A Practical Survey on Wireless Sensor Network Platforms

[1]Ghaida A. Al-Suhail, [2]Mehdi J. and [3]George Nikolakopoulos
[1]Department of Computer Engineering, University of Basrah, Basrah, Iraq,[2]Department of Computer Engineering, Al-Nahrain University, Baghdad, Iraq
[3]Department of Computer Science, Electrical and Space Engineering, Luleå University of Technology, Luleå, Sweden
Email: ghaida_alsuhail@yahoo.com, meh_elec1970@yahoo.com, geonik@ltu.se

Abstract— **In recent years, applications of wireless technologies can be seen everywhere in particular the applications of Internet of Things (IoT). The platforms at which wireless sensor networks (WSNs) are composed have witnessed great advances. In this article, a survey on WSN platforms and their characteristics such as hardware components, sensing capabilities, operating systems, programming languages, networking protocols, energy aspects, etc. is presented. A comparison between different platforms depending on their sizes, characteristics and capabilities, etc. is summarized and a selection according to specified application in different fields is also discussed.**

Keywords— Wireless Sensor Networks; Internet of Things; Platforms; Sensing; Operating Systems; Network Protocols

I. Introduction

As a result of the developments and advances of micro-electro-mechanical systems (MEMS) technology, wireless communications and digital electronics, WSNs represent a significant improvement over conventional sensor. Design and applications in embedded and wireless networked sensors are gaining tremendous developments integrated with the new era of technologies like "smart sensors", Internet of Things (IoT) and cloud computing [1-4]. These platforms are normally of a low cost and low power, small devices with limited sensing, data processing and wireless capabilities. However, IoT technology is mainly defined by the interconnection of different networked embedded devices used in the everyday life integrated into the Internet. It aims to automate the operation of different domains such as home appliances, health care systems, security and surveillance systems, industrial systems, transportation systems, military systems, electrical systems, and many others [5].

The platform of WSNs usually integrated with capabilities such as programming, different wireless communication technology such as GPS, Bluetooth, ZigBee, WiFi, cellular 2G/3G/4G LTE 5G -based IoT, storage, and sensing, and thus allowing researchers and developers to utilize more functionality and build robust systems [4][6]. New significant requirements for size, mobility, cost, network topology, life time, etc. need more investigation and deep research. To cope with these requirements, the platforms are being increased in size, computational resources, and complexity for both hardware and software [5]. More specifically, a combination of IoT hardware and software (wireless sensors and WiFi) technology produces trillions of data through connecting multiple devices and sensors with cloud and making sense of data with intelligent tools. However, anything can act as IoT devices if it can transmit and receive data over the cloud (Internet) and designed to process a unique. Thus a number of companies and research organizations have offered a wide range of projections about the potential impact of IoT on the Internet and the economy during the next five to ten years. For example, cisco projects become more than 24 billion Internet–connected objects by 2019; Huawei forecasts 100 billion IoT connections by 2025. McKinsey Global Institute suggests that the financial impact of IoT on the global economy may be as much as $3.9 to $11.1 trillion by 2025 [7]. In the late of nineties, the first commercial platform called crossbow Rene was appeared. It is emerged from a sensor node (mote) developed at university of California called Wec. A family of wireless sensor platforms such as Tmote, Mica2, MicaZ, Imote2 and BTnode has been afterwards appeared. The sensor featured low power consumption, hardware write protect, radio signal stability and good performance [8]. These nodes have been widely utilized in commercial and academic applications. Today various commercial platforms with different characteristics are available. A lot of these nodes with their characteristics such as hardware components, software programs and interfacing techniques will be presented in the sequel.

The WSN nodes can be characterized in terms of computing resources, sensor interfaces; software architectures, protocol stack, etc. while it can be utilized for different applications. Moreover, WSNs offer great advantages due to their small size, low power consumption, easy integration, and support for green applications [9]. The WSNs are composed of three individual embedded systems that are capable of interacting with their environment through various sensors, processing information locally, and communicating this information wirelessly with their neighbors. A sensor node typically consists of three components and can be either an individual board or embedded into a single system: Wireless modules or motes, sensor board and programming board [10], [11].

The platform of WSNs usually integrated with capabilities such as programming, communication, storage, and sensing, and thus allowing researchers and developers to utilize more functionality and build robust systems. New significant requirements for size, mobility, cost, network topology, life time, etc. need more investigation and deep research. To cope with these requirements, the platforms are being increased in size, computational resources, and complexity for both hardware and software [2].

In the late of nineties, the first commercial platform called crossbow Rene was appeared. It is emerged from a sensor node (mote) developed at university of California called Wec. A family of wireless sensor platforms such as Tmote, Mica2, MicaZ, Imote2 and BTnode has been afterwards appeared. The sensor featured low power consumption, hardware write protect, radio signal stability and good performance [8]. These nodes have been widely utilized in commercial and academic applications. Today various commercial platforms with different characteristics are available. A lot of these nodes with their characteristics such as hardware components, software programs and interfacing techniques will be presented in the sequel.

The WSN nodes can be characterized in terms of computing resources, sensor interfaces; software architectures, protocol stack, etc. while it can be utilized for different applications. Moreover WSNs offer great advantages due to their small size, low power consumption, easy integration, and support for green applications [9]. The WSNs are composed of three individual embedded systems that are capable of interacting with their environment through various sensors, processing information locally, and communicating this information wirelessly with their neighbors. A sensor node typically consists of three components and can be either an individual board or embedded into a single system: Wireless modules or motes, sensor board and programming board [10]-[11].

The research on WSNs platforms was mainly adopted by military applications such as biological, chemical attack detection, nuclear and battlefield monitoring. These applications were focused on ad hoc multi-hop network where a randomly distribution of immobile nodes over large area was incurred. The nodes of WSNs were very small and severely resource constrained. Thanks to great developments of electronics manufacturing, the technology emerged to civilian applications [12].

Essentially, every sensing, monitoring, and control system is made up of various sensors responsible for measuring specific physical parameters, data acquisition unit, processing unit with limited storage and radio transceiver. The units are activated by a small DC power supply usually a battery. The old monitoring system shows a star like topology where each sensor is connected to a central station, usually a personal computer, which can also called a base station. It acts as a data acquisition and storage device [13].

Wireless sensor networks are ideally suited for long-lived applications deployed at large densities for low cost. The design of WSNs is based on the following duty cycle principle: the node is asleep for the majority of the time, wakes up quickly on an event, processes and return to sleep. For the lowest power consumption, the stand by current and wake up time (time to transition from sleep to active mode) must be minimized since the active portion of the sensor network application is typically small [14]. The motivation behind writing this article is to contribute and support the efforts that have been done to build reliable and real-time applications of wireless sensor platforms. Also it presents an overview of the currently existed platforms. It can be considered as a guide for researchers and developers to be followed.

The rest of the contents are organized as follows: Section 2 illustrates the prior work related with WSNs platforms, while Section 3 addresses the hardware aspects of the platforms and the comparison between them. Section 4 presents some details related with software platforms and the programming languages and Section 5 introduces some applications of these platforms in different fields. The paper is concluded in Section 6.

II. Related Work

During the last years, many researchers have been conducted the development of WSNs platforms. Different issues related to design and implementation of such systems. Hardware architecture, software programs, protocols and power constraints are some of those issues. Polastre *et al.* [15][10] have been presented Telos platform (mote) for research and experimentation. Three issues have been discussed, minimal power computation, easy to use and increased robustness to hardware and software. The choice of hardware platform depending on radio range required, cost factor and power savings specific to different platforms such as Mica2 and MicaZ have introduced by Baleri [10].

Karl *et al.* [16] have investigated the network robustness to sensor nodes failure and the high power efficiency. Three types of sensor node hardware platforms have been categorized by Zhao *et al.* in [11]. These types are adapted as a general purpose computer, embedded sensor modules and system on chip hardware. Changsu *et al.* [17] have been introduced a wireless control system platform for intelligent home. Their platform unifies various home appliances, smart sensors and actuators and wireless communication technologies. Energy saving is of paramount importance for WSNs where Knight *et al.* [18] have been reviewed the state of the art technology in the field of energy storage and harvesting for wireless sensor nodes.

More specifically, the important applications of WSNs platforms are recently investigated, and their challenges with IoT technology are also explored by several researchers.

Among these, Song *et al.* [19] developed a mobile sensor network for monitoring applications in unfriendly environment. Other key technologies for wireless monitoring of intelligent automobile tires are extensively discussed by Matsuzaki and Todoroki [20]. In [21], Antoine-Santoni *et al.* presented a WSN as a reliable solution for capturing the kinematics of a fire front spreading over a fuel bed. A survey on wireless sensor networks and their applications can be found in [1][2][8] by Akyilidiz *et al.* They discussed also the requirements of the network design and the constraints of power consumptions, communication and data processing related to wireless sensor platforms. Several studies have devoted on the challenges of the integration of WSN and IoT. For instance, in [22] Alcaraz *et al.*, they tackle whether the devices can be completely integrated with IoT from the perspective of security at the network layer. In [3], Khan *et al.* 2015 also provided a comprehensive review of the state-of-the-art and an in-depth discussion of the research issues.

The problem of wireless sensor scheduling with a mobile sink is studied by Meheswarajaha *et al.* [14] with focus on minimizing the total energy consumed by sensor nodes while avoiding measurement losses. Regarding the software issues, Levis [23] has presented TinyOS as an operating system for wireless sensor node. Contiki as a software platform for sensor node is illustrated by Dunkels *et al.* [24] as a light weight and flexible operating system. Abrach *et al.* [25] have been investigated MANTI software for wireless sensor node.The applications of WSNs platforms are investigated and explored by several researchers. Flammini *et al.* [26] discussed some key issues related to wireless sensor networks integrated with the future of Internet of Things applications.

III. Platforms Hardware

The developments of hardware components of each platform depend on several parameters such as size, price, application, etc.

A. General Architecture

The general architecture of the wireless sensor node must contain the following basic units as shown in Fig. 1. Figure 2 depicts the picture of Tmote node with its main components like signal conditioning unit, analogue-to-digital converter (ADC), microcontroller, location finding system, memories, radio transceiver, DC powers supply. The wireless nodes can be divided mainly to sensor nodes and gateway nodes as follows [10].

- Generic nodes (take measurements)

These nodes are used to take the data from the field by measuring the physical parameters such as light, temperature, humidity, barometric pressure, velocity, acceleration, acoustics, and magnetic field. The data is exchanged between the nodes and transmitted to the gateway node and then to external world. Some of the sensors are integrated in the same node.

- Gateway (bridge) nodes

They are used to gather the data from generic sensors and relay them to the base station (laptop or computer). They must have higher processing capability, higher battery power, and higher transmission (radio) range. Table 2 illustrates some of these gateway nodes with some of their characteristics.

B. B. Basic Functionalities

The sensor nodes (motes) have to provide signal conditioning and data acquisition for different types of sensors, temporarily storage of the acquired data and data processing, self- monitoring of DC power supply, task scheduling and execution by microcontroller unit (MCU), management of sensor nodes and overall system configuration, reception, transmission and forwarding of data packets. Also it must be able to make coordination, managements of communication and networking [27]. Figure 3 shows the block diagram of the main functions of the wireless sensor node that each node must perform.

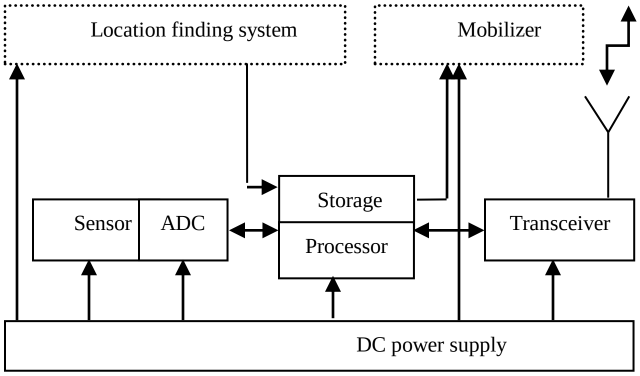

Fig. 1 Basic elements of wireless sensor node (platform).

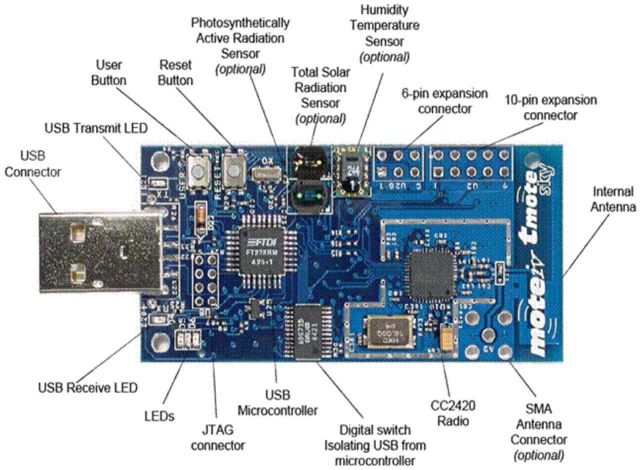
Photosynthetically Active Radiation Sensor (optional)
Humidity Temperature Sensor (optional)
User Button
Reset Button
Total Solar Radiation Sensor (optional)
6-pin expansion connector
10-pin expansion connector
USB Transmit LED
USB Connector
Internal Antenna
USB Receive LED
LEDs
JTAG connector
USB Microcontroller
Digital switch Isolating USB from microcontroller
CC2420 Radio
SMA Antenna Connector (optional)

Fig. 2 Picture of Tmote node (top layer) [10].

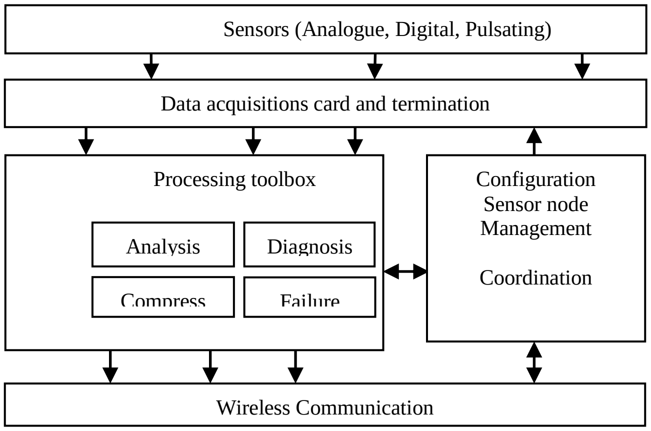

Fig. 3 Functionality block diagram.

C. *Hardware Platforms Classification*

Generally, the sensor node hardware platforms can be classified as [28].

- System on chip platforms

These platforms integrate MEMS sensors, microcontrollers and wireless transceiver with an application specific circuit. They are low power devices with small sizes.

- Embedded sensors platforms

These platforms are assembled from commercial off the shelf chips. These components are widely used, making them cheap because of big production quantities.

D. *D. Comparison between Platforms*

There are different brands of wireless sensor nodes used either as sensor or gateway (bridge) node. The comparison between them is done depending on the main units of each sensor node like MCU, available memory, transceiver, etc.

D-1 MCU and Memory

The widely used nodes in different applications with their associated MCU are listed in Table 1. It is obvious that there are two main types of the processors, TI MSP430Fxxx from Texas Instruments and ATmega128L from Atmel as they are illustrated in Table 2. It can be noted that platforms like Mica2, MicaZ [29]-[30] have the low power consumption necessary for long-term deployments of the wireless networked sensors. Their computational power and memory resources are highly restricted. For example, the widely used microcontroller MSP430F runs at speed of 8MHz and the Tmote node has only 10 Kb of RAM and 48 Kb of ROM. This lack of resources limits the types of applications the platform can support.

TABLE 1 MCUS MODULES

Sensor Node	Microcontroller	Sensor Node	Microcontroller
WisMote Dev	MSP430F5437	**K Mote**	TI MSP430
WisMote Mini	ATMEGA128 RFA2	**Mica**	ATmega 103 4 MHz 8-bit CPU
COOKIES	ADUC841, MSP430	**Mica2Dot**	ATMEGA 128
EIPC mote	TI MSP430	**Monnit WIT**	TI CC1110
GWnode	PIC18LF8722	**Mulle**	Renesas M16C
Imote	ARM core 12 MHz	**Neo Mote**	ATmega 128L
IMote 1	ARM 7TDMI 12-	**Nymph**	ATMEGA128L
INDriya_CS_03A14	ATmega 128L	**Pow Wow**	MSP430F1612
Iris Mote	ATmega 1281	**Preon32**	ARM Cortex M3
Sense Node	MSP430F1611	**Fire Fly**	ATmega 1281
Shimmer	MSP430F1611	**Ubimote1**	TI's CC2430 SOC based on 8051 Core
Sun SPOT	ARM 920T	**Ubimote2**	TI's CC2430 SOC based on 8051 Core
Telos B	TI MSP430 microcontroller	**Fire Fly**	ATmega 1281
Tiny node	TI MSP430	**Wasp mote**	Atmel ATmega 1281
T-Mote Sky	TI MSP430	**XYZ**	ML67 series ARM/THUMB
XM1000	TI MSP430F2618	**Zolerta Z1**	TI MSP430F2617

TABLE 2 MCUS VERSIONS

Version of MCU	Bus	Clock
TI MSP430F1611	16-bit	4-8 MHz
Atmega128L	8-bit	8 MHz
TI MSP430F2274	16-bit	16 MHz
Atmega1281	8-bit	8 MHz
Atmega9200	32-bit	180 MHz

While many other new mote platforms have been developed [13], most of these platforms still based on similar MCU as the one used in Tmote node described above. Some of the platforms like Imote2 provided developers with higher computation power (processing speed up to 180 MHz) with 32 Mb of ROM and 32 Mb of RAM. This came at the expense of energy efficiency where it could not be used for long term and unattended deployments. The small size of Imote2 made it preferred for some applications. Table 2 illustrates the different versions of most commonly MCU modules while Table 3 indicates the available memories for each node.

D-2 Transceiver

The widely used transceiver is that one from Chipcon which comes with different versions CCxxxx. It is used in motes listed in Table 4. It has an embedded antenna with frequencies 2.4GHz and 310.433MHz.

The transmitting range is 30 to 125 m. The network protocol for this transceiver is the well-known IEEE802.15.4. There's another chip called Zeevo (model Zv4002) used in BTnode rev3. This chip utilizes three frequencies and it depends on Bluetooth IEEE 802.15.1 protocol for network communications. Table 5 depicts some characteristics of this transceiver used by different wireless sensor nodes.

D-3 Interfacing and Networking

The current implementation of platforms supports peripheral controllers including USART, UART, ADC, I^{2}C, SPI and camera. The platform must be able to adapt with different sensors to collect data from physical world. The data acquisition card in each platform is characterized by number of channels, sampling rate, and resolution for either its ADC or DAC.

Some platforms provide multiple interfaces including Ethernet, WiFi, USB, or serial ports for connecting different motes to an enterprise or industrial network or locally to a PC/laptop. These boards are used either to program the motes or gather data from them [29].

There is a current interest to connect all networking objects, including miniature sized sensor motes, to the Internet using technologies such as 6LoWPAN. While providing each mote with an IPv6 stack has multiple advantages, the large memory and code size overhead of the address compression techniques may leave only a small amount of space for the user's application code to fit. The additional resources that PXA271 microcontroller (from Intel) based mote platforms such as Imote2 can be used to accommodate more complex user interfacing that previous motes cannot support [2].

TABLE 3 PROGRAM AND DATA MEMORY

Sensor Node	Program and Data Memory	Sensor Node	Program and Data Memory
WisMote Dev	RAM : 16 Kbytes flash : 256 Kbytes	**Mica**	128+4 KB RAM
WisMoteMini	RAM : 16 Kbytes flash : 128 Kbytes E^{2}PROM : 4 Kbytes	**Mica2Dot**	4 KB RAM
COOKIES	4 Kbytes + 62 Kbytes	**Monnit WIT**	4 KB RAM
EIPC mote	10 KB RAM	**Mulle**	31 KB RAM
GWnode	64 KB RAM	**Neo Mote**	4 KB RAM
Imote	64 KB SRAM	**Nymph**	
IMote 1	64 KB SRAM	**Pow Wow**	55 KB flash + 5 KB RAM
INDriya_CS_03A14	128 KB FLASH + 4 KB RAM	**Preon32**	64KB RAM + 256KB flash
Iris Mote	8 KB RAM	**Rene**	512 bytes RAM
K Mote	10 KB RAM	**Sense Node**	10 KB RAM
Shimmer	48 KB flash 10 KB RAM	**T-Mote Sky**	10 KB RAM
Sun SPOT	512 KB RAM	**Wasp mote**	8 KB SRAM
Telos B	10 KB RAM	**XM1000**	8 KB RAM
Tiny node	8 KB RAM	**Zolertia Z1**	8 KB RAM

TABLE 5 TRANSCEIVER VERSIONS

Transceiver model	Data rate	Frequency (MHz)	Tx power (dBm)	Rx sensitivity (dBm)
CC1000	76.8kBaud	300-1000	-30	-110
CC2420	250kbps	2400-2483.5	-24	-92
CC2500	500 kBaud	2400-2483.5	-55.8	-101

D-4 Miscellaneous Specifications

Other aspects that related with hardware components are the power supply which is usually a rechargeable or non-rechargeable battery. The life time of the node depends on the battery capacity. It comes in terms of required voltage and consumption of current. Some platforms have a life time of 1 year and others are more like Imote2. It has been noticed that most energy is consumed by transceiver through transmit and receive of data during the active and idle mode, while a very small energy is used by MCU through coding and decoding process [19].

The on-board sensors are limited to temperature and humidity. Some platforms may contain special sensors such as accelerometer, magnetic and light and that depends on the application in hand. The sensing and communication ranges are very important specifications for each node and that depends on the size of the node [2].

The leader for platforms manufacturer and developer is crossbow and Moteiv is the second. The prices of the platforms always subject to change and depends on the options and accessories, but it ranges from 70$ to 250$.Table 6 collects miscellaneous information related with three sensor nodes [29].

IV. Platforms Software

There are three aspects related to the platforms software: operating systems, programming languages and simulators.

A. Operating Systems

The operating systems are highly tailored to the limited node hardware. Their frame work is not a full blown operating system since they lack a powerful scheduling of tasks, memory managements and file system support. One of the wide spread operating software is the TinyOS. It is written in nesC [10], a language extended from C language. TinyOS supports event-driven component-based programming (a certain sensor reading exceeds a threshold) or from other components, triggering a specific action. The program is decomposed into functionally self-contained components. They interact by exchanging messages through interfaces. Contiki [11] is another operating system that can be used with wireless sensor nodes. It is designed to run on classes of hardware devices that are severely constrained in terms of memory, processing power and communications bandwidth. A typical Contiki system has memory on the order of several kilobytes.

TABLE 6 MISCELLANEOUS DATA FOR SOME SENSORS

Sensor Node		Tmote	Mica2	Imote2
Power supply	Voltage(v) min,max	2.1, 3.6	2.7, 3.3	3.2, 5
	Current (mA)			
	min, max	21.8, 1.8	39, 12	44-66, 31
Battery life	Year	<1	1	>1
Dimensions	W x L x H	6.6x3.2x0.7	6.6x3.2x0.7	4.8x3.6x0.9
	cm x cm x cm			
On-board sensors	Physical quantity	Temp., Humidity,	Temp., Humidity,	Temp., Humidity, light
Manufacturer	Company	Motiev	Crossbow	Crossbow
Prices	$	80	90	170

LiteOS [31] is a real time operating system (RTOS) designed to use in wireless platforms. LiteOS is a UNIX-like operating system that fits on memory-constrained sensor nodes. This operating system allows users to operate wireless networked sensors like UNIX. LiteOS provides a familiar programming environment based on UNIX, threads, and C. It follows a hybrid programming model that allows both event-driven and thread-driven programming. It is an open source, written in C and runs on the Atmel based MicaZ and Mica2 [29] etc. sensor networking platforms.

B. Programming Languages

The programming language that used in most platforms is C language. Network embedded systems C (nesC) language, is a component oriented programming, event-driven language used to build applications for the TinyOS platform. nesC is built as an extension to the C programming language with components "wired" together to run applications on TinyOS. Some platforms are programmed by C# like Telos B [32], while platform Iris mote is programmed by C++. Java is also used to program the wireless platforms such as in Nymph mote [33]. Table 7 illustrates the programming languages for different reviewed sensor nodes.

C. Simulators of WSN Platforms

There are many simulators can be used to simulate the deployment of platforms to conduct the analysis and expect the performance of the sensor nodes in the network. NetSim [34] is a popular network simulation tool used for network lab experimentation and research. Various wireless technologies such as WSNs, WLAN, WiMAX, TCP/IP, etc. are covered in NetSim simulator. OMNeT++ [35] is another simulator used with wireless sensor. It is a component-based, modular and open-architecture discrete event simulator framework. The most common use of OMNeT++ is for platforms simulation. NS-3, Castalia [36] and OPNET [37] are simulators for wireless networked platforms used widely with similar characteristics of others.

IV. Applications of Platforms

The applications that utilize the wireless platforms are huge ranging from simple monitoring tasks to supervisory monitoring and control of large facilities. WSNs may consist of diverse types of sensors including seismic, magnetic, thermal, visual, infrared, acoustic, and radar, which are able to monitor a wide variety of ambient conditions that include the following: temperature, humidity, pressure, speed, direction, movement, light, soil makeup, noise levels, the presence or absence of certain kinds of objects, and mechanical stress levels on attached objects. Hence, a wide range of applications are possible.

This wide spectrum of applications includes homeland security, monitoring of space assets for potential and human-made threats in space, ground based monitoring of both land and water, intelligence gathering for defense, environmental monitoring, urban warfare, weather and climate analysis and prediction, battlefield monitoring and surveillance, exploration of the solar system and beyond, monitoring of seismic acceleration, strain, temperature, wind speed and graphical positioning system data [38].

TABLE 7 PROGRAMMING LANGUAGES

Sensor Node	Programming Language	Sensor Node	Programming Language
WisMoteDev	C Programming	**Mica**	C# Programming
WisMoteMini	C Programming	**Mica2Dot**	nesC, C programming
COOKIES	C Programming	**Monnit WIT**	nesC programming
EIPC mote	C Programming	**Mulle**	C programming
G Wnode	C Programming	**Neo Mote**	C programming
Imote	C Programming	**Nymph**	Java
IMote 1	C Programming	**Pow Wow**	nes C Programming
INDriya_CS_03A14	C-programming & nesC compliant	**Preon32**	C and NesC programming
Iris Mote	nesC	**Rene**	nes C and C
K Mote	C Programming	**Sense Node**	Java
Shimmer	nesC Programming	**T-Mote Sky**	C Programming, nesC
Sun SPOT	nesC Programming	**Wasp mote**	C Programming
Telos B	C# Programming	**XM1000**	C Programming
Tiny node	C Programming	**Zolertia Z1**	C Programming

These ever-increasing applications of WSNs can be mainly divided into five categories: military, environmental, health care, home and industrial. In the military applications, the platforms are used for battlefield surveillance, troop monitoring, target tracking and intelligence. Mica2 platform is used in some of these applications. Environmental applications can be implemented by several motes such as Tmote, Mica2, Imote2, etc. It is used for forest detection, flood and structural health monitoring [2]-[13][29-30].

Applications in real time systems depend on the latency time of the node. It may be fractional of seconds for detection of nuclear radiation and months for environmental monitoring, so the selection of a certain platform node depends on the system requirements in hand [1]-[[39].

V. CONCLUSIONS

In this study, it has been noticed that wireless sensor network platforms can be used in everyday life. It is developing rapidly due to its large number of applications based on Internet of Things (IoT) technology that it can be used. The important issue that needs more research and development is the data transmission. However, it yet consumes the large amount energy which leads to decreasing the life time of the node and the overall life time of sensor network. As a result, the reliability of the overall system in terms of quality of service metrics (i.e., packet loss, jitter, delay and throughput) will be affected.

REFERENCES

[1] I. F. Akyilidiz and M. C. Vuran,” Wireless sensor networks,” John-Wiley & Sons Ltd. 2010.

[2] I. F. Akyilidiz, W. Su, Y. Sankarasubramaniam and E. cayrici,” Wireless sensor networks survey,” ELSEVIER, Computer Networks 38, pp.393-422, 2002.

[3] I. Khan, F. Belqasmi, R. Glitho R. N. Crespi, M. Morrow and P. Polakos, "Internet of Things: Wireless Sensor Networks," IEEE Communications Surveys &Tutorials, 2015.

[4] Y. Zhan, L. Liu, L. Wang, and Y. Shen, "Wireless Sensor Networks for the Internet of Things, " International Journal of Distributed Sensor Networks, 2013.

[5] L. Sanchez, et al., “SmartSantander: IoT Experimentation over a Smart City Testbed,” Computer Networks, vol. 61, pp. 217–238, March.2014.

[6] M. Jaradata et al., "The Internet of Energy: Smart Sensor Networks and Big Data Management for Smart Grid," The International Workshop on Networking Algorithms and Technologies for IoT (NAT-IoT 2015) Procedia Computer Scienc, 2015.

[7] Rose K., Eldridge S. and Chapin L. (2015) "Internet of Things: Overview", Internet Society.

[8] S. K. Mazumder,” Wireless networking based control,” Springer, New York, 2011.

[9] F. Xia,” Wireless sensor technologies and applications,” Sensors, doi: 10.3390/s91108824, 2009.

[10] G. Baleri, “Guidelines for WSN design and deployment,” Application Notes, Crossbow Technology, 2009.

[11] F. Zhao and L. Guibas,” Wireless sensor networks: An information processing approach,” Morgan Kaufman: San Francisco, CA, 2005.

[12] C. Buratti, A Conti, D. Dardiri and R. Verdone,”An overview of wireless sensor networks technology and evolution,” Sensors, 2009.

[13] I. F. Akyiydiz and I. Kasimoglu,” Wireless sensor and actuator network: Research challenges,” ad hoc network, 2004.

[14] D. Maheswararajah, S. Halgamagu and M.Prematne,” Energy efficient sensor scheduling with a mobile sink node for the target tracking applications,” Sensors, September, 2009.

[15] J. Polasre, R. Szewcyk and D. Culler,” Telos: Enabling ultra-low- power research, “in Information processing in sensor networks/SPOTS: Berkeley, 2005.

[16] H. Karl and A. Willig,” Protocols and architecture for wireless sensor networks,” John-Wiley& sons, 2005.

[17] S. Changsu and K. Young-Bae,” Design and implementation of intelligent home control system based active sensor networks,” IEEE Transaction on Consumer Electronics, Vol. 54, No.3, August, 2008.

[18] C. Knight, J. Davison and S. Behrens,” Energy options for wireless sensor nodes,” Sensors, August, 2008.

[19] G. Song, Y. Zhou, F. Ding, A. Song,” A mobile sensor network system for monitoring of unfriendly environments,” Sensors, 2008.

[20] R. Matsuzaki, A. Todoroki,” Wireless monitoring of automobile tiers for intelligent tires,” Sensors, 2008.

[21] T. Antoine-Santoni, J.-F Santucci, E. Gentili, X. Silvani, F. Morandini,” Performance of a protected wireless sensor network in a fire. Analysis of fire spread and data transmission,” Sensors, 2009.

[22] Alcaraz C., PNajera P., Lopez J. and Roman R. (2010) "Wireless Sensor Networks and the Internet of Things: Do We Need a Complete Integration?," International Workshop on the Security of the Internet of Things (SecIoT10), 2010.

[23] SP. Levis,” TinyOS: An operating system for wireless sensor network,” Ambient Intelligence, Springer-Verlage, 2006.

[24] A. Dunkels, B. Grenvoll and T. Voigt, “Contiki- a light weight and flexible operating system for tiny networked sensors,” in Proceedings of the 29th Annual IEEE International Conference on Local Computer networks, Washington DC, 2004.

[25] H. Abrach, S. Bhatti, J. Carlson and J. Rose, “MANTIS: system support for multi model network of in situ sensors,” in International Conference on Wireless Sensor Networks and Applications, New York, 2003.

[26] A. Flammini and E. Sisinni 2014, "Wireless Sensor Networking in the Internet of Things and Cloud Computing Era," Procedia Engineering, Vol. 87, pp. 672-679, 2014.

[27] S. Madden, M. Franklin, J. Hellerstein and W. Wang,” The design of an acquisitioned query processor for sensor network,” in Proceedings of SIGMOD, June, 2003.

[28] V. Handziski, J. Polastre, C. Sharp, A. Wolisz and D. Culler,” Flexible hardware abstraction for wireless sensor networks,” in (EWSN), February, 2005.

[29] Crossbow technology Inc.,” MICAz wireless measurement system” http://www.xbow.com, June, 2008.

[30] http://www.btnode.ethz.ch

[31] 24 Q. Cao, T. Abdelzaher, J. Stankovic, and T. He, "The LiteOS Operating System: Towards Unix-Like Abstractions for Wireless Sensor Networks," Proceedings of the 7th international conference on Information processing in sensor networks, 2008.

[32] http://www.monnit.com

[33] http://www.javaoracleblog.com/java/SunSPOT.jsf

[34] http://www.computer.org/portal/web/csdl/doi/10.1109/CSSE.2008.27

[35] http://www.omnetpp.org/home

[36] http://castalia.npc.nicta.com.au/pdfs/Castalia_AcademicPublicLicence.pdf

[37] http://www.OPNET.com

[38] J. Polastre,”Interfacing Telos to 51-pin sensor board,” http://www.tinyOS.net/hardware/telos-legacy-adapter.pdf, October, 2004.

[39] J. Hill and D. Culler, “Mica: A wireless platform for deeply embedded networks,” IEEE Micro, Vol.22, No.26, November, 2005.