IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 5, May 2016

An Overview on Wireless Sensor Networks Technology and Simulation Software's

Vijay S. Kale[1], Rohit D. Kulkarni[2]

Associate Professor, Department of Electronic Science, KTHM College, Nashik, Maharashtra, India[1]

M. Phil. Student, Department of Electronic Science, KTHM College, Nashik, Maharashtra, India[2]

Abstract: Advances in wireless networking, micro-fabrication and integration of sensors and actuators manufactured using micro-electromechanical system (MEMS) technology and embedded microprocessors have enabled a new generation of massive-scale sensor networks suitable for a range of commercial and military applications. Sensor networks promise to couple end users directly to sensor measurements and provide information that is precisely localized in time and/or space, according to the user's needs or demands. Node-level design simulators simulate the behaviour of a sensor network on a per-node basis. Using simulation, designers can quickly study the performance in terms of timing, power, bandwidth, and scalability without implementing them on actual hardware and dealing with the actual physical phenomena. The present communication gives the brief survey of wireless sensor network (WSN), WSN components, details of operating systems (TinyOS, RIOT, Nano-RK, MANTIS etc.) and WSN simulation software's (NS2, OMNET++, J-Sim, JiST, GloMoSim, SSFNet etc.). The aim is to provide a better understanding of the current research issues in this field and outline the use of certain tools to meet the design objectives which will be useful to learner/researcher to develop WSNs applications.

Keywords: Sensor nodes, Architecture, Operating system, Simulation software, Wireless technology.

I. INTRODUCTION

The term wireless communication was first introduced in 19th century. In 1895 Marconi demonstrated the first radio transmission from Isle of Wright to Tugboat 18 miles away. In this technology the information can be transmitted through air without any wires or electronic conductors by electromagnetic waves. Presently wireless communication technology finds their applications in sensing, monitoring, smart phones, laptops, Bluetooth, technology and in networking.

Sensor networks extend the existing Internet deep into the physical environment. The resulting new network is more expansive and dynamic than the current TCP/IP network and is creating entirely new types of traffic that are quite different from what one finds on the Internet now. Information collected by and transmitted on a sensor network describes conditions of physical environments for example, temperature, humidity, or vibration and requires advanced query interfaces and search engines to effectively support user-level functions.

Networked sensing offers many advantages over traditional centralized approaches. Dense networks of distributed communicating sensors can improve signal-to-noise ratio (SNR) by reducing average distances from sensor to source of signal, or target [1]. Increased energy efficiency in communications is enabled by the multi-hop topology of the network. Moreover, additional relevant information from other sensors can be aggregated during this multi-hop transmission through in-network processing. The greatest advantages of networked sensing are in improved robustness and scalability. A decentralized sensing system is more robust against individual sensor node or link failures. Decentralized algorithms are also far more scalable in practical and may be the only way to achieve the large scales needed for some applications [2].

WIRELESS SENSOR NETWORK ARCHITECTURE

Wireless sensor network (WSN) or wireless sensor & actuator network (WSAN) are spatially distributed sensors to monitor physical or environmental conditions such as temperature, humidity, fire etc. and to cooperatively pass their data through the network to the main location. WSN consist of three main components nodes, gateways and the software. The sensors measure the parameter of interest & transmit their data wirelessly through the gateway to the host system where the software collects the data, processes it so that it can be analysed [2, 3].

A. Sensor node or Mote

The main components of the WSN sensor node is radio modem, controller, sensor and power supply. The block diagram of sensor node is shown in Fig.1.

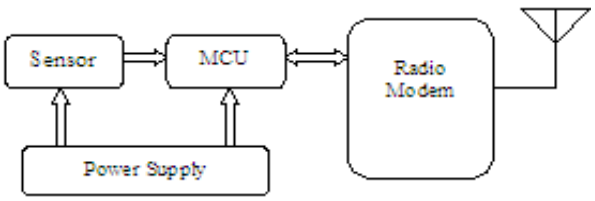

Fig.1. Block diagram of sensor node

1) Radio modem:

The work of the radio modem is to transmit data wirelessly across a range of tens of thousands of meters.

DOI 10.17148/IJARCCE.2016.55206

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 5, May 2016

They often make use of ISM band which gives free radio spectrum allocation and global availability. WSN uses three licence free communication frequencies bands 868MHz, 915MHz and 2.4GHz.

2) Controller:
The controller performs tasks, processes data and controls the functionality of other components in the sensor node. The analogue signals produced from the sensor are converted into digital signals by the analogue to digital converter (ADC) and fed to the processing unit.

3) Sensors:
They are the hardware devices that produce a measurable response to a change in physical condition. They have small size and are typically large in number so as to measure maximum parameters.

4) Power supply:
The WSN are designed without the need of human intervention as they are placed in hard to reach location. The battery forms the heart of the sensor system as it decides the lifespan of the system. Hence battery plays an important role in ensuring that there is adequate energy available to power the system.

B. Gateway
The Gateway acts as a bridge between the WSN and the other network. The Gateway collects the information received from the motes in a database and makes this information available usually via a wireless network [3]. Sensor networks may internetwork with an IP core network via a number of gateways. A gateway routes user queries or commands to appropriate nodes in a sensor network. It also routes sensor data, at times aggregated and summarized, to users who have requested it or are expected to utilize the information.

C. Software
A typical Operating System (OS) abstracts the hardware platform by providing a set of services for applications, including file management, memory allocation, task scheduling, peripheral device drivers, and networking. For embedded systems, due to their highly specialized applications and limited resources, their operating systems make different trade-offs when providing these services. For example, if there is no file management requirement, then a file system is obviously not needed. If there is no dynamic memory allocation, then memory management can be simplified. If prioritization among tasks is critical, then a more elaborate priority scheduling mechanism may be added. TinyOS, MATE, a Virtual Machine for the Berkeley motes and TinyGALS are examples of node-level programming tools. Different available OS are [4, 5]

TinyOs: TinyOS have no file system, supports only static memory allocation, implements a simple task model, and provides minimal device and networking abstractions. Furthermore, it takes a language-based application development approach, so that only the necessary parts of the operating system are compiled with the application. To a certain extent, each TinyOS application is built into the operating system. TinyOS organizes components into layers. Intuitively, the lower a layer is, the "closer" it is to the hardware; the higher a layer is, the "closer" it is to the application. In addition to the layers, TinyOS has unique component architecture and provides as a library a set of system software components. A component specification is independent of the component implementation. Although most components encapsulate software functionalities, some are just thin wrappers around hardware. The nesC language is used for programming. It is an extension of C to support and reflect the design of TinyOS v1.0 and above. It provides a set of language constructs and restrictions to implement TinyOS components and applications.

RIOT: It is a small operating system for networked, memory-constrained systems with a focus on low-power wireless Internet of Things (IoT) devices. It is based on microkernel architecture. RIOT allows application programming with C and C++ languages and provides multi-threading and real time abilities.
OpenTag: It is a DASH7 protocol stack & is designed to run on microcontrollers or a radio system on chip (SoC).It is written in C language. OpenTag is designed to be light and compact, as it is targeted to run on resource-constrained micro-controllers.
Nano-RK: "Nano" implies that the RTOS is small, consuming less power while "RK" is short for "resource kernel". It is an open source OS written in C and is a fixed, pre-emptive multitasking real time operating system designed for use in WSN.
LiteOs: It is open source, written in C and Unix-like operating system that's fit on memory constrained sensor nodes. LiteOS provides a familiar programming environment based on UNIX, threads and C. It follows a hybrid programming model allows both event-driven and thread-driven programming.
MANTIS: The MultimodAl system for NeTworks of In-situ wireless Sensors (MANTIS) provides a new multithreaded operating system for WSNs. It is an open source OS which is written in C. It provides automatic pre-emptive time slicing for fast prototyping.
Contiki: It is an event driven operating system. It makes the task of programming the sensor network closely resemble programming a PC. It is an open source OS. Contiki provides multitasking and a built-in Internet Protocol Suite (TCP/IP stack) with a full Graphical User Interface (GUI) features.

II. WSN SIMULATION SOFTWARE'S

Node-level design methodologies are usually associated with simulators that simulate the behaviour of a sensor network on a per-node basis. Using simulation, designers can quickly study the performance in terms of timing, power, bandwidth, and scalability of algorithms without implementing them on actual hardware and dealing with the actual physical phenomena. A node in a simulator acts as a software execution platform, a sensor host, as well as a communication terminal. In order for designers to focus

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 5, May 2016

on the application level code, a node model typically provides or simulates a communication protocol stack, sensor behaviours e.g., sensing noise and operating system services. If the nodes are mobile, then the positions and motion properties of the nodes need to be modelled. If energy characteristics are part of the design considerations, then the power consumption of the nodes needs to be modelled. Popular simulation software's are described below [6].

NS-2- The Network Simulator-2 (NS-2) is an open-source network simulator that was originally designed for wired, IP networks. Extensions have been made to simulate wireless/mobile networks (e.g., 802.11 MAC and TDMA MAC) and more recently sensor networks. While the original NS-2 only supports logical addresses for each node, the wireless/mobile extension of it introduces the notion of node locations and a simple wireless channel model. This is not a trivial extension, since once the nodes move, the simulator needs to check for each physical layer event whether the destination node is within the communication range. For a large network, this significantly slows down the simulation speed [7]. The main menu of NS-2 is shown in Fig.2.

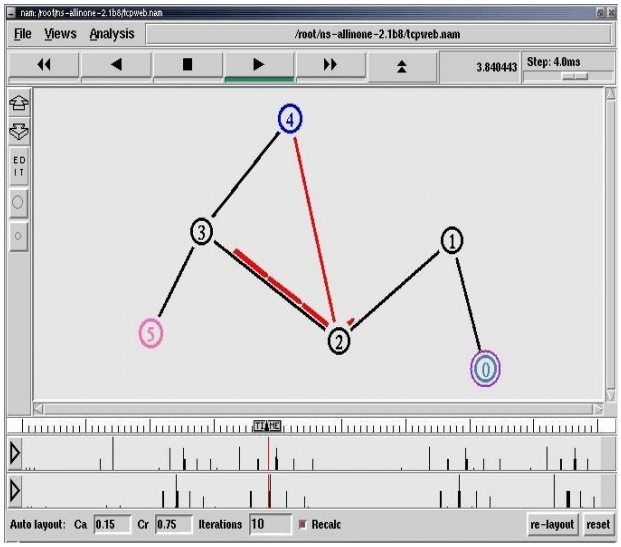

Fig.2. The main menu of NS2 simulator software

The main functionality of NS-2 is implemented in C++, while the dynamics of the simulation (e.g., time-dependent application characteristics) is controlled by Tcl scripts. Basic components in NS-2 are the layers in the protocol stack. They implement the handlers interface, indicating that they handle events. Events are communication packets that are passed between consecutive layers within one node, or between the same layers across nodes. The main advantage of NS-2 is its rich libraries of protocols for nearly all network layers and for many routing mechanisms.

OMNET++- It is a modular discrete event simulator implemented in C++. Getting started with it is quite simple, due to its clean design. OMNET++ also provides a powerful GUI library for 3-D virtualization, animation and tracing and debugging support. OMNeT++ has been used in numerous domains from queuing network simulations to wireless and ad-hoc network simulations, from business process simulation to peer-to-peer network, optical switch and storage area network simulations. OMNET++ opening menu is shown in Fig.3 [8].

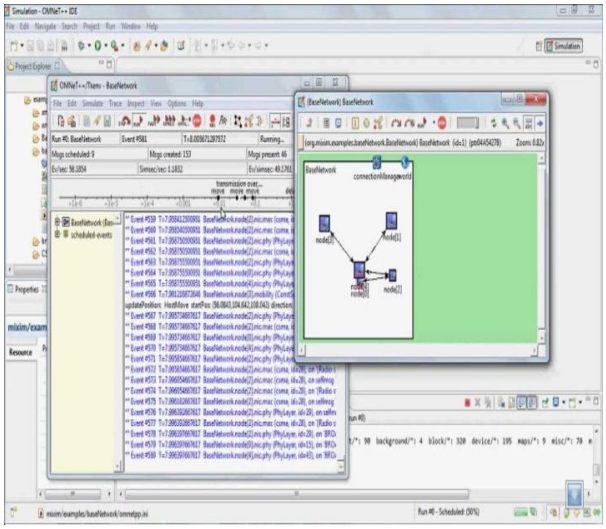

Fig.3. The opening menu of OMNET++

J-Sim- A component-based simulation environment developed entirely in Java. The main benefit of J-sim is its considerable list of supported protocols, including a WSN simulation framework with a very detailed model of WSNs and an implementation of localization, routing and data diffusion WSN algorithms [9]. J-Sim main menu is shown in Fig. 4.

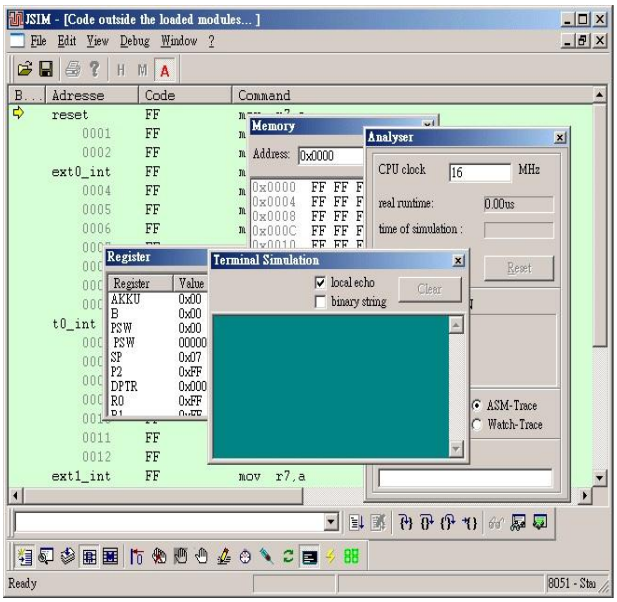

Fig.4. J-Sim main menu with different windows

NCTUns2.0- It is discrete event simulator whose engine is embedded in the kernel of a UNIX machine. The actual network layer packets are tunnelled through virtual interfaces that simulate lower layers and physical devices [10] and its screen shot is shown in Fig.5.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 5, May 2016

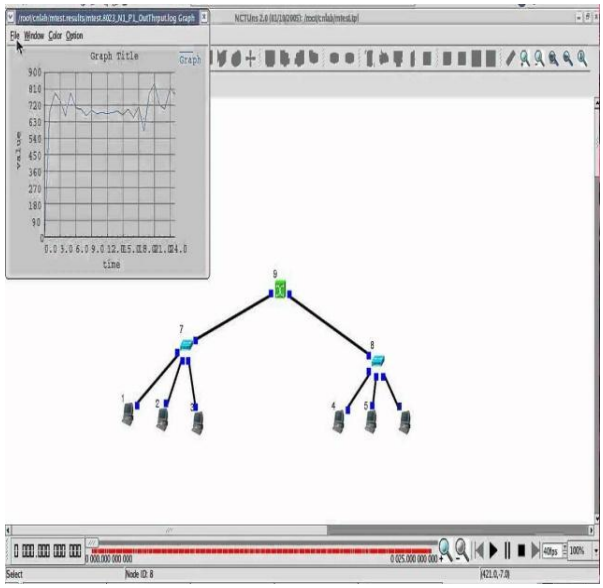

Fig.5. NCTUns2.0 simulator opening menu

JiST/SWANS- Java in Simulation Time (JiST) and Scalable Wireless Ad-hoc Network Simulator (SWANS) are discrete event simulation framework that embeds the simulation engine in the Java byte-code. Models are implemented in Java and compiled. Then the byte-codes are rewritten to introduce simulation semantics. Javis is a packet flow and network animator for JiST [11]. Javis simulator opening menu is shown in Fig.6.

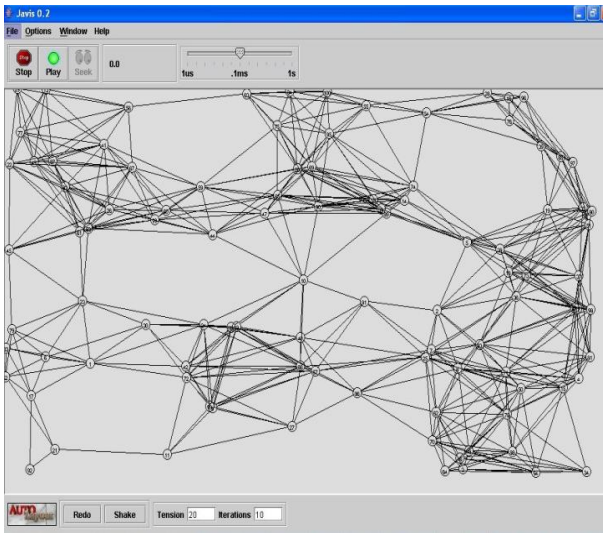

Fig.6. Jav is simulator opening menu

SSFNet- It is a set of Java network models built over the Scalable Simulation Framework (SSF) for modeling and simulation of Internet protocols and networks at and above the IP packet level. SSF is a specification of a common API for simulation that assures portability between compliant simulators. SSFNet models are self-configuring i.e. each SSFNet class instance can autonomously configure itself by querying a configuration database, which may be locally resident or available over the Web [12]. SSFNet simulator execution screen shot is shown in Fig.7.

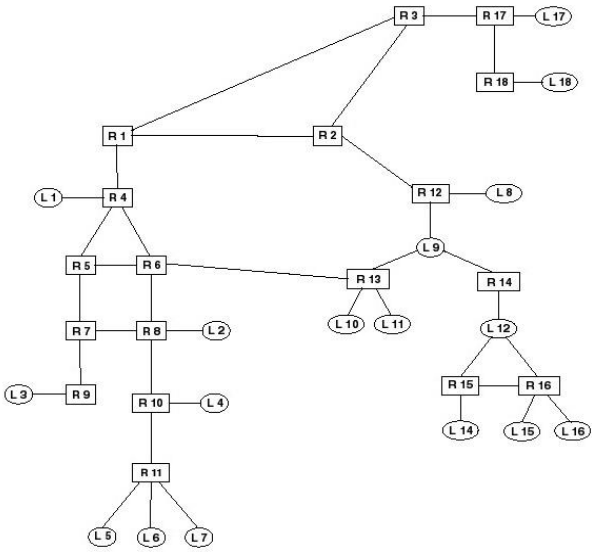

Fig.7. SSFet simulator Execution screen shot

GloMoSim- Simulation environment for wireless networks built with Parsec. Parsec is a simulation language derived from C that adds semantics for creating simulation entities and message communication on a variety of parallel architectures [13]. GloMoSim screen shot is shown in Fig.8.

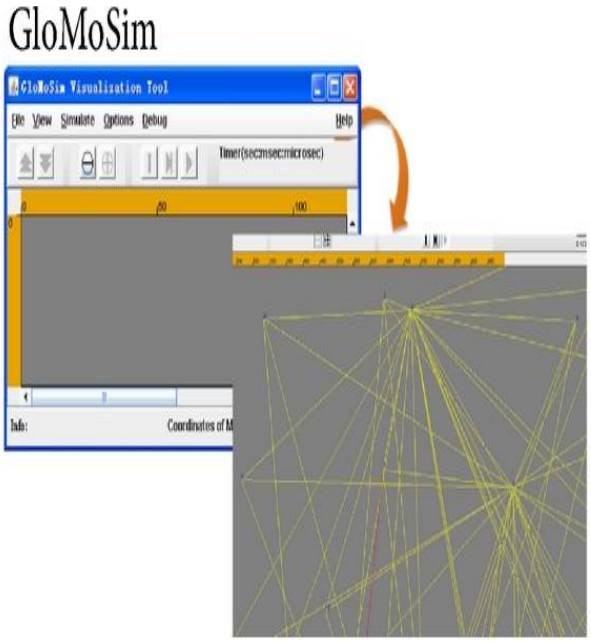

Fig.8. GloMoSim simulator opening menu

Ptolemy II- It contains Java packages that support different models of simulation paradigms (e.g. continuos time, dataflow, and discrete-event). It also addresses the modeling, simulation and design of concurrent, real-time, embedded systems. A major problem area being addressed is the use of heterogeneous mixtures of models of computation.The project is named after Claudius Ptolemaeus, the 2nd century Greek astronomer, mathematician, and geographer [14]. The opening menu of Ptolemy II is shown in Fig.9.

DOI 10.17148/IJARCCE.2016.55206

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 5, May 2016

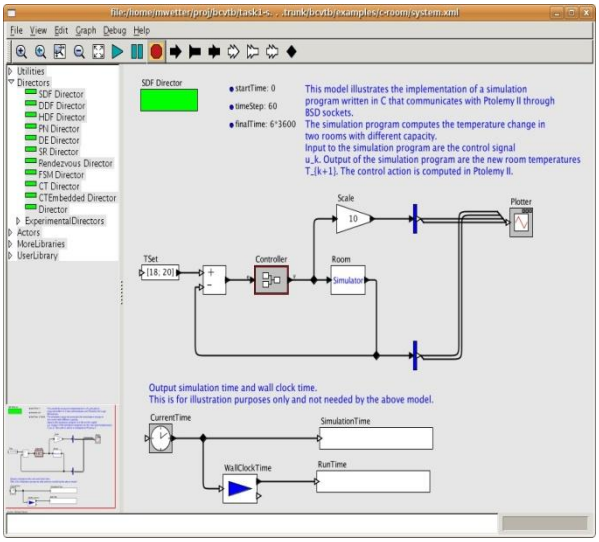

Fig.9. Opening menu of Ptolemy II simulator

III. DIFFERENT WIRELESS TECHNOLOGIES

To Use wireless technology for different applications, the Institute of Electrical and Electronic Engineers (IEEE) has proposed standards and protocols according to its need. Table 1 list the comparison of different standards [15].

Table1. Comparison of different communication standards

Standard	Bluetooth	Infrared	Wi-Fi	ZigBee
Governing body	Bluetooth SIG	Infrared Data Association	Wifi Alliance	Zigbee Alliance
IEEE Specification	802.15.1	802.11	802.11 a/b/g	802.15.4
Frequency Band	2.4 GHz	875 nm+-	2.4GHz, 5 MHz	868/915 MHz, 2.4 GHz
Standard range	1-100m	0.2-1 m	100 m	10-100m
Power Consumption (Days)	1 to 7	<200	1 to 5	100 to > 1000
Number of RF channels	79	50	14	1/10,16
Data transfer rate	3Mbits/s	4Mbits/s	54Mbits/s	250Kbits/s
Max no. Of nodes	8	2	2007	>65000
Modulation Type	GFSK	Pulse	BPSK, QPSK, M-QAM, CCM	BPSK(+ASk), O-QPSK
Applications	Cable replacement	Cable replacement	Web,E-mail video	Monitering & control

IV. APPLICATIONS

WSN are used in various applications. The list of different areas and the possible parameters measurement are given below [16].

A. Military or Border surveillance applications: Examples are Vehicle detection, Weapons, Chemical sensing etc.
B. Environmental Applications: Examples are tracking the movements and patterns of insects, birds or small animals etc.
C. Healthcare applications: Examples are Blood pressure, Heart beat, Stress, Body temperature, Sleep, Brain activities etc.
D. Home Intelligence: Examples are HVAC, Lighting controls system, Security, Gas leak detection, Energy saving etc.
E. Automobile Industry: Examples are Acceleration, Fuel consumption, Tire pressure, acknowledgment of illumination failures (turn lights, brake lights, front lights, and register plate lights) etc.
F. Precision Agriculture: Examples are Water level, Temperature, Humidity, Soil Moisture, pH level, Wind flow, Light Intensity etc.
G. Structural Monitoring: Examples are Wind and Weather, Traffic, Deck, Pylons, Ground, Prestressing, Stray cables etc.
H. Industrial Process Control: Examples are Temperature, Steam pressure, Liquid levels, Flow, Viscosity etc.
I. Environmental Conditions Monitoring: Examples are Earthquakes Volcano, Tsunami, Fire, Flood, Pollution etc.
J. Oil and Gas industry: Examples are Oil bunkering and theft, pipeline vandalization etc.

V. CONCLUSION

Advances in wireless networking, and integration of sensors and actuators manufactured using MEMS technology and embedded microprocessors have enabled a new generation of massive-scale sensor networks suitable for a range of commercial and military applications. Sensor networks promise to couple end users directly to sensor measurements and provide information that is precisely localized in time and/or space as per user's demands. There are several different types of OS for WSNs, all of which have different aspects, pros and cons. WSN OS are usually less complex because of the scarcity of the resources. The first OS created for WSNs is TinyOS. The main difference with this and other operating systems is that TinyOS is event driven programming while others tend to multithreading.

Node-level design simulators simulate the behaviour of a sensor network on a per-node basis. Using simulation, designers can quickly study the performance in terms of timing, power, bandwidth, and scalability without implementing them on actual hardware and dealing with the actual physical phenomena. The overview provides guidelines to help selecting a suitable simulation model for a WSN and a comprehensive description of the most used available tools. All the packages provide graphical user

DOI 10.17148/IJARCCE.2016.55206

interface. The OMNET++, NCTUns2.0, J-Sim and Ptolemy II provide powerful GUI libraries for animation, tracing and debugging. Parallel simulations should perform and scale better than sequential ones. Modeling problems arise when considering the new environment and the energy components. They also compromise scalability and accuracy. A deep study of these issues is mandatory for a better understanding and characterization of sensor networks and their corresponding simulators. ZigBee standard from ZigBee Alliance suites for the implementation of WSN which demands high reliability, low cost, low power etc. for monitoring and control applications.

The various application areas of WSN are military, environmental applications, healthcare, home intelligence, automobile industry, precision agriculture, structural monitoring, industrial process control, environmental condition monitoring, oil and gas industry etc.

ACKNOWLEDGMENT

This research is supported by the Principal **Dr. Dilip Dhondge** and Head, Department of Electronic Science, **Dr. M. B. Matsagar** of KTHM College, Nashik, Maharashtra, India.

REFERENCES

[1] Wireless Sensor Networks: An Information Processing Approach By Feng Zhao, Leonidas Guibas.
[2] Wireless Sensor Networks: Concepts and Components (© SpringerInternational Publishing Switzerland 2014 5 J. Cecílio, P. Furtado, Wireless Sensors in Heterogeneous Networked Systems, Computer Communications and Networks, DOI 10.1007/978-3-319-09280-5_2.
[3] On the design of a flexible gateway for Wireless Sensor Networks Marco ZENNARO, Hervé NTAREME and Antoine Bagula.
[4] Operating Systems for Wireless Sensor Networks: A Survey Technical Report Adi Mallikarjuna Reddy V AVU Phani Kumar, D Janakiram, and G Ashok Kumar
[5] Operating Systems for Wireless Sensor Networks: A Survey M.O.Farooq and Thomas Kunz Sensors (Basel). 2011; 11(6): 5900–5930.publish online 2011 May31 doi: 10.3390/s110605900
[6] Simulation Tools for Wireless Sensor Networks E. Egea-López, J. Vales-Alonso, A. S. Martínez-Sala, P. Pavón-Mariño, J. García-Haro Summer Simulation Multiconference - SPECTS 2005
[7] The Network Simulator, NS–2 [Online] http://www.isi.edu/nsnam/ns/
[8] OMNET++ discrete event simulator. [Online]. Available: http://www.omnetpp.org
[9] J-Sim [Online]. Available: http://www.j-sim.org
[10] NCTUns 2.0 Network Simulator and Emulator. [Online]. Available: http://nsl.csie.nctu.edu.tw/nctuns.html
[11] R. Barr, Z. J. Haas, R. van Renesse, "JiST: Embedding Simulation Time into a Virtual Machine." In Proc. 5th EUROSIM Congress on Modeling and Simulation, Paris, France, September 2004
[12] Scalable Simulation Framework (SSF). [Online]. Available: http://www.ssfnet.org
[13] Global Mobile Information Systems Simulation Library (GloMoSim).[Online].Available: http://pcl.cs.ucla.edu/projects/glomosim/
[14] Ptolemy II. Heterogeneous model and design. [Online]. Available: http://ptolemy.eecs.berkeley.edu/ptolemyII Weid, P. Oberson, and N. Gisin, "High resolution fiber distributed measurements with coherent OFDR," in Proc. ECOC'00, 2000, paper 11.3.4, p. 109.
[15] Comparative Analysis of different wireless Techologies, Neeraj Chhabra International Journal of Scientific Research in Network Security & Communication.Vol-1, Issue-5, ISSN: 2321-3256
[16] WSN Practical adaptations Ritika Sobti, Neha Bassi, International Journal of Engineering and Computer Science ISSN: 2319-7242 Volume 4 Issue 9 Sep 2015, Page No. 14443-14448

BIOGRAPHIES

Vijay Kale (M.Sc, M.Phil, Ph.D, PGDIM, and ADCSSA) is working as an Associate Professor (Department of Electronic science, KTHM College, Nashik, Maharashtra, India). He has been in the teaching profession (UG and PG) since last 27 years. He has been presented research paper in international conferences (USA, Bangkok). He published research papers in national and international journals. He received R. Chandrasekhar award from Indian Physics Association (IPA), Savitribai Phule Pune University. He has written five books. He worked as project guide for M.Sc. (Electronic Science) and research guide to M. Phil. students. He has worked on several academic committees of Savitribai Phule Pune University. He has worked as a resource person in refresher course, workshop etc. He is presently working on ARM microcontroller based sensor application, Wireless sensor application, e-CALLISTO etc.

Rohit D. Kulkarni (M.Sc, NET) is an M.Phil student in the Department of Electronic science, KTHM College, Nashik, Maharashtra, India. He has two years Industry experience and over six years of teaching experience. His active interest areas are wireless sensor networks, simulation software's, and embedded systems applications using sensors.

DOI 10.17148/IJARCCE.2016.55206