International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-3, September 2019

Experimental Tools and Techniques for Wireless Sensor Networks

Divya Pandey, Vandana Kushwaha

Abstract: ***Wireless Sensor Networks (WSNs) have earned an upsurging inquisitiveness of the research community over the past decade and for the upcoming era, due to which we can realize its countless applications in various fields including medical, military, environment and academia. Wireless sensor networks composed of thousands or sometimes millions of sensor nodes which are usually operated by batteries that have limited energy, data processing, storage and transmission capacity. Moreover, these sensor nodes are densely deployed in human inaccessible regions and connected through error prone wireless links to offer a long coverage, therefore It is very difficult rather impractical to analyze the behavior of WSNs with real experiments. Most researches in this particular field are carried out by using simulation tools in several educational associations and research hubs. For this reason study of existing experimental tools for analyzing the behavior of WSNs becomes essential. This paper is intended to present a review on experimental tools and techniques available in the literature for WSNs, including simulators, emulators and test beds. These tools are highly useful for analyzing the operational behavior of wireless sensor networks. This paper also provides a roadmap for selecting appropriate experimental tools for various kinds of WSNs and its applications by taking into account their capabilities, relative ease of use and accuracy.***

Index Terms: Emulation, Experimental Tools, Simulation, Testbed, Wireless Sensor Networks,

I. INTRODUCTION

In the present period of data and communication automation, Wireless sensor Networks (WSNs) [1] emerged as the most encouraging exploration territory. WSNs have procured broad thought from both scholarly and mechanical space over the globe. WSNs contains spatially disseminated self-organizable, self-ruling and autonomous sensing gadgets. Sensor nodes are generally operated by battery so they have constrained vitality that can be effectively depleted if not utilized in a productive manner. These nodes are embedded with at least one sensors, microchips and radio handsets in this manner they have sensing capacity as well as processing and communication capacity. Besides, they are conveyed in human unattainable regions where it isn't constantly conceivable to revive or supplant batteries, so hubs may come up short, move or join because of energy exhaustion, channel blurring, as well as environmental interference. These occasions lead to node density and topology changes. Subsequently, WSNs require self-organizable, self-configurable and fault tolerant nodes. The dynamic attributes, unique constraints, and prerequisites of WSNs entails the need of procuring test devices and procedures that are utilized to test and analyze the behavior of WSNs before and after physical implementation. With the utilization of Experimental apparatuses, the clever thoughts of specialists can be tried in a genuine situation which keeps away from inefficient and uneconomical equipment executions. [2]

This paper is organized as follows: In section 2 related works in the literature have been explored. In Subsequent Sections the overview of experimental tools has been presented in which section 3 talks about various simulators used for WSNs and its necessity and limitations. Similarly in section 4 Emulators and their necessity and limitations have also been demonstrated likewise in section 5 testbeds and their necessity and limitations have been elaborated. In section 6 we have outlined unique features of most intensively used simulation, emulation and test beds tools in table form. Section 7 includes wide discussion on above-listed tools for WSNs, Finally, Section 8 concludes the paper.

II. RELATED WORK

In the literature, Experimental devices and systems have been studied broadly with regards to wireless sensor networks. Some noteworthy study works identified with relevance to topic are referenced underneath. Korkalainen et al. (2009) [3] have presented five commonly used simulation tools for wireless sensor networks that include NS-2, OMNeT++, Prowler, OPNET, and TOSSIM. They further mentioned that in spite of the fact that there are numerous test systems created for a particular demonstrating task in which they are precisely suitable still these tools should be broadened or altered for an increasingly exact reenactment of sensor systems. Khan et al. (2011) [4] have reviewed some of the most widely-used and state-of-the-art simulation tools for WSNs, the simulation tools presented are NS-2, OMNeT++, GloMoSim, OPNET, SENSE, TOSSIM, and GTSNetS, and further, they distinguished the key constraints of the surveyed simulation tools.

Revised Manuscript Received on 30 September, 2019
* Correspondence Author
Divya Pandey, Computer Science Department, Banaras Hindu University, Varanasi, India.
Vandana Kushwaha, Computer Science Department, Banaras Hindu University, Varanasi, India.

They also analyzed their practicality for large scale WSNs. Fei Yu (2011) [5] explored some most popular simulators: NS-2, TOSSIM, EmStar, OMNeT++, J-Sim, ATEMU and Avrora, and provided comparison among them along with their merits and limitations. He also explained use and necessity of simulation, and considerable features while simulating WSNs. Al-Fayez (2012) [6] presented a comprehensive survey of widely used simulation tools in the field of WSNs. However, the author believes that these tools are not sufficient for the testing and evaluating purpose because in some cases simulation results show unrealistic values and setting up large-scale real testbeds require lots of effort, time and cost. So he suggested applying mix- mode solution to researchers. Mrs. Chhimwal et al. (2013)[7] have selected seven simulators and explained the pros and cons of these simulators when it comes to simulating WSNs. Chand et al. (2013)[8] have introduced, a suitable model for WSN simulation, highlighting features and criteria for choosing an appropriate simulation tool, model and framework. They discussed the most relevant simulation tools used to study WSN (NS-3, OPNET, GloMoSim, Mixim, Castalia, J-sim, and Avrora). Their features and implementation issues have also been illustrated. Rao et al. (2013) [9] presented advantages, disadvantages, and features of various simulators (NS-2, NS-3, OMNET++, TinyOS, J-Sim, OPNET, NetSim, SimPy and QualNet). A.K Dwivedi et al. (2011) provided an exploratory study of experimental tools for wireless sensor networks. He presented a review of nearly 63 simulators 14 emulators and other frameworks like visualization tools, debugging tools, network monitors. This work is extension to his work, although they have given a review of vast variety of experimental tools but did not give comparative analysis of these tools. Comparison would be valuable so as to enable researchers to pick the best test system for particular application environment.

The above research contributions demonstrate, in spite of the scientists have tried their endeavors to exhibit a survey of test devices for WSNs, however, we observed that in all inspected paper analysts talked about simulators only or used simulator and emulator term interchangeably. There is not any ongoing recent systematic comprehensive review of tools which separately describe simulators emulator and test beds, their necessity, and limitations. Review papers that can recognize research gaps, front line issues and present innovative ideas in this subject of experimental tools for WSNs are overwhelmingly very less in number. Unlike previous research articles, this survey article is gone for giving a fresh clarification of experimental tools simulator, emulator and testbeds widely used in WSNs.

III. EXPERIMENTAL TOOLS: AN OVERVIEW

When innovative ideas begin to grow in the researcher's brain, to turn it into fully-fledged solutions and in further strides to approve and check their perfection and deployment feasibility, requires a particular arrangement of trial apparatuses to work with. System exploratory instruments are frequently required in the planning stage before genuine execution. There is a wide collection of research and development tools available with divergent capabilities for WSNs. Contrasting to wired and traditional wireless networks, WSNs have certain prerequisites and requirements, which must be considered while picking these devices [10]. Knowledge of the merits, demerits of different experimental tools, nature and requirements of the network under consideration is important as based on this knowledge users can opt the most appropriate tool for their use. In this section, a plausible explanation of Wireless Sensor Network research and development tools is presented. Along with that, a revised taxonomy has been proposed regarding experimental tools for WSNs, shown in Figure 1. We have classified experimental tools for WSNs in four categories i.e., Simulators, Emulators, Test Beds, and other tools. Simulators are further categorized based on their features, whether they have been designed for the general purpose (applicable to other networks also) or they are specifically designed for WSNs only. We have also summarized the observations of reviewed tools and techniques used for Wireless Sensor Networks in the form of tables.

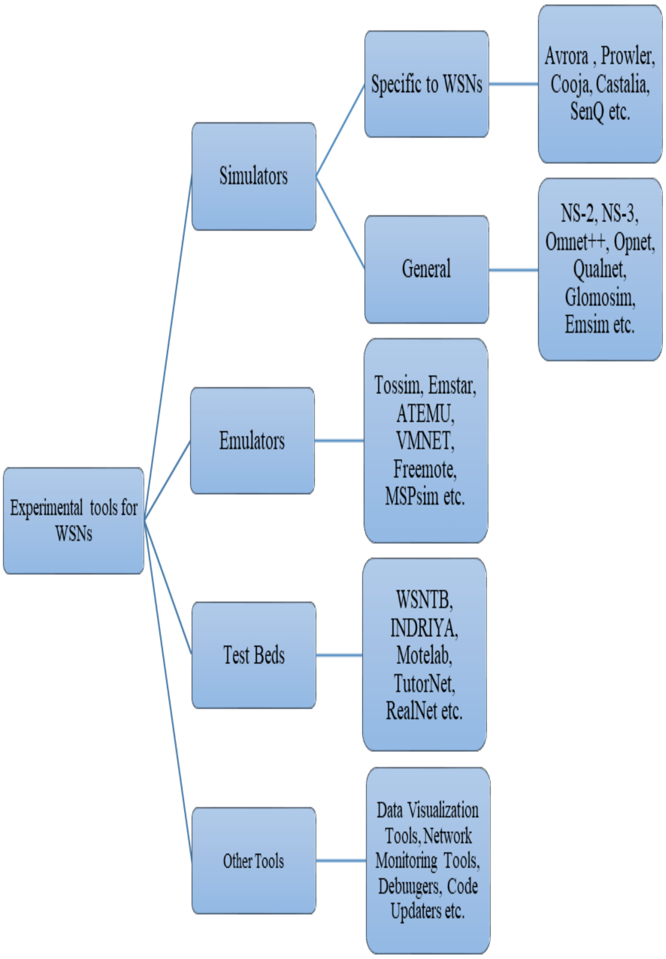

Fig 1. Experimental tools for WSNs

A. SIMULATION

Simulation plays a vital role in developing and overseeing embedded systems and/or network centric systems, throughout their life cycle. With the help of simulation detailed inspection of the networks and related features, their dynamic behavior and executions can be done even before the software is actually developed onto the target hardware.

Retrieval Number: C4441098319/19©BEIESP
DOI:10.35940/ijrte.C4441.098319
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Simulation works by creating conditions that are quite similar to actual conditions, then the training is provided under those conditions. A Simulator is commonly used as a software framework that predicts the behavior of computer networks by imitating few components of the real world. Various elements of the real-world system are mirrored and modeled depending on the intended function of the simulator [11]. To mimic a specific system, the initial step is to build up a model that speaks to the key qualities, practices, and elements of the chose physical or theoretical framework or procedure. So model imitates the system itself, whereas simulation performs the operation of the system over time. Simulators are an exercise tool to examine and gain new insights into new technology and to assess the performance of systems too complex for analytical solutions. In short, it can be said that simulation is a very efficient and economical approach to experiment with protocols and technique. Descriptions of some popularly used simulators in WSNs are listed below:

NS-2(Network Simulator version -2) [12]

NS-2 is an object oriented, open-source test system and it goes under the class of discrete occasion test system. NS had been created as a variation of the REAL system test system in 1989. NS-2 is an extremely incredible simulator for TCP, routing, and multicast conventions over wired and remote (neighborhood and satellite) systems. The Programming language used in NS-2 is C++ and an Object-situated form of Tcl called Otcl. Standard network protocols, for example, TCP and UPD, traffic source conduct, for example, FTP, Telnet, Web, CBR and VBR, router queue management, for example, Drop Tail, RED, and CBQ, routing calculations can be actualized using NS-2. The NS-2 recreation condition offers extraordinarily adaptable expansions for WSNs that is the reason it is widely being utilized for analyzing the attributes of WSNs. NS-2 works well on Linux Operating Systems or on Cygwin, which is a Unix-like environment and command-line interface running on Windows.

NS-3 (Network Simulator version-3) [13]

NS-3, a discrete-event network simulator for Internet frameworks was first created in 2006. It was structured fundamentally for research and instructive use. NS-3 is free software, authorized under the GNU GPLv2 permit, and is openly accessible for research and developmental use. With the help of NS3 simulator, we can create different virtual hubs (i.e., PCs in real life) and with the assistance of different helper classes, it enables us to install devices, web stacks, application, and so forth to these associated hubs. Ns3 gives us uncommon highlights which can be used for real-life integrations. A portion of these highlights are: tracing of nodes which enables us to follow the routes of the nodes which helps us to track how much information is sent or got, Network Animator: it is an animated form of how system will look in reality and how information will be moved from one node to another, Pcap file: which can be utilized to get all data of the packets (e.g., Sequence number, Source IP, destination IP, and so on), GNU plot: is used to plot diagrams and graphs from the information which we get from trace record of NS3.

OMNET++ [14]

OMNeT++ is a scalable, modular, segment-based, open source, object oriented simulation library and framework, principally used for building network simulators. Despite the fact that OMNeT++ isn't a network simulation system itself, it has increased across the board fame as a network simulation platform in researchers just as in mechanical settings, and working up a huge client network. OMNeT++ offers an Eclipse-based IDE, a graphical runtime condition, and a large group of different instruments. There are augmentations for constant stimulation, network emulation, database combination, SystemC reconciliation, and a few different OMNeT++ is disseminated under the Academic Public License. Omnet++ runs basically on all platforms whether it be Windows, Linux, or macOS.

J-SIM [15]

J-Sim is an open source, java based object-oriented discrete event simulator. J-sim was basically designed for queueing network simulation, however, J-sim can be used to simulate large scale system where object states change discretely. J-Sim provides facilities same as Simula simulation environment. J-Sim is purely Java written software and therefore completely portable. J-Sim has a GUI library, which encourages clients to display or arrange the Mathematical Modeling Language, a "content-based language" kept in touch with J-Sim models. J-sim has been used mostly in physiology and biomedicine field however, it can likewise be utilized in WSN simulation. J-Sim has been tested on Solaris 8, OS/2 Wrap 4, and eComstion 1.1, Windows NT 4, Windows XP and Linux.

AVRORA [16]

AVRORA an exploration task of the UCLA Compilers Group is a set of simulation and a framework for programs written for AVR microcontroller produced by Atmel and the Mica2 sensor nodes. Avrora provides flexibility for simulating and analyzing assembly programs, providing a clean Java API and infrastructure for experimentation, profiling, and analysis. This simulator is a combination of TOSSIM and ATEMU. Avrora also underpins energy consumption simulation. This test system gives open sources and online records. Avrora can simulate networks with thousands of node in lesser time and with much accuracy. If we talk about limitations of AVRORA, it does not support graphical user interface. In addition, as AVRORA has not any network communication tools so it cannot simulate network management algorithms.

OPNET [17]

OPNET is a short form of Optimized Network Engineering Tools. OPNET provides relatively much powerful visual or graphical displaying approach (Graphical user interface). OPNET is a specific business programming supplier for network and innovative work. It very well may be utilized to analyze communication systems, gadgets, conventions, and different applications we can see all the topology setup and simulation results.

The parameters can likewise be adjusted and the experiments can be rehashed effectively through simple procedure through the GUI. The simulation results and information can be broke down and showed all-around effectively. User-friendly diagrams, graphs, measurements, and network animation can be produced by OPNET for user convenience.

QUALNET [18]

QUALNET is a discrete-event simulator. It is a commercial version of GloMoSim used by Scalable Network Technologies (SNT) for their defense projects namely JTRS Network Emulator, BCNIS and Stratcom Cyber. Qualnet can be viewed as an extended version of GLOMOSIM with an advanced set of models and protocols. Qualnet software enables user to design advanced wireless modules, creating and animating network scenario (network topology) and user-friendly tools for analyzing simulation results. Qualnet has a modular design and an intuitive GUI that makes it user friendly. Key features of QUALNET is speed, scalability, model fidelity, portability and extensibility. Supported platforms for QUALNET are Microsoft windows and Linux (64-bit editions only).

GLOMOSIM [19]

GLOMOSIM stands for global mobile information system simulator and satellite network simulation setting for large and wireline communication network, adhoc mobile network set up at ucLA's paraell computing lab. GLOMOSIM is a scalable, discrete parallel event simulator that provides complex systems with a parallel simulation setting. GLOMOSIM supports two models of node mobility; node moving based on the model referred to as the model of random way point. Other GLOMOSIM mobility model referred to as a drunken random model. In 2000, GloMoSim stopped updating and published its commercial version called QualNet.

PROWLER [20]

PROWLER is a probabilistic wireless network simulator intended specifically to simulate wireless distributed systems on a big scale. Prowler, need MATLAB assistance that offers an easy way to prototype implementation and has good visualization skills. Originally designed to simulate Berkeley MICA motes, Prowler is now available for more particular platforms as well. Prowler is a wireless network simulator that is driven by events. Prowler can simulate transmission of radio, propagation, reception including collisions in ad hoc radio networks, and MAC-layer operation. Prowler works exceptionally fine for communication and routing protocol development, debugging, optimization, parameter tuning, and arbitrary application prototyping.

EMSIM [21]

EMSIM is a power simulation framework that simulates a straightforward integrated system with a powerful ARM microprocessor and Linux OS. This simulation framework is designed to promote research in the following fields: 1) to study the impacts of an OS on the general power usage of the system. 2) Characterize and macro-model Linux OS power usage 3) study the impacts of the architectural conversion of OS-related software.

COOJA [22]

COOJA is a versatile simulator based on Java intended to simulate sensor networks. COOJA is very helpful in the growth of distributed algorithms. In particular, the research community utilizes it widely to simulate small to comparatively big wireless networks of linked devices embedding sensors and/or actuators, frequently referred to as "motes," and thus to create, debug and assess WSN-based projects. Cooja also offers a graphical user interface (GUI) to design, operate, and evaluate WSN simulations based on Java's normal Swing toolkit.

Mannasim [23]

Mannasim is a simulation environment for Wireless Sensor Networks consisting of two alternatives: the Mannasim Framework and the Script Generator Tool. The Mannasim Framework is a Network Simulator (NS-2)-based WSN simulation module. Mannasim is an expanded NS-2 version with several fresh modules capable of designing, developing and analyzing multiple WSN apps. A front-end for simple development of TCL simulation scripts is the Script Generator Tool (SGT). SGT is mixed with Mannasim Framework and written in pure Java, making it an autonomous platform.

Castalia [11]

Castalia is an open source, OMNet++ Extension for WSNs. Castalia can simulate low-power embedded systems networks such as WSNs, body area networks and adhoc mobile networks. Castalia simulator is a useful choice to test distributed algorithms and/or protocols in realistic wireless channel and radio models, with a realistic node conduct particularly related to radio access. As it is extremely parametric and can simulate a broad variety of platforms, Castalia can also be used to assess distinct platform features for particular apps. Castalia's recent version is 3.3. Castalia's key characteristics include: physical process modeling, bias and noise sensing device, node clock drift, multiple MAC and built-in routing protocols.

B. NECESSITY OF SIMULATION

Widespread research on WSNs has raised the demand for simulation tools and frameworks because it makes the whole process easier, cheaper and faster. Simulation is being used universally all around in light of its accessibility, versatility and re-convenience features. Simulators are also a very useful tool for comparing new techniques to existing ones, by virtue of statistics obtained from simulators that are realistic, convenient and economical. Simulator designed for WSNs enables its client to seclude a few factors by tweaking configurable parameters [5]. It supports scripting languages of rich-semantics to define experiments and process results. Protocols, plans, even new thoughts can be assessed on a huge scale using simulators. Other features provided by simulators that increase its necessity are Safety aspects, Graphical Interface, debug and trace support, etc. [8].

C. LIMITATION OF SIMULATION

Simulation is the most promising tool for mobile networks. However, simulation tools do also have major drawbacks. Since simulation depends upon a proper model based on solid hardware and physical layer assumptions thus simulation results are not exact enough to catch the outright conduct of WSNs and depend on a specific situation under examination. The simulator does not consider the hardware platform constraints and is based on virtual clocking scheme, therefore simulation is unsuitable model for meeting the challenge development and deployment challenge on the realistic environment [12]. Another crucial problem while using simulator is to justify that the service and performances offered by the simulation model are consistent with the real experimental implementation of the protocol. Furthermore, there is an inherent problem of trade-off between scalability and accuracy when simulating WSNs. Abundant sensor nodes strongly affect simulation performance and scalability. Moreover, satisfactory outcomes require a precise portrayal of the sensor radio channel. New viewpointts indigenous to WSNs must be fused in simulators (e.g., a physical environment and an energy model). Ultimately, to attain a real-time evaluation of a communication protocol, just two arrangements are left: network emulation and testbeds [13].

IV. EMULATION

Since quite a long years, progressions in fast processing and networked systems have permitted the quick advancement of network emulators, such as, Tossim, emstar, and Atemu, and so on. Emulator can be seen as a blend of both simulators and live testing. In contrast, to simulators, emulators don't utilize absolutely scientific models of traffic, channels, and conventions. End-frameworks, for example, Computers can be attached to the emulator and will act as though they are connected to a system. Emulators produce virtual test systems which imitate live systems, for example, LAN or WAN. Also, emulator purposefully presents some counterfeit impedances" on the system to test fortes of the tested convention. These impairments incorporate packets failure, noise, clog, and decreased data transmission with particular planning or consolidating delay over the system [12]. The primary target of the emulator is to figure execution and anticipate the effect of changes in the system over time, in light of these actualities it understands an application's responsiveness, throughput and optimizes technology decision pending. Well known Emulator explicitly intended for WSNs are depicted as follows:

TOSSIM [22]

Tossim emulator has been designed to emulate all the TinyOS applications. Tossim being a discrete event simulator while simulation, it fetches events from event queue (sorted by time) and executes them. In Tossim network behavior is captured based on bit granularity not on the packet level. TOSSIM has been written using two programming interfaces: Python and C++. Python enables users to dynamically communicate as a strong debugger with a running simulation. However, the interpreter slows down the performance as it takes lots of time to show the outcome, TOSSIM also has a C++ interface. It's usually very easy to transform code from one to the other. Currently, TOSSIM does not favor the measurement of energy.

EMSTAR [23]

EmStar is an event driven emulator that is known for developing and deploying complex WSN applications because EmStar has been built by taken care of all the constraints of WSNs like node failure, topology changes etc. Emstar works on networks of Motes with 32-bit embedded Microserver platforms. EmStar consists of libraries that implement message-passing IPC primitives, tools that support simulation, emulation, and visualization of live systems, both real and simulated, and **ser**vic**es** that support networking, sensing, and time synchronization. Key features of EMSTAR are code reusability, modularity, ease of use, high visibility and reactivity features.

ATEMU [24]

ATEMU has been designed to bridge the gap between actual deployments of the sensor network and simulations of the sensor network. ATEMU uses a hybrid approach in which the operation of individual sensor nodes is emulated in instructive manner and their interactions with each other through wireless transmission are simulated in a realistic way. ATEMU's capacity to simulate a heterogeneous sensor network is a distinctive characteristic. Using ATEMU, implementation on the MICA2 platform can be simulated correctly, as well as a full sensor network where the sensor nodes are based on distinct hardware systems themselves. The ATEMU platform involves XATDB, a debugger frontend that offers an outstanding tool for big sensor networks to learn and debug. The ATEMU platform depends on the use of specification documents based on XML setup. The ATEMU platform offers both sensor network research and sensor network deployment communities with numerous functionality.

VMNET [25]

VMNet is a specifically designed network emulator to be used by the wireless sensor network for realistic WSN performance assessments. To assemble virtual hardware parts, VMNet has an extremely modular architecture. Most popular Crossbow MICA2 motes can be emulated in this hardware setup. VMNet emulates WSN as a VMN (Virtual Mote Network). The activities of every element like the sensing units and other hardware peripherals in a sensor mote is emulated at CPU clock cycle stage. And it also keeps the record of every emulation. VMNet also emulates the signal loss and noise of the radio transmission channel by the communication between VMs with the effects of signal loss and noise. Moreover, VMNet takes parameter values from the real world and detailed running status of application code. As a result, the binary code of the target WSN application can be run directly on the VMN, and the application performance, both in *response time* and in *power consumption*, can be reported realistically in VMNet.

FREEMOTE [26]

For the development of a Wireless Sensor Network (WSN) software, Freemote Emulator is a lightweight and distributed Java based emulation instrument. This platform's objective is to assist the developing Java-based Motes based on optimized JVM (such as Squawk, Sentilla Point) and platforms (such as Java Cards, SunSPOT). As the Java language is well established, this emulator reduces the development phase's length and complexity. The Freemote Emulator focuses on credibility of conduct by blending emulated nodes and actual nodes that can be reached through a specific bridge rather than on precision of time-based performance assessment. The Freemote Emulator also offers a strong network visualization tool that shows emulated nodes, message content sent, and physical topology. This characteristic enables the comprehension of the behaviors of complicated networks and debugs difficult execution issues.

MSPSIM [27]

MSPsim having MSP430 series microprocessor and Java-based instruction level emulator. It emulates full sensor networking platforms such as Tmote Sky and ESB/2 as opposed to CPU level emulators. MSPSim targets realistic simulation with precise timing for use as a study instrument, as well as excellent debugging support for use as a development instrument. MSPSim combines cycle precise interpretation of CPU instructions with the simulation of all other parts, both inner and external, based on a discrete event. MSPSim utilizes an event-based execution kernel that allows precise timing while minimizing the use of the host processor.

A. NECESSITY OF EMULATION:

Emulation is particularly useful in the debugging and testing phase of a system. Emulators give its users a privilege to connect their devices, services, and applications to test their performance, maturity and functionality against real-world network settings. These tools can also be used for various other purposes such as quality assurance checking, proof of concept, or troubleshooting. Once the network is tested in a controlled environment against actual network surroundings, users can have self-reliance that the elements being tested will perform as expected. Emulation addresses the original hardware and software environment of the digital object, thus it may provide more realistic observations. Emulators focus on regenerating an original computer environment. Results obtained from emulation are valuable because of its ability to maintain a proximate association to the authenticity of the digital object.

B. LIMITATION OF EMULATION

An emulator works by mimicking every facet of the original device's behavior, thus it offers more precise results. From this description, it seems as, an emulator is a better solution than the simulator, but in reality, it is not always true because emulators are really slow specially when the careful conduct of the framework to be emulated isn't recorded and must be deduced trouth reverse engineering. Emulating the actual hardware usually makes the software run considerably slower than it would have on the original hardware.

V. TESTBEDS

A Testbed is a platform that forms the basis for experimentation in real-world settings. Also, they are used by numerous analysts to assess applications pertaining to specific areas as as it permits thorough, adaptable, straightforward and replicable testing of hypotheses, computational instruments, and advancements [36-37]. Live testing is a dominating technique where real hardware equipment, as well as programming peripherals, are utilized to test and investigate network protocol during the system development stage. A testbed is a straightforward and reliable platform for directing precise, testing of computational apparatuses, logical hypotheses and new advancements. Testbeds give a situation that supports the assessment of various physical parameters in a controlled and dependable condition. This condition contains equipment, instrumentations, test systems and other subordinate components expected to lead a test. If we contrast WSN-Testbed with WSN simulator, we found that testbed empowers progressively reasonable and solid experimentation in catching the nobility of the fundamental equipment, programming, and elements of the remote sensor organize [13]. In any case, this methodology is extravagant if the system under thought has an enormous area. It is viewed as that live testing is unyielding to assess all parts of the convention being tried. List of some widely used testbeds for WSNs are given beneath:

WSNTB [28]

WSNTB is a heterogeneous, versatile, reconfigurable, and expandable testbed. Clients can explore whenever and anyplace and abstain from investing energy in setting up the WSNs condition.

The neighborhood mode, continuous control convention, and equipment reset devices on WSNTB is the fundamental belief. Clients can run any redid applications independent from anyone else and work with any hub over the Internet. Using testbed website, clients can book a timeframe furthermore, picks hubs to explore. At that point, clients can transfer their very own projects and reinventing them. There are other numerous capacities, for example, compiler, online editorial manager, data logger, working status screen, work line framework, etc. By using our testbed, analysts can spare heaps of time on building a colossal trial condition and decrease the equipment cost, yet improve the devise utility rate, experimental results.

INDRIYA [29]

INDRIYA is a scalable, inexpensive, 3D WSN testbed deployed at the National University of Singapore across three floors of the computer college. Indriya utilizes TelosB equipment and is based on a USB infrastructure. The infrastructure functions as a distant back-channel programming and also provides sensor equipment with electrical energy. Indriya is intended to decrease the cost of a large-scale test bed installation and maintenance.

MOTELAB [30]

MoteLab, a Web-based sensor network testbed, comprises of a collection of permanently deployed sensor network nodes linked to a main server that handles programming and information logging while offering a web interface to create and schedule testbed employment. By streamlining access to a big, fixed network of actual sensor network devices. By streamlining access to a big, fixed network of actual sensor network devices, MoteLab accelerates the implementation of applications. By automating information logging, it accelerates debugging and growth, enabling offline evaluation of the efficiency of sensor network software. In addition, MoteLab provides local and remote customers with access to the testbed and its scheduling and quota scheme to guarantee fair sharing by offering a web interface. The source of MoteLab is freely accessible, simple to install and as of now being used at a few other research organizations.

TUTORNET [31]

Tutornet is a WSN testbed that can be operated wirelessly with less power requirement and is installed at the University of Southern California (USC), Ming Hsieh Department of Electrical Engineering. It is widely used at USC for studies and teaching. Initially, Tutornet was produced using 113 sensor nodes (91 TelosB, 13 MicaZ, 9 OpenMote) in the Ronald Tutor Hall building covering two adjoining floors. It features three generations of wireless sensor nodes, all running on the 2.4 GHz frequency band with IEEE802.15.4-compliant radios.

REALNET [3]

REALNET is an environmental WSN testbed with an idea is to monitor ecological parameters like temrature, water level, soil humidity, light etc and to turn into a conventional stage that serves various purposes like a decisive framework for the scholastic and business networks, a compelling ecological control device, and a supportability cognizant device. REALNET was developed and implemented in 2001 at a university campus with an intial goal to measure tempratue and water level of a pond in the campus so that it can alert residents about danger of flooding.

A. NECESSITY OF TESTBEDS

WSN-Testbeds provide the means to researchers and developers to acquire firsthand experience and to discover and explore different kinds of scenarios. Testbeds conquers any hindrance among simulation and development of real devices by giving realistic environments for testing. In this way testbeds are useful to improve the speed of advancement and gainful research [14]. Thinking about the real weaknesses of test systems and emulators, using WSN testbeds to assess calculations and conventions of WSNs is essential before applying them into certifiable applications. To accomplish high constancy in WSNs tests analysts use testbeds.

B. LIMITATIONS OF TESTBED

Generally, testbeds specially designed for wireless sensor networks provide a more accurate, realistic and reliable validation mechanism for algorithm and protocols. However, the cost of deployment and sustainment of large-scale testbeds limits their applicability [15]. The Cost gets increases when the network domain increases so it is beyond the realm of imagination to expect to lean toward testbeds to assess certain parameters since it will be progressively costly and take colossal exertion and time. Moreover, in certain applications, there are a few circumstances when it ends up difficult to utilize this methodology effectively in light of the fact that new technology support is not yet validated or available. Besides in some applications where destructive conditions are being contemplated, for example concoction contamination, a real testbed becomes an unwanted choice.

VI. COMPARISON OF EXPERIMENTAL TOOLS

Network simulator is of different sorts and they can be analyzed using a few criteria like Range (from the extremely easy to the complex), specifying nodes and connections between those nodes and traffic between the nodes, determine everything about the conventions used to deal with traffic in a system, graphical applications (enable clients to effectively envision the operations of their reenacted

Table 1 Simulation Tools for WSNs

Simulator	Type	OS	Language used	Latest Version	License Type	Specific to WSN	Released year
NS-2 [12]	Discrete event	Unix, Windows, Cygwin	C++, OTcl	2.35 (4/11/2011)	Open Source	No	1989
NS-3 [13]	Discrete event	Unix, Linux	C++, Python	NS-3.29 (4/9/2018)	Open Source	No	2006
Omnet++ [14]	Discrete event	Windows, Linux, Mac OS X, Unix	C++	5.4.1 (29/06/2018)	Open source, commercial	No	2005
J-Sim [15]	Discrete event	Windows, Linux, OS/2 , Solaris	Java	0.6.0 (6/07/2006)	Open source	No	2006
Avrora [16]	Discrete Event	Tiny OS	Java	1.7.106 (15/06/2010)	Open source	Yes	2008
Opnet [17]	Discrete event	Windows, Linux	C, C++	18.7.1 (22/06/2018)	Commercial	No	1987
Qualnet [18]	Discrete Event	Windows , Linux, Unix, Solaris	C,C++	8.2 (31/05/2018)	Commercial	No	2000
Glomosim [19]	Discrete event	Windows , Unix	C/Parsec	2.03 (2010)	Open Source, Commercial both	No	1999
Prowler [20]	Discrete Event	Tiny OS	Matlab, Java	1.25 (28/01/2004)	Open Source	Yes	2004
Emsim [21]	Discrete Event	Windows Mac, Linux	Java	2.0 (22/03/2013)	Open Source	No	2007
Cooja [22]	Discrete Event	Contiki OS	Java	3.0 (25/08/2015)	-	Yes	2011
Mannasim [23]	Discrete Event	Unix Windows	Java, Tcl	2.35	Open Source	Yes	2012
Castalia [11]	Discrete Event	Windows,linux, Mac OS X, Unix	C++	3.3	Open Source	Yes	2006

condition.), text-based applications (grant further advanced types of customization), programming-oriented tools and so on. In Table 1 list of most extensively used simulators (NS-2, NS-3, OMNET++, J-Sim, Avrora, Opnet, Qualnet, Glomosim, Prowler, Emsim, Cooja, Castalia and Mannasim) for WSNs are introduced. Table 2 gives the framework of reviewed emulators (Tossim, Emstar, ATEMU, VMNET, Freemote, and MSPSim) that are widely being utilized for WSNs. Type, Language, platform, most recent form, and introducing year of each simulators and emulators have been investigated. In Table 3 unique features of testbeds (WSNTB, INDRIYA, Motelab, TutorNet, and RealNet) have been featured empowering specialists in picking up most appropriate tools to their necessities.

Table 2: Emulation Tools for WSNs

Emulator	Type	Operating System	Language Used	Latest Version	License Type	Specific to WSN	Releasing Year
Tossim [24]	Discrete Event	Tiny Os, Linux, Cygwin	Python, C++	2.1.1 (6/4/2010)	Open Source	Yes	2003
Emstar [25]	Discrete Event	Linux	C	-	Open Source	Yes	2004
ATEMU [26]	Discrete Event	Solaris Linux, Tiny OS	C	0.4 (11/07/2006)	Open Source	Yes	2004
VMNET [27]	Discrete Event	Windows, Linux	C	1.0.2 (30/10/2005)	Open Source	Yes	2004
FreeMote [28]	Discrete Event	Tiny OS	Java	Version 9 (20/10/2010)	Open Source	Yes	2008
MSPSim [29]	Discrete Event	Windows	Java	0.97 (30/04/2009)	Open Source	Yes	2008

Table 3: Testbeds for WSNs

Test Beds	Features
WSNTB [30]	• Designed for heterogeneous WSN experiments. • Users can use both the web-based interface and the special function, called local mode, to run their applications on testbed. • It improves two WSNs and three gateways. Each WSN has 17 sensor nodes.
INDRIYA [31]	• It is a large scale 3D WSN testbed with 140 TolesB nodes. • INDRIYA has only two platforms in terms of heterogeneity of hardware setup. • It does not consider node mobility.
MoteLab [32]	• It is web based sensor network testbed. • It is an open source tool. • All Motelab code is also available for institutions to build their own testbeds.
TutorNet [33]	• It is a Tiered Wireless Sensor Network testbed consist of 3 tiers, 13 clusters and 113 Motes. • It provide emote and parallel programming mote.
RealNet [34]	• RealNet is an environmental WSN testbed. • RealNet is primarily used to monitor physical parameters from the air (atmospheric temperature, humidity, and pressure, light), ground and water.

VII. DISCUSSION

Analyzing methods to study the performance of wired and wireless networks can be classified into three major categories: analytical methods, usage of experimental and research tools, and physical measurement [10]. Sensor networks are imposed with many constraints such as lack of Power supply, distributed collaboration and dynamic characteristics, and due to these imperatives calculations intended for sensor systems turns out to be considerably progressively perplexing and normally inadmissible for systematic techniques that have been persuaded to genuinely successful for traditional networks [35]. Many network particularities in WSNs are not planned and institutionalized at this point.

Setting up WSN testbeds is extravagant and troublesome (as far as both physical and mental work) [36]. Moreover, running real experiments is consistently a tedious procedure. Using research tools is the only workable approach for the quantitative analysis of Sensor networks [37]. In this context three standard methods of Research and development being discussed are: Simulation, Emulation, and live testing. Out of the three techniques, network simulation is cheap and provides immediate results [38]. It is an inarguable fact that in some circumstances the use of simulation software package is the only remaining option for the researchers that can satisfy their need, or at least can make the entire procedure simpler and quicker [39]. Most recent research in WSNs is being done using simulation, for example, to analyze state-of-the-art congestion control protocols in WSNs, such as LEACH-FL, Fuzzy AQM, and FCCTF. The most popular simulators are NS-2 and MATLAB. [40]. For energy-aware routing, clustering and security mechanisms OMNet++ and prowler simulators are the most frequent choice. That is why simulation is the preferred choice for the majority of the mobile ad hoc network community [41]. However, if a simulation experiment is not properly designed, and the appropriate simulator is not used there could be a huge difference between the obtained result and the actual result. [42]. Even the simulators that have been regarded as best for particular applications still cannot simulate real wireless communication environment as far as completeness, accuracy, and authenticity is concerned [43]. On the other hand, the live testing method is very authentic, but it is rarely applied due to the lack of reproducibility and great expense [44]. System Emulation lies between the above two procedure and is an integration of a simulated environment and a real testbed because it allows clients to frame a controlled examination with a high level of reproducibility and give an instrument to take a shot at a genuine framework viably. In this way, network emulator helps to improve the accuracy and scalability of testing results [45].

VIII. CONCLUSION

Rigorous research activities are going on to discover more possibilities of deployment and applications of Wireless Sensor Networks in the real world. So various researchers across the world are moving towards this field with their novel ideas, conventions, methods, and calculations to conquer the difficulties encompassing sensor systems. These thoughts and strategies require cautious assessment and investigation however but in case of wireless sensor networks it isn't doable to test and assess the conventions or speculations through real experiments because of cost and complexity factor. So, as a solution to these problems, experimental tools and techniques are proposed to analyze and validate the performance of protocols and algorithms. In this article apparatuses and procedures utilized for the wireless sensor system has been profoundly investigated. We have explored various state-of-the art simulators, emulators and test beds that have been used for experimental purpose in WSNs further we have discovered their plus points and pointed out their limitations also. In this way, this paper will assist the analysts in choosing suitable instruments for their necessities and approval by thinking about their qualities and impediments. The future extension is to build up an adaptable exploratory tools for the assessment of complex elements of wireless sensor networks that would consider both its resource limitations as well as its application requirements.

REFERENCES

1. Zheng J, Jamalipour A. Wireless sensor networks: A networking perspective, A John Wiley & Sons, INC., 2009.
2. B. Musznicki and P. Zwierzykowski, Survey of simulators for wireless sensor networks. *Int. J. Grid Distrib. Comput.* vol. 5, no. 3, pp. 23-50, 2012.
3. M. Korkalainen, M. Sallinen, N. Karkkainen, P. Tukeva, Survey of wireless sensor networks simulation tools for demanding applications, In: *Proc Fifth IEEE International Conference on Networking and Services (ICNS), 2009.*
4. MZ. Khan, B. Askwith, F. Bouhafs, and M. Asim, Limitations of simulation tools for large-scale wireless sensor networks, In*: Proc. 2011 IEEE Workshop Int. Conf. Adv. Inf. Netw. Appl. (WAINA),* Mar. 2011, pp. 820-825.
5. F Yu, A Survey of Wireless Sensor Network Simulation Tools 2011. In: *Journal of Wireless Networking and Communications 2015.*
6. F. Al-Fayez, A. Abuarqoub, M. Hammoudeh, A. Nisbet, Wireless Sensor Network Simulation: The Current State and Simulation Tools, In: *Sensors & Transducers,* Vol. 18, January 2013, pp. 145-155.
7. Chhimwal, P., Rai, D.S., Rawat, D. Comparison between different wireless sensor simulation tools. *IOSR-JECE* 2013, 5, 54-60.
8. Chand BS, Rao KR, S. Sreedhar Babu Exploration of New Simulation Tools for Wireless Sensor Networks, *IJSR*, *India,* Volume 2, April 2013.
9. Lakshmanarao K, Ch.R. Vinod Kumar, Kanakavardhini K, Survey on simulation tools for Wireless networks, *IJERT* Vol 2, October-2013.
10. A. Nayyar and R. Singh, A comprehensive review of simulation tools for wireless sensor networks (WSNs*)," J. Wireless Netw. Commun*, vol. 5, no. 1, pp. 19-47, 2015.
11. A. Dwivedi and O. Vyas, An exploratory study of experimental tools for wireless sensor networks," *Wireless Sensor Netw.*, vol. 3, no. 7, pp. 215-240, 2011.
12. https://www.isi.edu/nsnam/ns/
13. https://www.nsnam.org/research/wns3/wns3-2019/
14. Kopke et al. Simulating wireless and mobile networks in omnet++: The mixim vision. In *First International OMNeT++ Developers Workshop, March* 2008.
15. http://www.kiv.zcu.cz/j-sim/
16. B.L. Titzer, D.K. Lee, and J. Palsberg. Avrora: scalable sensor network simulation with precise timing. In *Proc: 4th international symposium on Information processing in sensor networks (IPSN), April* 2005.
17. Chang, X. (1999, December). Network simulations with OPNET. In *Proc: 31st conference on Winter simulation: Simulation---a bridge to the future-* vol 1 (pp. 307-314). ACM.
18. https://www.scalable-networks.com/qualnet-network-simulator-software
19. GloMoSim: a library for parallel simulation of large-scale wireless networks. In *Parallel and Distributed Simulation*, *1998. PADS 98. Proceedings . Twelfth Workshopon* (pp. 154-161). IEEE.
20. Simon, G. (2003). Prowler: Probabilistic wireless network simulator. *Institute for Software Integrated Systems, Nashville*.
21. https://www.princeton.edu/~cad/emsim/
22. F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and ¨ T. Voigt. Cross-level sensor network simulation with COOJA. In *Proc: First IEEE International Workshop on Practical Issues in Building Sensor Network Applications (SenseApp 2006),* Tampa, Florida, USA, November 2006.
23. Pereira, R.M., Ruiz, L.B., & Ghizoni, M.L. (2015). MannaSim: A NS-2 Extension to Simulate Wireless Sensor Network.
24. Levis P , Lee N, Welsh M, Culler D,TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications,*SenSys 2003*, pp.1-50.
25. Girod L, Elson J, Cerpa A, T. Stathopoulos, N. Ramanathan, D. Estrin EmStar: a Software Environment for Developing and Deploying Wireless Sensor Networks.
26. ATEMU.Sensor Network Emulator/Simulator/Debugger http://www.isr.umd.edu/CSHCN/research/atemu.
27. https://www.cse.ust.hk/vmnet/

28. Maret T, Kummer R etal. Freemote Emulator: A Lightweight and Visual Java Emulator for WSN, In Proc: *6th International Conference, WWIC 2008* Tampere, Finland Proceedings pp. 92-103, May 28-30 2008.
29. J. Eriksson, A. Dunkels, N. Finne, F. Osterlind, and ¨ T. Voigt. MSPSim – an extensible simulator for MSP430-equipped sensor boards. In *Proc: European Conference on Wireless Sensor Networks (EWSN)*, Poster/Demo session, Delft, The Netherlands, January 2007.
30. Sheu JP, Chang CJ, Sun CY , Wei-Kai Hu et al. WSNTB: A testbed for heterogeneous wireless sensor networks, In: *2008 First IEEE International Conference on Ubi-Media Computing.*
31. Doddavenkatappa M., Chan M.C., Ananda A.L. (2012) Indriya: A Low-Cost, 3D Wireless Sensor Network Testbed. In: Korakis T., Li H., Tran-Gia P., Park HS. (eds) *Testbeds and Research Infrastructure. Development of Networks and Communities. TridentCom 2011*. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 90. Springer, Berlin, Heidelberg.
32. Allen GW, Swieskowski P, Welsh M, MoteLab: a wireless sensor network testbed, In: *Fourth International Symposium on Information Processing in Sensor Networks, 2005.(IPSN 2005).*
33. https://anrg.usc.edu/www/tutornet/
34. J Albesa ,R Casas ,MT Penella , M Gasulla , REALnet: An environmental WSN Testbed In: *2007 International Conference on Sensor Technologies and Applications (SENSORCOMM 2007).*
35. V Kushwaha, A Study of Research Tools and Techniques in Network Congestion Control, *IJERT*, Vol.3, February - 2014.
36. V Kushwaha, R. Gupta Congestion Control for high-speed wired network: A systematic literature review *IN: J NETW COMPUT APPL* pp. 62-78, July 2014.
37. DP Kumar, T. Amgoth, CS Rao Annavarappu "Machine Learning Algorithms for wireless sensor networks: A survey" Information Fusion vol. 49 2019.
38. Oh Seok Kwon, Kye-Dong Jung, and Jong-Yong Lee, "WSN Protocol based on LEACH Protocol using Fuzzy",In : International Journal of Applied Engineering Research, Volume 12, Number 20 (2017) pp. 10013-10018.
39. Rezaee, A.A. & Pasandideh, F. Wireless Pers Commun (2018) 98: 815. https://doi.org/10.1007/s11277-017-4896-6
40. B. T. Nguyen, L. Murphy and G. Muntean, "Energy-Efficient QoS-Based Congestion Control for Reliable Communications in Wireless Multimedia Sensor Networks," 2018 IEEE International Conference on Communications Workshops (ICC Workshops), Kansas City, MO, 2018, pp. 1-6. doi: 10.1109/ICCW.2018.8403757
41. Yang X, Chen X, Xia R, Qian Z. Wireless Sensor Network Congestion Control Based on Standard Particle Swarm Optimization and Single Neuron PID. Sensors (Basel). 2018; 18(4):1265. Published 2018 Apr 19 doi:10.3390/s18041265.
42. R. V. Kulkarni, A. Forster and G. K. Venayagamoorthy, "Computational Intelligence in Wireless Sensor Networks: A Survey," in IEEE Communications Surveys & Tutorials, vol. 13, no. 1, pp. 68-96, First Quarter 2011.doi: 10.1109/SURV.2011.040310.00002
43. K.Darymli, MH. Ahmed, Wireless Sensor Network Testbeds: A Survey". 2012.
44. V Kushwaha, Ratneshwer, “A Review of Router based Congestion Control Algorithms”, International Journal of Computer Network and Information Security; Vol. 6, no. 1, pp.1-10.
45. L. P. Steyn and G. Hancke, A survey of wireless sensor network testbeds," In: *Proc. AFRICON*, Sep. 2011, pp. 1-6.

AUTHORS PROFILE

Divya Pandey is a Research fellow under supervision of Dr. Vandana Kushwaha in Department of Computer Science, Institute of Science, Banaras Hindu University, Varanasi (India).Her research interests include, Wireless Sensor Network, Network algorithms and Congestion control protocols.

Dr. Vandana Kushwaha is working as an Assistant Professor in Department of Computer Science, Institute of Science, Banaras Hindu University, Varanasi (India). She has 11 years of teaching experience. She has done her Ph.D. in Computer Science at the Department of Computer Science, Institute of Science, Banaras Hindu University, Varanasi (India). Her research interests include High Speed Network, Wireless Sensor Network, Network algorithms and Congestion control protocols. She has 6 papers in international Journals and one paper in international conference. She has 2 book chapters (from CRC Press) in her credit.