Nile Journal of Communication & Computer Science
Volume 7, June 2024
Journal Webpage: https://njccs.journals.ekb.eg

A Comparative Study of Simulation Tools for WSNs and UWSNs Research

Nesma Abd El-Mawla[*1], Hegazi Ibrahim[2]
[1,2]Communication and Electronics Department, Nile Higher Institute for Engineering and Technology, Mansoura, Egypt,
[*1]ORCID ID: 0000-0003-2232-1035, [2]ORCID ID: 0000-0002-9437-8367.

Abstract

Wireless sensor networks (WSNs), which use radio waves to transmit data from individual sensors, have found widespread application in many disciplines. Coordinated sensing allows continuous tracking of environmental conditions like temperature, humidity, light levels, and gas concentrations at each node. The WSN area is continuously researched and developed. However, many researchers in the academic community and industry are working on WSNs that do not have live sensors to test, creating a need for simulation. Simulation tools reduce deployment costs and time. System simulation tools are often required during the pre-run system configuration phase. Nodes in an Underwater Sensor Network (UWSN) can either operate independently or collect and transmit data to other nodes or mobile stations, depending on their depth. UWSN has several research directions. Deploying complete testbeds with complex network structures, topologies, protocols, and data links to validate new network protocols or algorithms is expensive. Therefore, there is an urgent need for a simulation environment that can depict real underwater scenarios. Many simulation tools are proposed for UWSNs, but selecting the proper one due to the research requirements is very important to validate and interpret the research results. Furthermore, it is essential to see that the test systems' outcomes are sound and meet standard benchmarks. This paper presents a detailed comparison of different WSNs simulators to reduce the time and effort of researchers and application developers. This paper aims to provide a detailed comparison and review of different WSNs and UWSNs simulation tools due to their performance and main features. The research directions highlighted in this study will assist researchers in choosing the optimal tool and incorporating additional features into UWSN simulators for real-time underwater simulations.

Keywords: *WSN, sensor networks, simulation, simulators, underwater.*

1. INTRODUCTION

Nowadays, there is fast growth in both wireless and communication technologies. This growth leads to a widely used wireless networking technology called Wireless Sensor Networks (WSNs). It eases the interaction between humans and the physical world. Such networks have numerous applications in our daily lives as sensors are installed in shopping centers and big malls to help people find the required products easily. Sensors in hospitals monitor patients' conditions, while those in forests can alert authorities to impending dangers like wildfires. These benefits, while substantial, are just a taste of what the widespread use of WSNs could bring [1-3]. WSNs' potential utility in various environments can be demonstrated by future research.

New research has been put out globally to confirm that underwater sensor networks (UWSNs) are operational and extremely effective against all limiting conditions. However, it can be expensive in terms of time, money, and effort to evaluate novel protocols, models, architectures, and procedures in a real-time setting. Therefore, simulation-based testing is the ideal approach to streamline these tests in terms of cost, time, and behavior observation of the protocol/model and architecture.

Even the simulation-based solution can test massive networks with tens of thousands of UWSN nodes and is simple to install. Knowing the various simulation tools for UWSN scenarios is crucial for designing, deploying, and testing novel protocols in real-world UWSN simulation situations. Although simulator designs are precise, packed with features, and simple to use, not all simulators are appropriate for all situations and research. Therefore, UWSN and its simulations should be known to developers and researchers. These simulators aid researchers and engineers in seeing and thoroughly comprehending underwater sensor network real-time scenarios with various dynamic factors.

Moreover, discrete-event simulations, trace-event simulations, and Monte Carlo simulations are the three different forms of simulations. Since diverse activities may be readily simulated on various sensor nodes, discrete-event simulation is the foundation of most sensor network simulation tools. This simulation's initialization, input, output, and trace procedures make it simpler for programmers to manage dynamic memory. Tracing simulation, on the other hand, is helpful when thinking about real-time system simulation. The computerized Monte Carlo simulation, used to simulate a wide range of engineering issues, is based mostly on mathematical simulation techniques.

This paper makes a significant contribution in three key areas. Firstly, conceptual elucidation: it provides a comprehensive clarification of WSN technology, encompassing its significance, applications, advantages, and limitations. Additionally, the paper offers an in-depth exploration of the various WSN types. Secondly, simulation landscape analysis: the paper presents a well-structured study of WSN and UWSN simulation environments. This includes a discussion on the purpose of simulation, essential requirements for effective simulations, and the architectural considerations for WSN, and UWSN simulators. Thirdly, future research directions: the paper identifies and explores potential challenges that future WSN and UWSN simulators will need to address. This forward-looking perspective paves the way for further advancements in simulation tools.

Finally, This paper is organized as follows: Section 2 provides an overview of WSN, discussing its significance, usefulness, benefits, and drawbacks. In this section, we also introduce the various WSN types. The purpose of simulation, its primary requirements, and the simulator's architecture are discussed in Section 3. In Section 4, we will discuss the various simulation tools currently in use and their benefits and drawbacks. The paper concludes with suggestions for where the field could go.

2. WIRELESS SENSOR NETWORKS

Sensor nodes in a wireless sensor network (WSN) collect and transmit data about environmental conditions such as temperature, humidity, and light intensity. To develop communication and information systems that increase the effectiveness and dependability of infrastructure systems, WSN is viewed as a creative method of data collection. WSNs are distinguished by simpler deployment and more flexible device options than wired alternatives [4]. The training of each node in this network has several issues. Recent advancements in wireless technology have aided in creating compact wireless sensors that are low-cost, low-power, and multifunctional [5].

WSN is now widely used in various applications and offers too many advantages. It is widely used to control commercial buildings and agricultural and environmental wireless sensors. It is used in home automation and security applications: biological, chemical, and nuclear wireless sensors (sensors for explosives, biological agents, and toxic chemicals). Also, it has many applications in metropolitan operations, industrial monitoring, and military purposes [6-7]. However, WSNs have many strengths and weak points, which can be summarized as follows; The pros of WSNs are: (i) Reduce cost: avoid complex and much wiring. (ii) Extendable: adding a new device at any time. (iii) Flexibility: to go through the physical partition. (iv) Easy monitoring: access could be done using a centralized monitor. The cons of WSNs are: (i) Speed: comparatively low speed of communication. (ii) Security: easy for hacking. (iii) Cost: costly at large. (iv) Energy: the life of nodes and energy life. (v) Disturbance: get distracted by various elements.

2.1. Wireless Sensor Networks Classification

Wireless sensor networks can be classified due to their structural nature or use [8-9], as shown below in Figure 1. The best Simulators for WSNs, in general, and Underwater WSNs, in particular, are discussed in this paper.

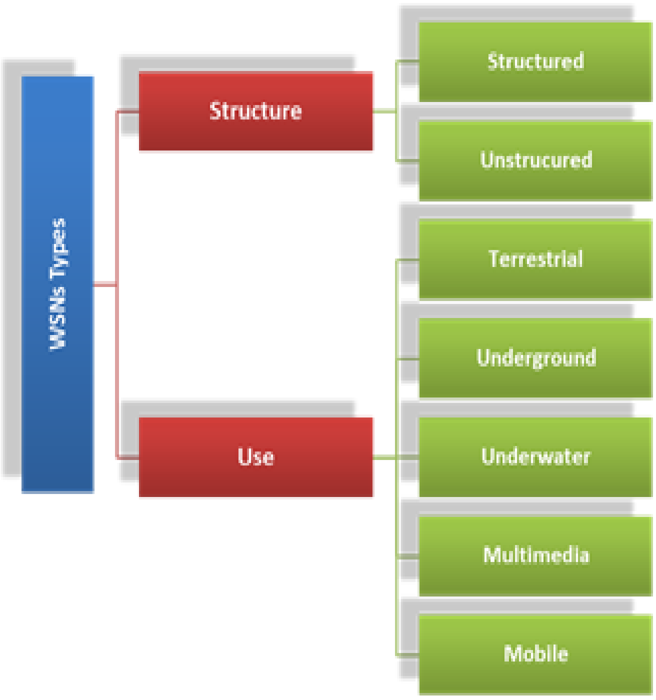

Fig.1. Types of WSNs.

2.2. Importance of Simulation in WSNs and UWSNs

Due to the rapid growth of WSN applications in every field of life, there are too many researchers around the globe working on it. Using good simulators reduces cost, time, and effort. It also shows the probable faults and failures in WSN. For example, one of the WSNs' common uses is underwater networks. In UWSNs, it is too expensive and hard to deploy a full testbed to test and validate the results, so there is a big need for modeling and simulation in this area [10-11].

3. REQUIREMENTS OF WSN MODELING AND SIMULATOR ARCHITECTURE

There is great growth in the number of simulation tools developed and deployed yearly. Unfortunately, this great number of simulators makes researchers confused in deciding which of them they should use. This section presents a well understanding of good simulator architecture. Also, this section discusses the characteristics and evaluation parameters of any simulator.

3.1.Simulator Architecture and Types

Types of simulation in wireless sensor networking can be classified into Monte Carlo, trace-driven, and Discrete Event [15, 16]. Discrete event and trace-driven simulations are widely used in wireless sensor network simulators. The main difference is that discrete event simulators are easy to use and present dynamic memory management. In addition, it eases the process of adding and deleting various simulation nodes, allowing users to evaluate code in parallel while the simulation is running successfully. In contrast, trace-driven simulators are effective in real-time applications because they provide more specific information that allows the user to get more details about the simulation model and add complexity to it.

3.2.Characteristics of a Good Simulation Tool

The key properties and features to say that a specific simulator is good can be defined as [26-27]: (i) Reusability and Availability: The simulation tool includes common model implementations, and new models can be easily modified or integrated into existing models. (ii) Performance and Scalability: It is tied to the effectiveness of programming languages. The simulation of 100,000 nodes is an open topic and a challenging problem. (iii) aid for a powerful semantic scripting language used for experiment definition and data analysis: It depends on several variables used to define the network, such as the number of nodes, their locations, their movements, their energy model, and the physical environment. (iv) graphical debug and trace support: It is important to observe and track simulation runs to detect improper behavior or errors in your network. Figure 2 briefs the main characteristics of a good simulator.

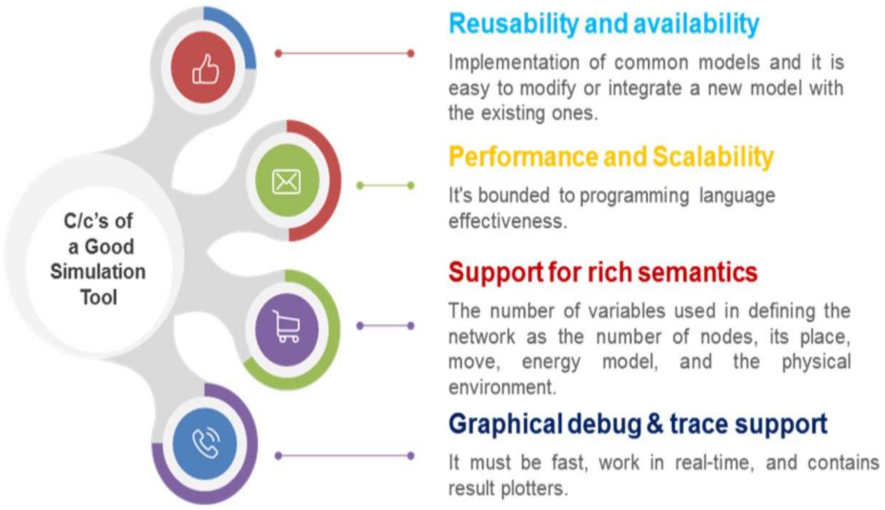

Fig.2. Characteristics of a Good Simulation Tool.

3.3. Evaluation Parameters

Many software programs are available for simulating WSNs, confusing users about which simulator to use in their work. So, criteria to evaluate each single simulation tool to determine which one matches the requirements of each user application is required [17, 18] [20-23]. The following parameters help users do this best: (i) Simulator Type: whether the simulator is generic, code level, or firmware [19]. (ii) Software License: It is about whether the simulation software is open source and free or commercial and its cost. (iii) Software Platform: It is the operating system that the simulator works on (Windows, Linux, Mac, cross-platform). (iv) Popularity: user number, number of hits on each search engine, and the software ranking ("Wireless Sensor Network <simulator name>" query measures popularity). (v) WSN Platforms: It supports different sensor types and platforms. (vi) Ease Of Code: It is about supporting high-level programming language with speed execution. (vii) Timing and Execution: whether the simulator is a discrete event or continues working as explained in (section 3.1.). (viii) Simulation Speed: How quickly the simulator executes different simulation scenarios. (ix) Available Models and Protocols: The number of protocols the simulator supports for each layer separately. (x) Energy Consumption Model: It is about the availability of energy consumption model support in the simulator. (xi) Graphical Results: Does the simulator have a GUI and not graph the results?

4. TOOLS FOR WIRELESS SENSOR NETWORK SIMULATION

Currently, there is already an excessive number of WSN simulators, causing confusion among researchers and users. Many simulators are simple, while others are more complex due to the in-depth details and control it present for users. This section discusses the various WSN simulators' ability to simulate UWSN. Also, this section presents a detailed comparison of commonly used simulators in both the research and industrial fields.

4.1. NS-2

Based on the first REAL network simulator, version 2 of the Network Simulator (NS-2) was developed in 1989. It is supported by grants from the NSF and the Defense Advanced Research

Projects Agency [13]. Simulating discrete events is the base of NS-2. It works on both Linux Operating Systems and Cygwin on Windows. It can simulate both wired and wireless areas. Modules in NS2 represent several parts of the OSI reference model, including the transport layer protocol, the routing layer, the application layer, etc. In Ns-2, an easy-to-use scripting language helps researchers configure the simulated network and observe the generated results. Finally, it is used widely in WSNs all over the world.

4.2. NS-3

NS3 is a discrete-event network simulator for Internet-based applications. NS3 is suitable for research and learning and makes up most of its users. In 2006, the NS-3 was released as a free program. It aids in model development, model maintenance, issue fixing, and result analysis and sharing [13]. The core C++ software that establishes the topology and starts the NS-3 simulator is developed as a library that is statically or dynamically linked. Additionally, practically all of NS-3's API is exported to Python.

4.3. TOSSIM

TOSSIM is a TinyOS WSN-based embedded system simulator that uses discrete events. It was developed to work with Linux distributions and Cygwin on Microsoft Windows. It is great for debugging, testing, and analyzing algorithms in controlled and repeatable environments. TOSSIM's main goal is to introduce a highly accurate simulation of TinyOS applications. While TOSSIM helps understand the causes observed in real-world behavior, it does not capture all of them and should not depend on it for complete evaluations.

4.4. J-Sim

J-Sim is a platform for component-based simulation created completely in Java. J-Sim offers process-based simulation in real-time. It may be used to implement WSN algorithms for data spreading, routing, and positioning [12–13]. J-Sim offers support for physical and sensory phenomena. IEEE 802.11 is the only MAC protocol available for wireless networks. At the Distributed Real-Time Computing Laboratory, the team created j-Sim (DRCL). J-Sim is open source and free to download.

4.5. OPNET

Compared to NS and GloMo-Sim, OPNET is different as it models a variety of sensor-specific devices. Additionally, it may be used to provide unique packet formats. An event-based network-level simulation tool is called OPNET [12–13]. Every simulation is run at the "packet level" of the fixed network. It has a large collection of precise models of fixed line hardware and protocols offered for sale in the market. Simulators offer much promise, typically missing from today's wireless systems. OPNET is regarded as a tool for network construction and analysis as well as for doing research. For developers, the user threshold is high; for end users, it is low. The main components of OPNET are the C and C++ source code blocks that make up the high-level user interface and the large library of OPNET-specific functions.

4.6. OMNET++

OMNET++ has more desirable simulation capabilities and better performance than NS2, OPNET, and many other WSN simulators compared to other well-known simulators. Designing a directed spreading protocol and running a performance analysis while simulating a WSN illustrates how to employ WSN simulation. In WSN simulations, OMNET++ scales better than NS2. Additionally, OMNET++ has proven to perform better than other simulators with big WSNs. OMNET++ is a C++ class library at the same time. In this regard, NS2 functions pale compared to OMNET++ functions, and C++ beats C in WSN simulations. The OMNET++ energy module also figures out how much energy is used. Sensor modules gather sensor data. When defining and connecting modules, use the NED language.

Finally, many other frameworks are available than OMNET++, such as INET. An OMNeT++ open-source library contains models for agents, protocols, and other tools for anyone learning about or researching communication networks. Using the OMNeT++ simulation IDE or the command line, you can develop, assemble, parameterize, run different models, and easily evaluate the results. Figure 3 presents a taxonomy of different simulation tools of WSNs [14].

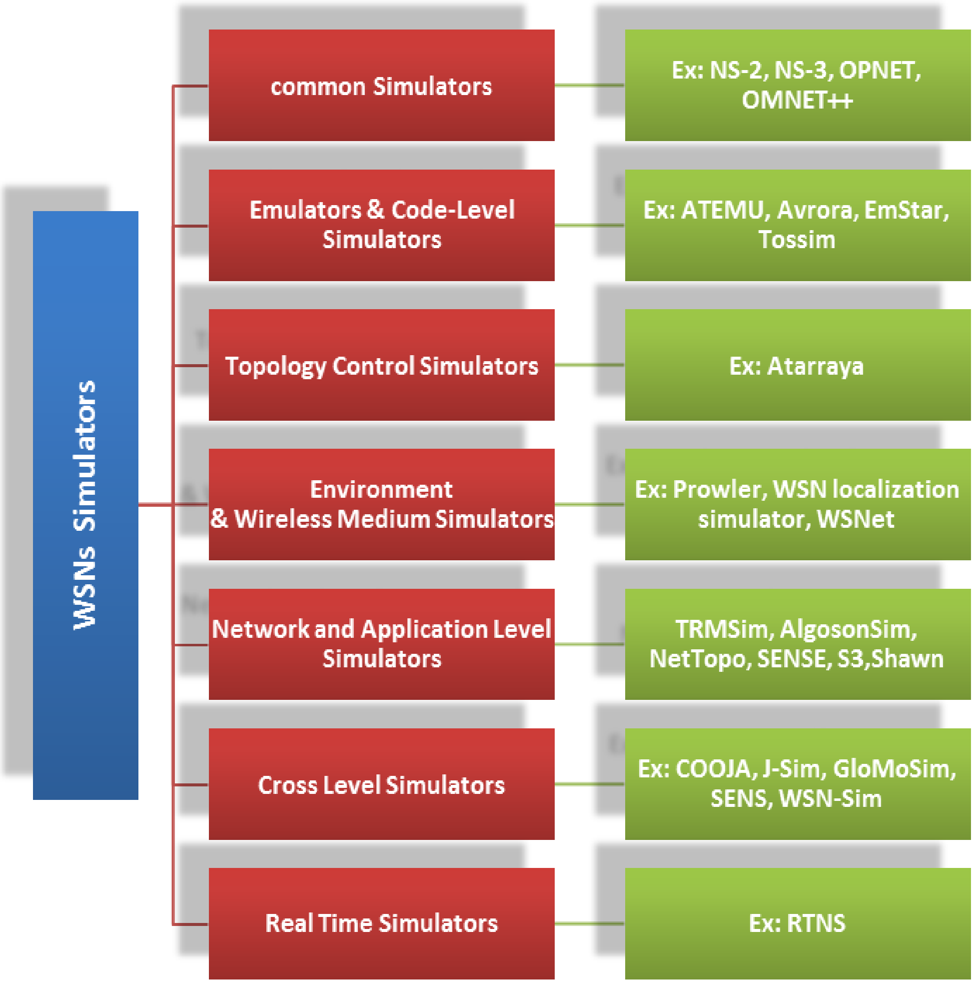

Fig.3. Taxonomy of WSNs Simulators.

Based on the characteristics of the good simulator in (section 3.2.) we can compare different simulators discussed in the current section. Table 1 contains a detailed comparison of different WSNs simulators according to different parameters, such as Discrete-Event Simulations (DES) or Trace-Driven Simulation (TDS)

5. SIMULATION TOOLS FOR UNDERWATER WIRELESS SENSOR NETWORKS

This section provides a high-level overview of simulators that can be used to model underwater sensor networks. Figures 4 and 5 present the hierarchy of various simulation tools and their classification. Some are designed specifically for use in underwater situations, while others are made primarily for use on land but can be modified for use in the water. In addition, not all simulators are created equal; some are designed specifically for validating and testing software, while others can be used for testing in real time.

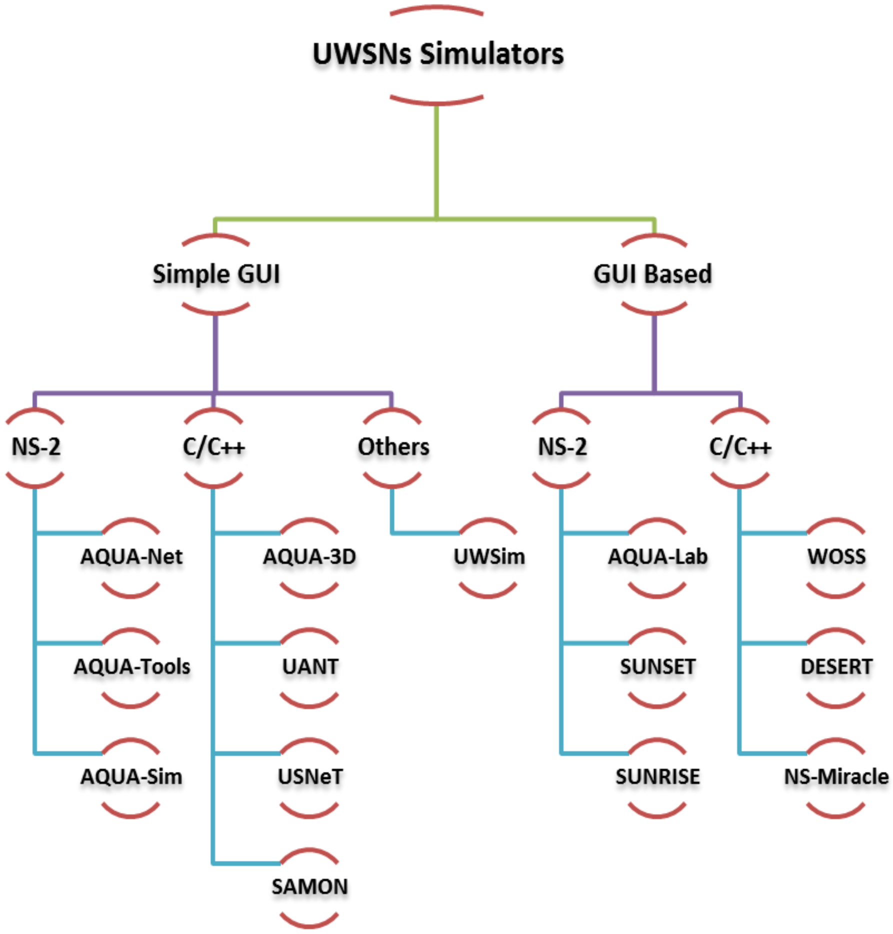

Fig.4. Taxonomy of UWSNs simulation tools based on programming language.

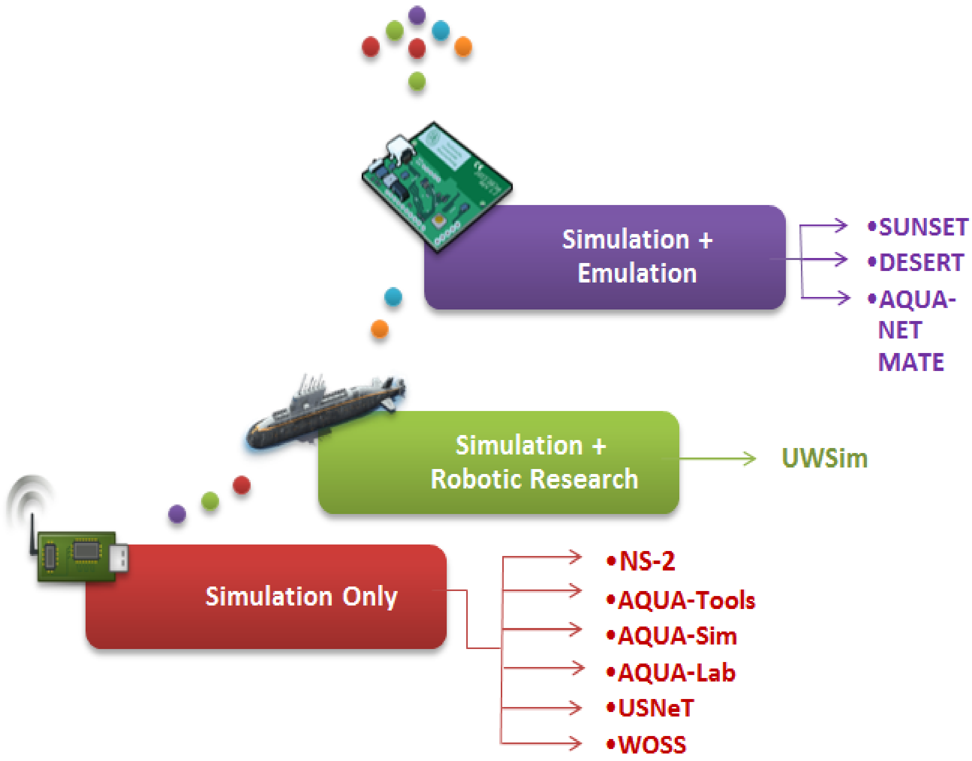

Fig.5. Classifications of UWSNs simulation tools.

The next sub-sections will highlight some of the best UWSNs simulation tools with simple or high GUI.

5.1. NS-2

A free and open-source discrete-event simulator is NS-2. It was founded by researchers from the University of California and Cornell University to facilitate collaborative networking studies. NS-2 is written in C++ and Object-Oriented Tool Command Language (oTCL) [37].

5.2. AQUA-Net

AQUA-Net is a complete hardware and software UWSN simulator [38]. It uses optimization techniques of varying types and relies on complex hierarchical structures. The experts developed it at UConn in Storrs. As a result, the simulator can now simulate effectively and reliably across the whole OSI model's hierarchy.

5.3. AQUA-Tools

The Computer Science Group at Jacobs created AQUA-Tools [39]. The software is a simulation toolbox for hydroacoustic networks. Based on his NS-2 simulator, this simulator is well-suited for modeling various underwater acoustic networks. High-efficiency elements like high transmission, water acidity, temperature, and nodal depth are featured in the simulator's 3-channel model.

5.4. AQUA-Sim

Researchers believe that AQUA-Sim, based on NS-2 and simulates a variety of UWSN-related problems, is the most extensively used simulator. This simulator was created using an object-oriented design methodology. The simulator's network elements have been developed as classes. AQUA-Sim is a very effective simulator for accurately modeling acoustic packet collision and signal attenuation conditions in UWSN. Additionally, 3D-UWSNs architecture is supported [35].

5.5. AQUA-3D

The University of Connecticut created a potent UWSN visualization tool for research and development [35]. The wxWidgets library for OpenGL and the GUI for rendering 3D graphics is both included in this C++-based system. The UWSN simulator generates trace files, which AQUA-3D reads and displays in 3D visuals.

5.6. NS-Miracle

NS-Miracle is considered a framework on top of NS-2, facilitating advanced networks such as UWSNs and ad-hoc sea networks. It provides an efficient engine for processing various cross-layer messages, allowing multiple modules to be used at each protocol stack layer. NS-Miracle was developed at Padua's Signet Institute. The MIRACLE physical layer module (M-PHY) and associated UMTS library are newly enabled modules in NS-Miracle. [35-37].

5.7. AQUA-Lab

The University of Connecticut at Storrs has developed a tool called AQUA-Lab, a sensor network for use underwater. AQUA-Lab provides researchers with easy access, customization, and specialized features in a realistic channel environment. The simulator is mostly compatible with the OSI model's physical layer [35–37].

5.8. WOSS

WOSS is a state-of-the-art simulation tool for UWSN that incorporates realistic propagation modeling and is based on the ideas behind NS-2 and NS-2 Miracle. Networking protocol testing involving acoustic propagation, cross-layer specifications, and physical layer modeling is done using it [35]. The University of Padova in Italy created it. Three UWSN protocols—ALOHA, aT-Lohi, and DACAP—were used to test WOSS. It offers developers a very adaptable interface for designing and developing new routing protocols at all OSI model layers.

5.9. DESERT

The University of Padua in Italy used C/C++ to create the DESERT [40] simulator for the underwater sensor network protocol. Over the network, data connection, and physical layers, DESERT can support all application and transport layer protocols. Through the enhancement of NS-Miracle, DESERT creates a novel protocol for UWSN. Both single-hop and multi-hop transfers using the same code have been successfully carried out on the DESERT testbed. It is strongly suggested that new protocols be designed and tested for UWSN. Table 2 presents a comparative study of the previous UWSN simulators due to their main features and the services produced. Furthermore, we can compare the different simulators we discussed in the current section based on the characteristics of each simulation tool. Finally, tables 3and 4 contain a detailed comparison of different UWSNs simulators.

6. CONCLUSIONS

In conclusion, the evolution and widespread application of WSNs have underscored the critical need for efficient simulation tools to propel research and development in this field. The intricate nature of UWSNs further emphasizes the necessity for accurate and reliable simulation environments. By providing a comprehensive comparison of various WSN and UWSN simulation tools, this study not only streamlines the selection process for researchers but also paves the way for enhanced real-time underwater simulations. As we navigate the complexities of modern sensor networks, the quest for advanced simulation tools remains paramount in driving innovation and progress in this dynamic field. Finally, we point out that researchers choose the right platform for their application based on comparing different simulation tools. Research motivations allowed researchers to choose the right simulation tool for the task. For future work, more research is needed to design common open-source simulation tools that can be accessed from anywhere in the world via the Internet. Such tools help researchers collaborate and conduct experiments remotely with reliable results at a relatively low cost. UWSN development will accelerate as more research efforts are put into simulation and experiment platforms.

Table 1. WSNS SIMULATION TOOL COMPARISON [24-30].

Name	Simulator or Emulator	DES or TDS	GUI	Open-Source / Free License	General or Specific simulator	Underwater Support	Details
NS-2	Simulator	DES	No	Yes	General simulator	Yes	1. It cannot simulate more than 100 nodes at once. 2. It cannot replicate the power consumption or bandwidth difficulties of WSNs.
NS-3	Simulator	DES	No	Yes	General simulator	Yes	It offers a device model of a basic Ethernet network using the CSMA/CD protocol architecture with exponentially growing back-off to compete for the same transmission medium.
OPNET	Simulator	DES	Yes	No	General simulator	Yes	1. It provides Customizable wireless modeling. 2. It works based on Discrete Events, Hybrid, and Analytical simulation.
OMNeT ++	Simulator	DES	Yes	noncommercial license, commercial license	General simulator	---	1. it is compatible with various WSN protocols, including the MAC protocol and several regional variants. 2. It is a power and channel control simulator. 3. it offers a few protocol options. 4. It is cross-platform software.
AT-EMU	Emulator	DES	Yes	Yes	Specifically designed for WSNs	---	1. it can simulate sensor nodes in both homogeneous and heterogeneous networks. 2. It can simulate power consumption or radio frequencies. 3. it takes a lot more time to run a simulation.
Avrora	Simulator	DES	No	Yes	Specifically designed for WSNs	---	1. It can simulate thousands of nodes. 2. It reduces the time required to carry out the procedure.
TOS-SIM	Emulator	DES	Yes	Yes	Specifically designed for WSNs	---	1. It is capable of simulating networks with thousands of nodes. 2. it simulates radio models and code executions. 3. Only emulate homogeneous applications. 4. powerTOSSIM is required for energy consumption modeling.
EmStar	Emulator	TDS	Yes	Yes	Specifically designed for WSNs	---	1. One major drawback is that it cannot simulate many sensors. 2. Only run in real-time simulation and only apply to IPAQ-class sensor nodes and MICA2 motes.
Atarraya	Simulator	TDS	Yes	Yes	Specifically designed for WSNs	---	It is good for topology construction and topology maintenance simulation.
TRM-Sim	Simulator	TDS	Yes	Yes	Specifically designed for WSNs	---	1. It presents trust and reputation models for WSNs. 2. Network topology can be loaded via XML files.
J-Sim	Simulator	TDS	Yes	Yes	General simulator	---	1. It can simulate too many nodes, around 500. 2. It can simulate power consumption and radio channels. 3. Long execution time than usual.
GloMo-Sim/ Qualnet	Simulator	TDS	Yes	Commercial	General simulator	Yes	1. It provides the scalability and portability to run hundreds and thousands of nodes with high-fidelity models on various platforms. 2. Parallel computing. 3. Linux-based platform.
RTNS	Simulator	Periodic and Event Simulation	Yes	Yes	Specifically designed for Real-Time Networks	---	By combining the Ns-2 environment and the Real-Time Operating System simulator, it offers real-time mechanisms for distributed networked applications (RTSim).

Table 2. COMPARISON OF MAIN UWSN SIMULATION TOOLS DUE TO THEIR FEATURES [35-37].

Name of simulator	GUI	Support for heterogeneity	Operating system	Key Features
NS-2	Simple	High	Linux and Windows	• The simulator may create, build, and test new protocols. • It has a visual simulation tool called NAM (Network Animator) that allows you to explore all the simulation properties. • The Xgraph tool may be used to evaluate all the findings.
AQUA-Net	Simple	Low	Linux	• Layered simulator. • Cross-layer optimization is supported. • Simple real-time integration with connection to physical sensors
AQUA-Tools	Simple	Medium	Linux	• Easy configuration, reusability, and manageability, of static and mobile sensor nodes. • Easy setup and an interactive GUI. • Highly adaptable scripting interfaces for creating and evaluating current and next proposed routing protocols. • Different channel models and climatic environments. • Support the development and advancement of routing protocols that are power-efficient routing protocols.
AQUA-Sim	Simple	Medium	Linux	• Discrete event simulator with a CMU Wireless package. • Highly accurate modeling of underwater acoustic channels. • A complete protocol stack, from the underlying physical layer up to the uppermost application layer, is implemented efficiently.
AQUA-3D	Simple	Low	Linux	• A powerful GUI that is interactive. • Simulation using exact time and speed adjustments. • Create a 360° rendering using a completely programmable camera. • The accuracy of 3D visualization can be verified through test scenarios generated from field tests.
NS-Miracle	Good	High	Linux	• Easy installation. • Simple add-ons may be developed FORNS-2. • No need for NS-2 code recompilation. • Dynamic libraries that are loaded during simulation
AQUA-Lab	Good	Medium	Linux	• Output real-time transmission loss. • BELLHOP ray tracking model. • Highly integrated environmental parameter databases enable end users to assess the procedures in nearly real-time circumstances.
WOSS	Good	High	Linux	• Compare new protocols to those already in use. • A web-based GUI for simulations and experiments by researchers. • Powerful in the design of new protocols.
DESERT	Good	Medium	Linux	• Provides a link between the simulator and the UWSN node performing the actual data collection in real-time. • Including various network layer, MAC, and application layer protocols.

Table 3. COMPARISON OF SIMPLE UWSNS SIMULATION TOOLS [35-37].

Name of simulator	Programming language	Simulator Type	Support for heterogeneity	Operating systems supported	Open source/commercial	Propagation model	Specification
NS-2	C, C++, oTCL	Discrete-Event	High	Linux and Windows	Open Source	Radio model	IEEE 802.15.4, IEEE 802.11, and IEEE 802.3
AQUA NET	NS-2	Discrete-Event	Low	Linux	Open Source	Thorp's Model	IEEE 802.11
AQUA-Tools	NS-2	Discrete-Event	Medium	Linux	Open Source	Fisher and Simmons, Thorp's, and Ainslie and McColm	IEEE 805.11
AQUA SIM	NS-2	Discrete-Event	Medium	Linux	Open Source	Thorp's Model	IEEE 802.15.4
AQUA-3D	C++, wxWidgets OpenGL	Discrete-Event	Low	Linux	Open Source	Thorp's Model	–
UANT	C++, TinyOS, TOSSIM	–	Medium	Linux	Open Source	–	IEEE 805.11
USNeT	C	Discrete-Event	Low	Linux	N/A	–	–
SAMON	C, C++	–	Medium	Linux	Open Source	–	–
UWSim	C#, .net Framework	Discrete-Event	Low	Linux	Open Source	–	–

Table 4. COMPARISON OF UWSNS SIMULATION TOOLS WITH GOOD GUI [35-37].

Simulator Name	Programming Language	Simulator Type	Heterogeneity Support	Operating systems supported	Open source/commercial	Propagation model	Specification
NS-Miracle	C, C++, oTCL	Discrete-Event	High	Linux	Open Source	–	IEEE 802.15.4
AQUA-Lab	NS-2, XML, PHP, AJAX, JavaScript	Discrete-Event	Medium	Linux	N/A	–	IEEE 805.11
SUNSET	NS-Miracle and NS-2	Discrete-Event	High	Linux	Open Source	Various acoustic models in a variety of simulation settings	IEEE 802.11
SUNRISE	NS-Miracle and NS-2	Discrete-Event	High	Linux	Open Source	–	IEEE 802.11
WOSS	C++	Discrete-Event	High	Linux	Open Source	–	IEEE 802.15.4
DESERT	NS-Miracle, NS-2, C and C++	Discrete-Event	Medium	Linux	Open Source	–	IEEE 802.15.4

REFERENCES

[1] Fahmy, H. M. A. (2020). Wireless sensor networks. Springer International Publishing.

[2] Abd El-Mawla, N., & Nagy, A. (2020). IoT and Civil Engineering Based Solutions for Global And Environmental Risks of 2019. Development, 20, 24.

[3] El-mawla, N. A., Badawy, M., & Arafat, H. (2019). Security and key management challenges over WSN (ASurvey). International Journal of Computer Science & Engineering Survey (IJCSES), 10(1), 15-34.

[4] Raza, S., Duquennoy, S., Chung, T., Yazar, D., Voigt, T. and Roedig, U., 2011, June. Securing communication in 6LoWPAN with compressed IPsec. In Distributed Computing in Sensor Systems and Workshops (DCOSS), 2011 International Conference on (pp. 1-8). IEEE.

[5] Varadarajan, P. and Crosby, G., 2014, March. Implementing IPsec in wireless sensor networks. In New Technologies, Mobility and Security (NTMS), 2014 6th International Conference on (pp. 1-5). IEEE.

[6] Pfeifer, T., Olariu, S. and Fersha, A., 2005. Wireless sensor networks and their applications. special issue of Computer Communications, 28.

[7] Sohraby, K., Minoli, D. and Znati, T., 2007. Wireless sensor networks: technology, protocols, and applications. John Wiley & Sons.

[8] Patel, H., Singh Rajput, D., Thippa Reddy, G., Iwendi, C., Kashif Bashir, A., & Jo, O. (2020). A review on classification of imbalanced data for wireless sensor networks. International Journal of Distributed Sensor Networks, 16(4), 1550147720916404.

[9] Priyadarshi, R., Gupta, B., & Anurag, A. (2020). Deployment techniques in wireless sensor networks: a survey, classification, challenges, and future research issues. The Journal of Supercomputing, 1-41.

[10] Jain, U., & Hussain, M. (2020). Underwater Wireless Sensor Networks. In Handbook of Computer Networks and Cyber Security (pp. 227-245). Springer, Cham.

[11] Rao, P. V., Varma, N. M. K., & Sudhakar, R. (2020). A Systematic Survey on Software-Defined Networks, Routing Protocols and Security Infrastructure for Underwater Wireless Sensor Networks (UWSNs). In Emerging Research in Data Engineering Systems and Computer Communications (pp. 551-559). Springer, Singapore.

[12] Kochhar, A., & Kaur, P. (2020). Simulation Platforms for Wireless Sensor Networks: How to Select?. In Information and Communication Technology for Sustainable Development (pp. 539-545). Springer, Singapore.

[13] Pesic, D., Radivojevic, Z., & Cvetanovic, M. (2017, May). A survey and evaluation of free and open source simulators suitable for teaching courses in wireless sensor networks. In 2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO) (pp. 895-900). IEEE.

[14] Joshi, H., & Eklarker, R. V. (2019, August). Modeling and Analysis of Energy Efficient Media Access Control Protocols for Wireless Sensor Networks Using OMNET++. In International Conference on Inventive Computation Technologies (pp. 579-586). Springer, Cham.

[15] Jain, R., 1990. The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling. John Wiley & Sons.

[16] Discrete Event Simulation: URLhttp://en.wikipedia.org/wiki/Discrete_event_simulation#Network_simulators". Accessed on 17 August, 2014.

[17] Jevtić, M., Zogović, N. and Dimić, G., 2009, November. Evaluation of wireless sensor network simulators. In Proceedings of the 17th Telecommunications Forum (TELFOR 2009), Belgrade, Serbia (pp. 1303-1306).

[18] Mekni, M. and Moulin, B., 2008, August. A survey on sensor webs simulation tools. In Sensor Technologies and Applications, 2008. SENSORCOMM'08. Second International Conference on (pp. 574-579). IEEE.

[19] Eriksson, J., 2009. Detailed simulation of heterogeneous wireless sensor networks (Doctoral dissertation, Uppsala universitet).

[20] Jevtić, M., Zogović, N. and Dimić, G., 2009, November. Evaluation of wireless sensor network simulators. In Proceedings of the 17th Telecommunications Forum (TELFOR 2009), Belgrade, Serbia (pp. 1303-1306).

[21] Mekni, M. and Moulin, B., 2008, August. A survey on sensor webs simulation tools. In Sensor Technologies and Applications, 2008. SENSORCOMM'08. Second International Conference on (pp. 574-579). IEEE.

[22] Sundani, H., Li, H., Devabhaktuni, V., Alam, M. and Bhattacharya, P., 2011. Wireless sensor network simulators a survey and comparisons. International Journal of Computer Networks, 2(5), pp.249-265.

[23] Singh, C.P., Vyas, O.P. and Tiwari, M.K., 2008, December. A survey of simulation in sensor networks. In Computational Intelligence for Modelling Control & Automation, 2008 International Conference on (pp. 867-872). IEEE.

[24] Korkalainen, M., Sallinen, M., Kärkkäinen, N. and Tukeva, P., 2009, April. Survey of wireless sensor networks simulation tools for demanding applications. In Networking and Services, 2009. ICNS'09. Fifth International Conference on (pp. 102-106). IEEE.

[25] Kawathekar, A.A. and Vatti, R., Wireless sensor Network Simulation Tools: A Survey.

[26] Nayyar, A. and Singh, R., 2015. A comprehensive review of simulation tools for wireless sensor networks (WSNs). Journal of Wireless Networking and Communications, 5(1), pp.19-47.

[27] Egea-Lopez, E., Vales-Alonso, J., Martinez-Sala, A.S., Pavon-Marino, P. and García-Haro, J., 2005, July. Simulation tools for wireless sensor networks. In Summer Simulation Multiconference, SPECTS (pp. 2-9).

[28] Musznicki, B. and Zwierzykowski, P., 2012. Survey of simulators for wireless sensor networks. International Journal of Grid and Distributed Computing, 5(3), pp.23-50.

[29] Abuarqoub, A., Hammoudeh, M., Al-Fayez, F. and Aldabbas, O., A Survey on Wireless Sensor Networks Simulation Tools and Testbeds.

[30] Mishra, V. and Jangale, S., 2014. Analysis and comparison of different network simulators. International Journal of Application or Innovation in Engineering & Management.

[31] Dhviya, V.P. and Arthi, R., 2014. Analysis of simulation tools for underwater wireless sensor networks. International Journal of Computer Science & Engineering Technology (IJCSET).

[32] Heidemann, J., Stojanovic, M. and Zorzi, M., 2012. Underwater sensor networks: applications, advances and challenges. Phil. Trans. R. Soc. A, 370(1958), pp.158-175.

[33] Das, A.P. and Thampi, S.M., 2017. Simulation tools for underwater sensor networks: a survey. Network Protocols and Algorithms, 8(4), pp.41-55.

[34] Luo, H., Wu, K., Ruby, R., Hong, F., Guo, Z. and Ni, L.M., 2017. Simulation and Experimentation Platforms for Underwater Acoustic Sensor Networks: Advancements and Challenges. ACM Computing Surveys (CSUR), 50(2), p.28.

[35] Luo, H., Wu, K., Ruby, R., Hong, F., Guo, Z., & Ni, L. M. (2017). Simulation and experimentation platforms for underwater acoustic sensor networks: Advancements and challenges. ACM Computing Surveys (CSUR), 50(2), 1-44.

[36] Jain, U., & Hussain, M. (2020). Underwater Wireless Sensor Networks. In Handbook of Computer Networks and Cyber Security (pp. 227-245). Springer, Cham.

[37] Nayyar, A., & Balas, V. E. (2019). Analysis of simulation tools for underwater sensor networks (UWSNs). In International Conference on Innovative Computing and Communications (pp. 165-180). Springer, Singapore.

[38] Peng, Z., Zhou, Z., Cui, J. H., & Shi, Z. J. (2009, October). Aqua-Net: An underwater sensor network architecture: Design, implementation, and initial testing. In OCEANS 2009 (pp. 1-8). IEEE.

[39] A. Sehgal, I. Tumar, and J. Schonwalder, "Aquatools: An underwater acoustic networking simulation toolkit," IEEE Oceans, Sydney, Australia, May 2010.

[40] Morozs, N., Gorma, W., Henson, B., Shen, L., Mitchell, P., & Zakharov, Y. (2020). Channel Modeling for Underwater Acoustic Network Simulation.