ISSN No. 0976-5697
Volume 4, No. 3, March 2013 (Special Issue)
International Journal of Advanced Research in Computer Science
CASE STUDY AND REPORT
Available Online at www.ijarcs.info

A Survey of Wireless Sensor Network Hardware Platforms and Simulation Tools

Manoj Kuri
Assistance Professor, Deptt. of Electronics & Communication Engineering Engineering College
Bikaner, India,
mkuri@rediffmail.com

Virendra Choudhary
Associate Professor, Deptt. of Electronics & Communication Engineering Engineering College
Bikaner, India,
virendrachoudhary@yahoo.com

Prof. K.R Chowdhary
Professor, Dept. of Computer Science and Engineering
M.B.M. Engg. College, J.N.V. University Jodhpur, India
kr.chowdhary@acm.org

Abstract: To give a general picture of main-stream hardware platforms used in Wireless Sensor Networks (WSNs), and help people to choose different sensor nodes platforms according to different needs, a comparison of commonly used sensor node hardware platforms has been presented in this paper. Simulation tools for wireless sensor networks are increasingly being used to study sensor networks and to test new applications and protocols in this evolving research field. This paper provides comparisons of various popular sensor network hardware platforms and simulators with a view to help researchers choose the best available for a particular application environment.

Keywords: Wireless Sensor Networks, Simulator, Emulator, Graphical User Interface (GUI)

I. INTRODUCTION

A wireless sensor network (WSN) is a wireless network consisting of spatially distributed autonomous devices using sensors to cooperatively monitor physical or environmental conditions. The development of such networks was originally motivated by military applications such as battlefield surveillance. However, wireless sensor networks are now used in many civilian application areas, including environment, healthcare applications, home automation, industrial process control, field irrigation and disaster management [1].

The data collected by sensors and sensor nodes (also called as motes) is transmitted to a special node equipped with higher processing capabilities called "Base Station" or "sink" or "Gateway". The sink collects filters and aggregates data sent by sensor nodes. Data from these sink nodes may be further transmitted to the Internet or any other remote network [2].

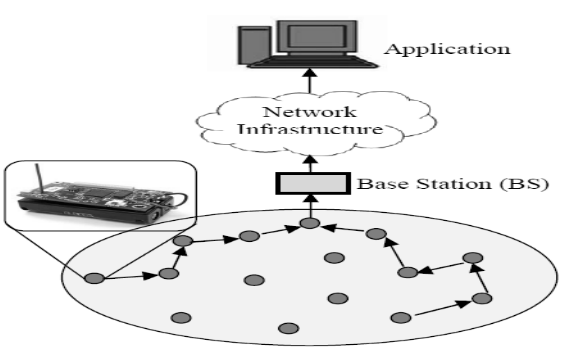

Figure1: Wireless Sensor Network (WSN) Architecture

II. WIRELESS SENSOR NODE ARCHITECTURE

A typical Wireless Sensor Network (WSN) is built of several hundreds or even thousands of sensor nodes. Each node has the ability to communication with every other node wirelessly, thus a sensor node is made up of five basic entities: sensors, processor, memory, radio, and power entity. They may also Akyildiz et al. [3] have application dependent additional components such as location finding system, a power generator and a mobilizer. Sensors are electronic devices that are capable to detect environmental conditions such as temperature, sound, chemicals, or the presence of certain objects. They send detected values to the processor which runs the sensor operating system and manages the procedures required to carry out the assigned sensing task. This processor retrieves the application code from the memory unit which stores also the operating system and the sensed values. The radio permits to the wireless sensor nodes to communicate with other nodes, updates from the sink and to send sensed data to the sink.

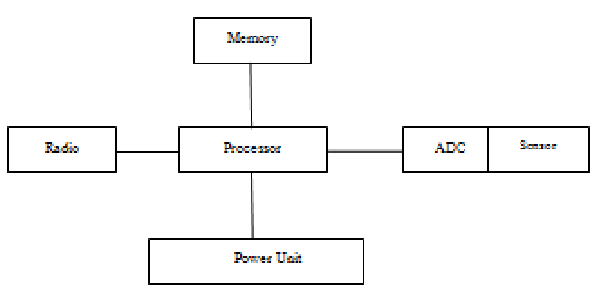

Figure2: Basic Architecture of Sensor Node

CONFERENCE PAPER
II International Conference on
"Advance Computing and Creating Entrepreneurs (ACCE2013)"
On 19-20 Feb 2013
Organized by
2nd SIG-WNs, Div IV & Udaipur Chapter , CSI , IEEE Computer Society Chapter India Council , IEEE Student Chapter Geetanjali Institute of Technical Studies, Udaipur, Rajasthan, India

The key element in a sensor node is the power entity which is generally composed of a couple of batteries. The size of these batteries usually determines the size of the sensor. Further, studies Baronti et al. (2007) are currently under way to replace/integrate battery sources with some power scavenging methods such as solar cells. In fact, there are some limits about the actual effectiveness of such methods.

III. COMPARISON OF WIRELESS SENSOR PLATFORMS

In this section comparison of some of the important features of sensor nodes has been such as Imote[4],Imote2 [5], TelosB [6], Mica-family which includes Mica1 [7], Mica2 [8], Micaz [9,10], Tmote[11], Rene[12], Stargate[13] etc. Some of these platforms are only for specific use, but on the other hand some of them are for general purpose use and these sensor nodes having large number of extension connecters to integrate various sensor boards for different applications. The early sensor platforms were designed with limited processing power, less on board memory and limited energy supplied by the batteries, so their applications were very limited which requires low processing capability and consumes less energy[14]. Now a day's these sensor nodes comes with high processing capabilities and have batteries with power generation facilities such as solar power or lithium rechargeable batteries or any other methods. So, these sensor nodes may be used for applications which requires high processing capabilities, large memory, higher data rates and power consuming, such as Multimedia applications in which camera and videos are integrated with these nodes and are used for image and video transfer and processing applications, these types of networks are called as Wireless Multimedia Sensor Networks (WMSNs) and today almost all the sensor nodes are developed for these WMSNs compatible [15].
In table below comparison of the features of wireless sensor nodes are given. Comparison is made on the basis of processor, storage capacity, radio and available bandwidth or maximum data rate.

Table 1: Comparison of the Features of Wireless Sensor Nodes

Node	CPU	Memory	Sensors and I/O	Radio	Data Rate (Kb/s)
Imote 1.0	ARM 7TDMI 12-48 MHz	64KB SRAM, 512 KB Flash	USB,UART, GPIO,I2C,SPI	Bluetooth (range 30 m)	500
Imote 2.0	PXA271 Intel XScale (32 bit)	256 KBSRAM, 32MB SDRAM 32 MB flash	I2C, SPI, UART, CIF, USB, SDIO	ChipCon CC2420 (IEEE 802.15.4 based)	250
TelosB	TI MSP430 microcontroller, (16 bit),8MHz	10KB RAM, 1MB Flash	Microphone, Speaker, Accelerometer, Light Sensor	IEEE 802.15.4 (CC 2420 radio)	250
Mica1	Atmel ATmega 103L (8 bit), 4 MHz	4K RAM, 128K Flash	Sensor board with temperature, light, pressure, humidity, infrared	TR1000 radio trans-receiver.	40
Mica2	Atmel ATmega 128L (bit), 7.37 MHz	4K RAM, 128K Flash	Large expansion connector for Microphone, Photo Sensor, Accelerometer, Magneto-meter.	RFM & CC1000 chipcon (315, 433 or 868/916, MHz) multi-channel radio transceiver.	38.4
MicaZ	Atmel ATmega 128L (8 bit), 7.37 MHz	4K RAM 128K Flash	Large expansion connector for light, temp., pressure, acceleration, seismic, acoustic, magnetic	CC2420 (802.15.4/ ZigBee) RF transceiver	250
Tmote Sky	TI MSP430 microcontroller, (16 bit) 8MHz	10KB RAM and 48k Flash	Integrated Humidity, Temperature, & Light sensors	2.4GHz IEEE 802.15.4 Chipcon Wireless Transceiver	250
Rene	ATMEL AT90LS 8535	512B RAM 8K Flash	Sensor board with Large expansion connector	RF Monolithics, TR1000 (916 MHz radio)	19.2
Stargate	Intel PXA255 Xscale, (32 bit), 400 MHz	64 MB SDRAM 32 MB Flash	PCMICA/ CF, com ports, USB, Ethernet, GPS receiver,	802.11b, CC2420 (IEEE 802.15.4)and Bluetooth	250 Kbps – 11 Mbps

IV. CLASSIFICATION OF SENSOR PLATFORMS

Depending on their processing power and storage capacity and sensors integration and radios, these wireless nodes can be classified into three classes:

a. ***Lightweight-class Platforms:*** The devices in this category are designed initially for detecting scalar data, such as temperature, light, humidity etc., and their main concern is to consume less amount of energy as possible. Therefore, these devices have low processing power capability and small storage and most of them are equipped with a basic communication radio which consumes less power while sending and receiving data. In table Table1

CONFERENCE PAPER
II International Conference on
"Advance Computing and Creating Entrepreneurs (ACCE2013)"
On 19-20 Feb 2013
Organized by
2nd SIG-WNs, Div IV & Udaipur Chapter , CSI , IEEE Computer Society Chapter India Council ,
IEEE Student Chapter Geetanjali Institute of Technical Studies, Udaipur, Rajasthan, India

examples of lightweight-class wireless motes are shown, which are Mica-family [16] motes and Telos.

b. ***Intermediate-class Platforms:*** The devices in this group have better computational and processing capabilities and larger storage memory than lightweight-class devices. However, they are also equipped with low bandwidth and data rate communication module. TmoteSky[17] is an example of Intermediate-class mote designed by Moteiv (now Sentilla) that uses low power 8 MHz 16-bit MSP430 F1611 RISC processor from Texas Instruments featuring10kB of RAM, and 48kB of flash. Tmote Sky uses Chipcon CC2420 radio [18] for IEEE 802.15.4/ Zigbeefor maximum data rate of 250 Kbps. It is having extension for integration with cameras.

c. ***PDA-class Platforms:*** The devices in this category are more powerful in terms of computational and processing power and they are designed to process multimedia content in a fast and efficient manner. These devices can run different operating systems (e.g., Linux, TinyOs, and run Java applications) and support multiple radios with different data rates (e.g., IEEE 802.15.4,IEEE 802.11, and Bluetooth). However, these devices consume relatively more energy. Stargate andImote2 are examples of PDA-class platforms. Stargate board, designed by Intel and manufactured by Crossbow, uses 400 MHz 32-bit Marvell's PXA255 XScale RISC processor with 32 MB of Flash memory and 64 MB of SDRAM and runs Linux operating system. Stargate board can be used as a sensor network gateway, robotics controller card, or distributed computing platform. It forms a camera mote when it is connected with camera device. So can be used as camera node for wireless multimedia sensor networks (WMSNs)[19].

Imote2, also designed by Intel and manufactured by Crossbow, is a wireless sensor node platform built around the low-power32-bit PXA271 XScale processor and integrates an 802.15.4 radio (CC2420) with a built-in 2.4 GHz antenna and includes 256 KBSRAM, 32 MB Flash memory, 32 MB SDRAM, and several I/O options. It can run different operating systems such as TinyOs and Linux with Java applications and it is also available with .NET micro framework. It integrates many I/O options making it extremely flexible in supporting different sensors including cameras, A/Ds, radios, etc and can be used multimedia and video operations.

V. SIMULATION IN WSNS

Running real experiments on a testbed[20] is costly and difficult. Besides, repeatability is largely compromised since many factors affect the experimental results at the same time. Moreover, running real experiments are always time consuming. Therefore, WSNs simulation is important for WSNs development. Consequently, simulation is essential to study WSNs, being the common way to test new applications and protocols in the field. This leads to the recent boom of simulator development.

A. *Simulator and Emulator:*

Simulator is universally used to develop and test protocols of WSNs, especially in the beginning stage of these designs. The cost of simulating thousands of nodes networks is very low, and the simulation can be finished within very short execution time. Both general and specialized simulators are available for uses to simulate WSNs. The tool, which is using firmware as well as hardware to perform the simulation, is called emulator. Emulation can combine both software and hardware implementation. Emulator implements in real nodes, thus it may provide more precision performance. Usually emulator has highly scalability, which can emulate numerous sensor nodes at the same time.

There are basically three types of simulation: Monte Carlo Simulation, Trace-Driven Simulation and Discrete-Event Simulations [21]. The last two simulations are used commonly in WSN. Discrete-event simulation is widely used in WSNs, because it can easily simulate lots of jobs running on different sensor nodes. Discrete-event simulation [22] includes some of components. This simulation can list pending events, which can be simulated by routines. The global variables, which describe the system state, can represent the simulation time, which allow the scheduler to predict this time in advance. This simulation includes input routines, output routines, initial routines, and trace routines. In addition, this simulation provides dynamic memory management, which can add new entities and drop old entities in the model. Debugger breakpoints are provided in discrete-event simulation, thus users can check the code step by step without disrupting the program operation.

However, Trace-Driven Simulation provides different services. This kind of simulation is commonly used in real system. The simulation results have more credibility. It provides more accurate workload; these detail information allow users to deeply study the simulation model. Usually, input values in this simulation constant unchanged. However, this simulation also contains some drawbacks. For example, the high-level detail information increases the complexity of the simulation; workloads may change, and thus the representativeness of the simulation needs to be suspicious.

VI. SIMULATION TOOLS COMPARISON

Both general simulators and specific simulators [23, 24] are evaluated in this section. The general simulators usually lack some functions to provide specific simulations in WSNs, however specific simulators with more comprehensive functions may perform better. According to different targets to choose different simulation tools in WSNs will be more efficient and effective.

In the table below comparison for simulation tools commonly used simulators in WSNs: NS-2[25, 26], TOSSIM[27,28], OMNeT++[29], EmStar[30], JavaSim [31], and GlomoSim [32] and ATEMU[33] analyzes the merits and limitations of each simulation tool.

Table 2: Comparison of WSNs Simulators/Emulators

Tool Name	Emulator or Simulator	Features	General Purpose or specific	Merits	Limitations
Network Simulator (NS)-2	Discrete event simulator	Built in Object-Oriented extension of Tool Command Language(OTcl) and C++	Non-specific network simulator or can used in both wire and wireless area.	Open source model. It can support a considerable range of protocols in all layers.	Complex and time-consuming than other simulators. Trouble to simulate more than 100 nodes. Poor graphical support. Difficult to add new protocols or node components
TOSSIM	bit-level discrete event network emulator	Built in Python, a high-level language emphasizing code readability, and NesC code, ,runs on TinyOS	Specifically designed for WSN	Can support thousands of nodes simulation and has a good GUI (TinyViz). Each node can be evaluated under perfect transmission condition. Can support more precise simulation result at component levels	Cannot simulate the performance metrics of other new protocols. Energy consumption simulation is only possible with Power TOSSIM simulator extending the power model to TOSSIM. Emulate only of homogeneous type applications.
OMNeT++	a discrete event network simulator built in C++	Supports module programming model which is specifically used for WSNs simulation and has a mobility frame work.	Non-specific network simulator, which can be used in both wire and wireless area	Strong GUI, makes tracing and debugging much easier. Can support MAC protocols as well as some localized protocols in WSN. Can simulate node power consumption.	Number of available protocols is not larger enough. The compatible problem, since individual researching groups developed the models separately.
EmStar	a trace-driven emulator	Built in C emulator and supports to develop WSN application on better hardware sensors	Specifically designed for WSN	Good GUI & Modular programming model allows the users to run each module separately. It can mitigate faults among the sensors, and debug and evaluate much easier.	Cannot support large number of sensors simulation, and the limited scalability. EmStar is can only run in real time simulation.
J-Sim (Java Sim)	discrete event network simulator	Built in Java simulator. Models in J-Sim have good reusability and interchangeability, which facilities easily simulation.	Was not originally designed to simulate WSN	Can simulate larger number of sensor nodes. can also support data diffusions, routings and localization simulations. Can simulate radio channels and power consumptions in WSNs.	The execution time is much longer than that of NS-2. Inherently design of J-Sim makes users hardly add new protocols or node components.
GloMoSim	discrete event network simulator	Built in Parsec (a C-based discrete event simulation language).	Specifically for mobile wireless networks	Layered architecture with easy plug-in capability. Modular, extensible library for network models.	Support for fixed protocol layers.
ATEMU	discrete	Emulator of an AVR	specifically	Emulate homogen	Simulation time is

CONFERENCE PAPER
II International Conference on
"Advance Computing and Creating Entrepreneurs (ACCE2013)"
On 19-20 Feb 2013
Organized by
2nd SIG-WNs, Div IV & Udaipur Chapter , CSI , IEEE Computer Society Chapter India Council ,
IEEE Student Chapter Geetanjali Institute of Technical Studies, Udaipur, Rajasthan, India

	event network Emulator	processor for WSN built in C; AVR is a single chip micro controller in the MICA platform	designed for WSNs	eous or heterogeneous network[34]. Can emulate power consumptions or radio channels.	much longer

VII. CONCLUSION

There are many different sensor nodes platforms are commercially available now a days, each of them is designed for a specific or general purpose use. Each one is having its own benefits and limitations, but their use depends upon the need, size, application or purpose of the sensor network is designed and suitability of the sensor node for that sensor network and currently there is no common WSN simulation tool. There is always an overriding concern when using simulation that the results may not reflect accurate behavior. It is therefore essential to know the strengths and weaknesses of these simulators, which are being discussed in this work.

VIII.REFERENCES

[1]. Sensor Networks: Evolution. Opportunities and Challenges. Chee-Yee Chong, Srikanta P. Kumar, proceeding of the IEEE, vol. 91, August 200.

[2]. Wireless Sensor Network Platforms, Reinhard Bischoff, Jonas Meyer and Glauco Feltrin Encyclopedia of Structural Health Monitoring, John Wiley & Sons, 2009.

[3]. Wireless sensor networks: A Survey, I.F. Akyildiz", W.Su, Y. Sankarasubramaniam, E. Cayirci, Elsevier, Computer Networks, 2002.

[4]. Intel Mote: Sensor Network Technology for Industrial Applications. Ralph Kling, Robert Adler, Jonathan Huang, Vincent Hummel, Lama Nachman, Intel Corporation Research, 2004.

[5]. Imote2: Serious Computation at the Edge. Lama Nachman, Jonathan Huang, Junaith Shahabdeen and Robert Adler, IEEE, 2008.

[6]. TelosB.URL:http://www.willow.co.uk/TelosB_Data sheet. Description: an introduction of TelosB.

[7]. Mica. URL:http://stomach.v2.nl/docs/ Hardware/Data Sheets/ Sensors/ MICA_data_sheet.pdf. Description: an introduction of Mica sensor board.

[8]. Introduction to Crossbow Mica2 Sensors, Prasanna Ballal, Automation & Robotics Research Institute, University of Texas at Arlington.

[9]. Mica2.URL:https://www.eol.ucar.edu/rtf/facilities/is a/ internal / CrossBow/DataSheets / mica2.pdf.

[10]. Micaz, URL: www.memsic.com/ support/ documentation/ wireless-sensor-networks/categorydatasheets.html/download/ micaz. pdf.

[11]. TmoteSky.URL:http://sentilla.com/files/pdf/eol/tmot e-sky-datasheet.pdf. Description: an introduction of Tmote sky.

[12]. Rene.URL: http://www.ruf.rice.edu/mobile/elec518 / lecture/ 2011 -chrispaul.pdf. Description: an overview of Rene mote.

[13]. Stargaze.URL:http://www.eol.ucar.edu/rtf/isa/intern al CrossBow /DataSheets/stargate .pdf

[14]. Survey of Hardware Systems for Wireless Sensor Networks. Mark Hempstead, Michael J. Lyons, David Brooks, and Gu-Yeon Wei, Journal of Low Power Electronics, Vol.4, 2008.

[15]. "Wireless multimedia sensor networks: A survey", Ian F. Akyildiz, Tommaso Melodia, Kaushik R. Chowdhury, IEEE Wireless Communications, December 2007.

[16]. Crossbow.URL: Inc., http://www.xbow.com/, Description: an overview of crossbow products.

[17]. Tmote Sky Datasheet. URL:http://www.sentilla.com/ pdf / eol/tmote-sky-datasheet.pdf.

[18]. CC2420.2.4GHzIEEE802.15.4/ZigBee-readyRF Transceiver. URL:http://inst.eecs.berkeley.edu/~cs150/Document s/CC2420.pdf.

[19]. Wireless Multimedia Sensor Networks: Current Trends and Future Directions", Islam T. Almalkawi, Manel Guerrero Zapata, Jamal N. Al-Karaki and Julian Morillo-Pozo, Sensors 2010.

[20]. A Survey of Simulators, Emulators and Testbeds for Wireless Sensor Networks", Muhammad Imran, Abas Md Said, Halabi Hasbullah, Information Techonology (ITSim), International Symposium in, June 2010.

[21]. Art of Computer Systems Performance Analysis Techniques For Experimental Design Measurements Simulation And Modeling. Wiley Computer Publishing, John Wiley & Sons, Inc, 1991.

[22]. Discrete_event_simulation.URL:http://en.wikipedia. org/ wiki /Discrete_event_simulation, Description: an introduction of discrete-event-simulation in wiki webpage.

[23]. Simulation Tools for Wireless Sensor Networks. E. Egea-Lopez, J. Vales-Alonso, A. S. Martinez-Sala, P. Pavon-Marino.J.Garcia-Haro, Summer Simulation Multiconference, SPECTS, 2005.

[24]. Wireless Sensor Network Simulators A Survey and Comparisons. Harsh Sundani, Haoyue Li, Vijay K. Devabhaktuni, Mansoor Alam, & Prabir Bhattacharya, International Journal Of Computer Networks (IJCN), Vol-2 : Issue-5.

[25]. NS-2.URL: http://www.isi.edu/nsnam/ns/, Description: a webpage introduced NS-2.

[26]. Performance Evaluation of NS-2 Simulator for Wireless Sensor Networks. Yunjiao Xue, Ho Sung Lee, Ming Yang, Kumarawadu, P., Ghenniwa, H.H., Weiming Shen, Electrical and Computer Engineering, CCECE Canadian Conference on, 22-26 April 2007.

[27]. TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications. Philip Levis, Nelson Lee, Matt Welsh, David Culler, SenSys, 2003.
[28]. TOSSIM.URL:http://docs.tinyos.net/index.php/TOSSIM, Description: a webpage introduced TOSSIM.
[29]. Omnet++.URL:http://en.wikipedia.org/wiki/Omnet%2B %2 B, Description: an introduction of Omnet++ in wiki webpage.
[30]. EmStar: a Software Environment for Developing and Deploying Wireless Sensor Networks", Lewis Girod, Jeremy Elson, Alberto Cerpa, Thanos Stathopoulos, Nithya Ramanathan, Deborah Estrin, USENIX Technical Conference, 2004.
[31]. J-sim.URL:http://sites.google.com/site/Description:a webpage introduced J-sim.
[32]. GloMoSim: A Library for Parallel Simulation of Large-scale Wireless Networks", Xiang Zeng, Mario Gerla.
[33]. ATEMU.URL:http://www.hynet.umd.edu/research/atemu/ Description: a webpage introduced ATEMU.
[34]. ATEMU: A Fine-grained Sensor Network Simulator",J. Polley, D. Blazakis, J. McGee , D. Rusk, J.S. Baras, First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, Santa Clara, CA, October 4-7, 2004.