Internet of Things," A Ph.D thesis submitted to the University of Lu"beck.

[15] Dr.MiloshStolikj, "Building Blocks for the Internet of Things," A Ph.D thesis submitted to the Eindhoven University of Technology.

[16] Adrian McEwen and Hakim Cassimally, "*Designing the Internet of Things*," Wiley India Publishers, 2014.

[17] Dr.OvidiuVermesan and Dr. Peter Friess, "*Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems*," River Publishers, 2013.

[18] AniruddhaChakrabarti, "*Emerging Open and Standard protocol Stack for IoT*," article on LinkedIn, December 2015.

[19] https://micrium.com/iot/devices

[20] https://standards.ieee.org/develop/project/1451-99.html

[21] https://www.postscapes.com/internet-of-things-protocols/

[22] http://postscapes2.webhook.org/internet-of-things-software-guide