Research Article

Survey on Simulators for Underwater Sensor Networks

Geeta Godi,
School of Electronics and communication REVA University Bangalore, India. geeta.godi@gmail.com**,**
Dr. Manjula R. Bharamagoudra,
School of Electronics and communication REVA University Bangalore, India. manjula.rb@reva.edu.in,
Dr. Sunilkumar S. Manvi,
School of Computer and Information Technology REVA UniversityBangalore, India. ssmanvi@reva.edu.in

Article History: Received: 11 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 4 June 2021

Abstract: The deployment of the Underwater Sensor Network (UWSN) is extremely complex due to water characteristics. The real-time implementation of UWSN applications needs to analyze the practical consideration. To prevail these difficulties, UWSN simulation tools play a significant role especially in studying the underwater environment and also to test the new algorithms, applications, and protocols. In this paper, we explore various underwater simulators and classified them based on the programming languages. Further, we discuss on credibility and challenges posed by the simulators. This work encourages the researcher to choose an appropriate simulation tool for the UWSN application.

Keywords: UWSN, Underwater Simulators, NS2, NS3, Python based Simulators.

1. Introduction

The major resource on the earth's surface is water and it covers almost 72% of it. To achieve the best utilization of water resources, humans essentially rely on advances in the technology of an underwater environment, especially Underwater Wireless Sensor Networks (UWSN) and the Internet of Underwater Things (IoUT). These technologies have the promise of revolutionizing in the field of environmental monitoring, disaster prevention, military, surveillance applications, etc. The UWSN deploys the underwater sensor nodes in a targeted monitoring area to sense the expected parameters. The information collected by the sensor nodes will be transmitted to the surface gateways via acoustic communication further the information will be relayed to the remote monitoring center. The surface gateways are equipped with both acoustic and RF modems. IoUT is an advancement of the UWSN and IoT technologies. As it combines both UWSN with the IoT, the underwater sensors/ underwater things are digitally connected to the remote station through the internet.
However, these technologies also pose various significant challenges (Cui, 2006, Akyildiz, 2005, & Manjula, 2011) listed below.

- The transmission rate of data is severely limited by the water environment.
- The propagation delay in water is higher as compared with the terrestrial WSN.
- It suffers from a high bit error rate and network connectivity is fails temporarily in shallow zones.
- The sensors used in UWSN costs higher due to the extra protection sheath needed for sensors and the limited number of suppliers is available.
- Battery power is limited and usually, batteries cannot be recharged because solar energy cannot be exploited.
- Maintenance of underwater sensors is complex and sensors are prone to failure because of corrosion and fouling.

The study of the water environment and testing of the algorithms/implementation of underwater network structure can be done in two ways.

- **Real time implementation:** The architecture of underwater environment is realized using the hardware wherein the sensor nodes, AUVs and surface gateways are deployed in water bodies to measure the intended parameter and convey the information to the surface station.
- **Simulation:** The model related to the underwater environment is realized through simulation tools by studying their behavior based on physical characteristics. Basically, simulation enables the researcher to test their models in par with the real time implementation.

The real time implementation is extremely costly and complex to deploy a complete underwater network structure with data links (Test-bed). It is hard to validate the protocols or algorithms used in the UWSN network on the test-bed. The deployed test-bed may not be suitable for all types of applications. Repeated implementation affects the experimental results and further it is time consuming (Dhviya, & Arthi, 2014). These practical challenges demand for a simulation environment which can replicate the actual underwater scenario.

The findings of this paper will help the researchers (i) To understand UASN architecture, various simulators available in the market and their advantages and disadvantages etc. (ii) Researchers can properly choose the simulators based on the programming languages, operating system and the applications for testing and validation of the research work.

In this paper, we review the various simulators used for UWSN. We focus on the architecture, and challenges posed by UWSN. The rest of the paper is organized as follows. Section II discusses the architecture of UWSN. Section III gives the classification of simulators based on programming languages such as C++ based, JAVA based and Python-based simulator. The credibility of simulators and various challenges in simulators has been explained in section IV and V respectively. Finally, a conclusion is given in Section VI.

2. UWSN Architecture

To realize the applications of UWSN information needs to be gathered from water environment and relayed to the control center. The three types of underwater communication architectures (Akyildiz, Pompili, & Melodia, 2005) are 2D, 3D and 3D with AUV.

- **2D communication architecture:** In this architecture, all the sensor nodes are placed at the bottom of the ocean floor to sense the surrounding parameters and the communication of these nodes with the sink node takes place through acoustic mode of communication.
- **3D communication architecture:** In this architecture, sensor nodes are placed at a distinct depth (sensor nodes are floated at various depth with buoyancy) to observe the water column. The communication between the sensor nodes as well as between the sensor node and the sink node takes place through acoustic mode of communication.
- **3D communication architecture with AUV:** This communication architecture consists of static sensor nodes placed at the ocean bottom, sensor nodes at different depths, and mobile sensors such as AUV's (Autonomous Underwater Vehicle) dwelling in the water environment. The communication between these elements is performed through acoustic mode of communication.

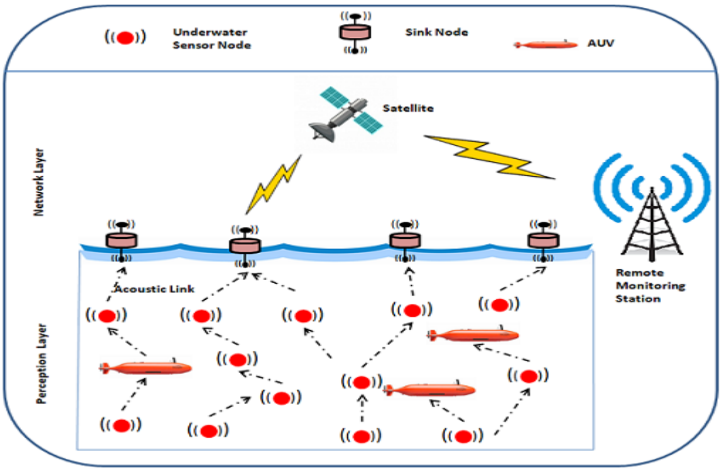

Figure 1. Architecture of UWSN

The UWSN communication architecture can be encompassed with any of the underwater architectures. In Figure 1; 3D communication architecture with AUV is depicted consists of mainly three layers: perception layer, network layer and application layer. The perception layer consists of underwater objects such as underwater sensor, autonomous underwater vehicles, sink nodes and sensor nodes. The main intention of this layer is to aggregate the underwater information. The network layer consists of a wired or wireless individually established networks, Internet, cloud computing platforms, network administration systems etc. In this layer the information acquired from the perception layer is analyzed, processed and transmits to application layer. In application layer the received information from network layer is managed and controlled by servers. The application services are implemented according to the prerequisites of end users and these services receive information from server and execute them.

3. Classification of UWSN Simulators

There are various simulators available for UWSN simulation, the selection of a suitable simulation tool as per the research prerequisites is highly essential in verification and analysis of results. The main motivation of this section is to provide awareness about the key features and performance assessment offered by various UWSN simulators that will help the researchers for choosing the suitable research platforms as per their particular prerequisites (Das & Thampi, 2016). Figure 2, depicts the classification of UWSN simulator based on the programming language/platform used to implement. The important key features and limitations of underwater simulators are listed in Table I and are also discussed below.

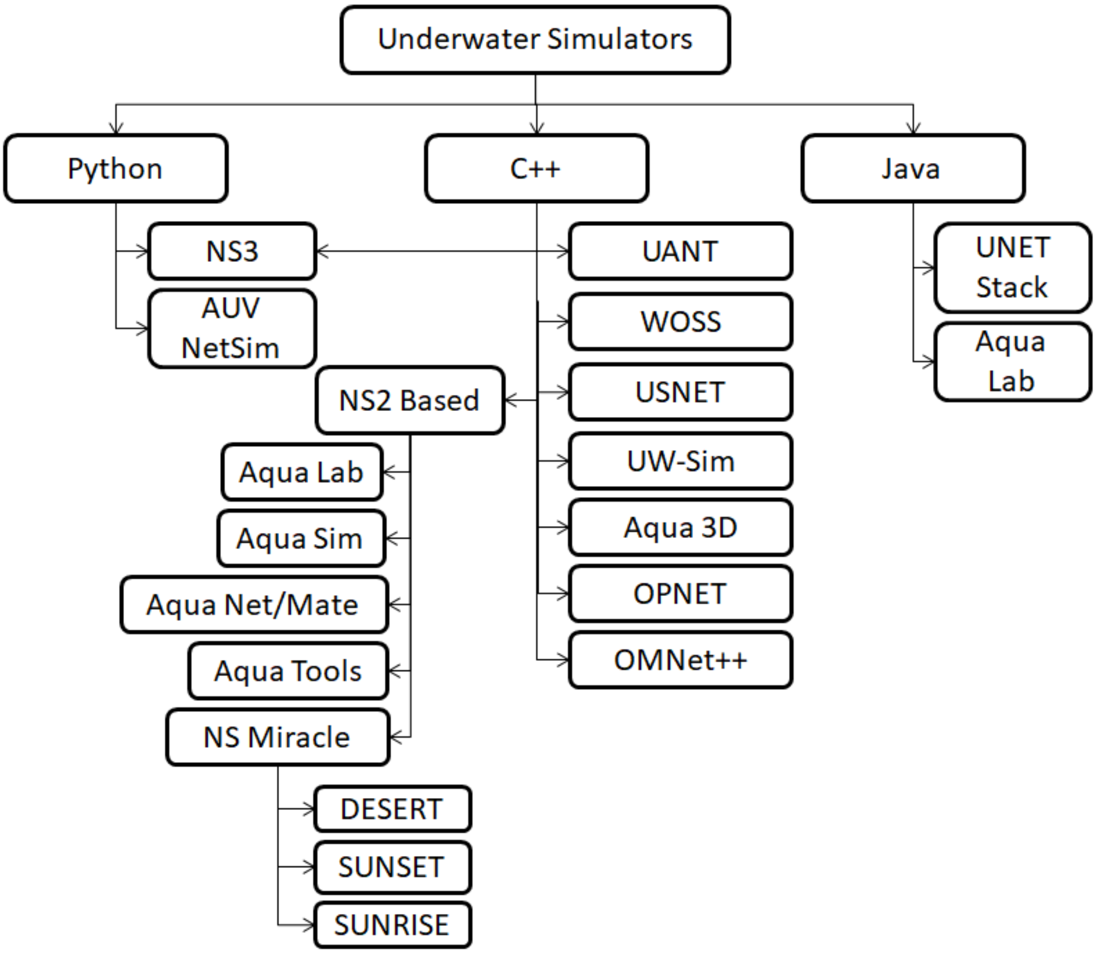

Figure 2. Classification of UWSN simulators based on programming language

3.1. C++ Based Simulators

3.1.1. UANT

Underwater Acoustic Networking platform (UANT) is developed in Networked and Embedded Systems Lab, University of California, Los Angeles in 2009 (Demirors & Emrecan, 2015). This framework provides reconfiguration flexibility for physical, MAC and application layers of UWSN. UANT provides cross-layer design flexibility to researchers by using open-source tools (Software-defined GNU radio, TinyOS and Linux OS). The cost of deployment is relatively low. It is suitable for conduction of small-scale experiments and preferred for early-stage testing. TinyOS helps the researcher to transfer the programs implemented in simulation to the real underwater sensor network.

3.1.2. WOSS

World Ocean Simulation System (WOSS) is a discrete event, an open-source simulation tool for underwater networks. It was developed in the Department of Information Engineering, University of Padova, Italy. It is designed based on NS-2 and NS Miracle to facilitate a testing environment for modeling physical layer, acoustic channel, and cross-layer protocols. WOSS is a highly flexible tool for developer’s design and implements new protocols of OSI model layers (Guerra, Casari, & Zorzi, 2009). WOSS is a multi-threaded C++ framework can be integrated into any network simulation tool that supports C++ language. WOSS is integrated with world database for the environmental parameters to analyze the real-time applications. It uses BELLHOP ray-tracing model to produce nearly real-time transmission loss that includes shadow zones.

3.1.3. USNeT

Underwater Sensor Network Simulation Tool (USNeT) is a user-friendly front-end environment, provides real time process-based simulation. It supports three-dimensional deployment. This simulator follows the object-oriented design style and all network modules are implemented using C++ encapsulating thread mechanisms. To handle multiple tasks in USNeT framework, threads have been used (Ovaliadisx, & Savage, 2013). It is simple to use since it has interactive GUI. It supports 3D capabilities to form and view better of the cluster-based network topology. Simulations can be run for longer duration in UsNeT.

3.1.4. UWSim

Underwater Simulator (UWSim) is an open-source simulator (Dhurandher, Misra, Obaidat, & Khairwal, 2008) implemented using C++ and primarily designed for Underwater Sensor Network. UWSim mainly focuses on underwater communication performance parameters such as low-bandwidth, low frequency, high transmission, and limited memory. It is based on component-based approaches than a layer/protocol-based approach. It supports various simulated sensors for researchers to integrate with simulation environment such as camera, range sensor, object picker, pressure, DVL, IMU, GPS, force and structured light projector, etc.

3.1.5. Aqua-3D

Aqua-3D is animator/visualization tool for underwater sensor network, developed by the Department of Computer Science in the University of Connecticut, Storrs. To install the Aqua-3D an external graphical interface card (NVIDIA or ATI) is required. It is capable to read trace files generated by UWSN simulators and animate the simulations in 3D graphics (Tran, Zuba, Le, Zhu, Peng, & Cui, 2012). Aqua-3D has the interactive GUI that can be customized with various options such as adjusting the camera angles, drawing the lines between the nodes (transmission paths), adjust the network topology, trace files, etc.

3.1.6. OPNET

Optimized network engineering tool (OPNET) is a discrete event driven simulation tool, implemented using C++ language. It provides the access to different models of protocols as well as allows the researcher to create and simulate the different network protocols. It also allows the researcher to implement the new algorithm, modify the existing model and also analyze the network protocols (Chang, 1999). OPNET is GUI based framework hence debugging and analysis are easier. It can be used for research as well as for network design and analysis purpose. It supports both 32-bit and 64-bit parallel simulation kernel.

3.1.7. NS2-2 Based Simulators:

Network Simulator 2 (NS-2) developed in 1989 by California and Cornell University is an advanced version of REAL network simulator. It is an open-source, discrete event network simulator. Basically, the core of NS-2 is implemented in C++, with Object-Tcl (OTcl) based scripting (NS, 2021). The compatible operating system for NS-2 is Linux and it can be run on the Windows platform with the Cygwin package. It is designed for analyzing the performance of all sort of networking research works. NS-2 supports a large volume of protocols in various layers. The object-oriented feature allows the user to design, test and implement new protocols in the protocol stack. To observe the network parameters (such as channel link, mobility of nodes, the capacity of a network, etc.) NS-2 provides a GUI tool called Network Animator (NAM).

3.1.8. Aqua-Lab

The Aqua-Lab is a low-cost testbed developed by the University of Connecticut in 2007. The simulator primarily works on the physical layer of the OSI model. It provides low-cost configurable environments by using real modems. It helps the beginners to set up a lab-level testbed and initially, a developer can be familiar with simple underwater acoustic communication concept (Luo, Wu, Ruby, Hong, Guo, & Ni, 2017). Aqua-lab is powerful to design and implement novel approaches and distinguish them with the existing approaches in UWSN.

3.1.9. Aqua-Sim

Aqua-Sim is a discrete event-driven, NS-2 based open-source simulator. It follows OOPS concepts and the hence entities are implemented as classes. It was developed in the underwater sensor networks lab, University of Connecticut, Storrs in 2009. Aqua-Sim works on different layers of ISO Model such as Physical, MAC and Routing layers and it is suitable to implement complete protocol stack. Aqua-Sim efficiently simulates the attenuation of channels and the packet collision parameters in acoustic underwater networks (Raj, & Sukumaran, 2015). Aqua-Sim provides highly accurate experimental results when compared with the real-time implementation. Aqua-sim is purely NS-2 based open-source simulator. It supports both 3D and 2D communication architecture of UWSN. The deployment cost of Aqua-sim testbed is considerably low (Nayyar & Balas, 2019).

3.1.10. Aqua-Net and Aqua-NetMate

Aqua-Net and Aqua-Net Mate are developed in the Department of computer science engineering, University of Connecticut, Storrs. Aqua-Net is suitable for embedded systems but it does not support for simulation mode (Peng, Zhou, Cui, & Shi, 2009). It supports BSD Socket to interface the user applications and various UWSN protocols like ALOHA, VBF routing, etc. Aqua-Net Mate works as simulator as well as the emulator. It is transparent to Aqua-Net and consists of mainly two components such as Aqua-net adapter/driver: converts packets from Aqua-Net to Aqua-Net Mate and vice-versa and Virtual modem: provides an interface and a method to transmit and receive the packets from Aqua-net adapter. It supports all the layers of the OSI Model. Aqua-Net Mate can ensure the real-time features since all events are real-time scheduler and each layer runs as single process (Zhu, Le, Pu, Lu, Peng, Cui, & Zuba, 2013).

3.1.11. Aqua-Tool

Aqua tool is an underwater acoustic sensor networking simulation tool kit. It was designed in the Department of computer science, Jacob University, Bremen. It is mainly targeted to evaluate the networking aspects of underwater acoustic communication. The acoustic channel models support for significant parameters like acidity, temperature, salinity, and depth of nodes (Sehgal, Tumar, & Schönwälder,. 2010). It supports the simulation of both static and mobile nodes of the acoustic communication network. It has been used to develop a power management scheme for underwater acoustic networks. It also provides the flexible scripting interface to set up the simulator.

3.1.12. NS-Miracle Based Simulators

NS Miracle is a Multi-Interface Cross-Layer extension for NS or simply an NS-2 library developed to facilitate advanced networks such as 4G, UWSN, sea ad-hoc. It was designed in SIGNET Lab at the University of Padova. NS Miracle operates in all the layers of the protocol stack (Peng, et al., 2009). It is expected to help the researcher to design and implement the cross-layer protocols. NS Miracle defines a novel approach to produce a multi-layer architecture with cross-layer communication functionalities. It also improves the performance of NS-2. The SUNSET, SUNRISE and DESERT simulators are developed based on NS miracle.

3.1.13. SUNSET

SUNSET (Luo, et al., 2017) is an open-source platform developed by NS-2 and along with its extension NS-2 Miracle by Sapienza University in 2012. It is a toolkit for implementation and testing of underwater network protocols. This framework provides underwater simulation, emulation, and real-time field testing. During the simulation, SUNSET uses different acoustic channel models to evaluate the protocol stack and the implementation of routing protocols. SUNSET can be interfaced with real external hardware in emulation mode. SUNSET was selected as a standard testbed in SUNRISE, which was extensively experimented over a period of four years. It is entirely working to test underwater acoustic devices (such as modems, AUV's, surface nodes, sink nodes and other environmental sensors) (Nayyar, et al., 2019).

3.1.14. DESERT

DESERT (Masiero, et al., 2012) is NS Miracle based framework simulator and emulator testbed for underwater sensor network protocols. It was designed using C/C++ by the University of Padova, Italy in 2012. The libraries of C/C++ help to design and implement all the layers of the OSI model. It is suitable for the implementation and testing of new protocols for UWSN. It uses same code for single-hop and multi-hop transmissions (Nayyar, et al., 2019). It has different modules for evaluation of node mobility parameter in 2D and 3D scenarios.

3.1.15. SUNRISE

SUNRISE is an environment-based test bed for UWSN developed in the Department of Information in La Sapienze University, Italy. It provides an advanced interface over SUNSET to design, test, and compare protocols for UWSN. It supports a wide range of MAC, routing, and cross-layer protocols for UWSN (Petrioli, et al., 2014). SUNRISE is a software-defined open architecture modem and protocol stack that will guide open collaborative developments. It supports application domains, data analysis, and scalability. It allows seamless integration of additional platforms and devices made available by partners or third parties. SUNRISE supports security, privacy and trust by providing an environment in which solutions for underwater security can be developed and tested (Raj, et al., 2015).

3.2. JAVA Based Simulators

3.2.1. UnetStack

UnetStack (Montana, 2008) is an agent-based software stack and simulator for underwater networks. It is used in discrete event simulation mode to simulate and test the performance of underwater network protocols and in real time simulation mode, it can be used to debug protocol implementation and test the deployment scenario before an experiment. It supports implementation on portable platform Java Virtual Machine (JVM) and allows the researcher to develop software using Groovy script. Groovy is a dynamic language for JVM. It builds upon the strength of JAVA and also supports additional powerful features inspired by other languages such as Python, Ruby and smalltalk. It is easy to install, implement and test the protocols and deploy the protocols on UnetStack compatible modem without recompilation. based on java.

3.2.2. Aqua-Lab

It provides the researcher to experience field test in lab-controlled environment. It is intended to test the performance of underwater network protocols through simulation environment. The Aqua-lab software (Peng, Cui, Wang, Ball, & Freitag, 2007) mainly consists of : Low level, high level libraries and an emulator. The low-level library interfaces between the end user and the modem. The high-level library is built on low level libraries to provide socket like programming. Emulator helps to interface physical underwater environment parameters. It has web-based user-friendly graphical user interface (GUI), which is implemented in PHP/Ajax/XML/ Java-scipt and uses Ajax library.

3.2. Python Based Simulators

3.3.1. NS-3

Network simulator-3 (NS, 2021), (Thumpi, Manjula, & Sunilkumar, 2013) & (Bharamagoudra, & Manvi, 2017). is an open source, discrete-event network simulator mainly used for research and educational purpose. NS-3 is entirely written in C++ language with optional python scripting. It is not backwards-compatible with NS-2. It is a completely new simulator with integrated visualization package Netanim and it also supports external animators, data analyzers and visualization tools. NS-3 allows users to run real implementation code in the simulator. NS-3 has innovative and useful features such as scalable, flexible, clean design and real time integration. It provides an open environment for researchers to contribute and share their software modules. Users can work at command line with C++ and / or python languages. NS-3 is compatible for the Linux operating system, Mac OS and windows with Cygwin package.

3.3.2. AUVNetSim

AUVNetSim (Montana, 2008) is a simulation library written in Python for testing acoustic networking algorithms. A discrete event simulation package, SimPy is used in this simulator. AUVNetSim provides great flexibility for developers to run the simulations and modify system parameters without having to explicitly deal with Python code. Developer can implement new protocols by taking the advantage of the existing structure. The required prerequisite library package software is freely available under GNU license.

4. Credibility of Simulators

The Table I gives the details of various simulators with its features such as supported programming language, specification, user interface, availability etc. and limitations. Simulation has confirmed to be a valued tool in lots of regions wherein analytical strategies aren't relevant and experimentation isn't always feasible. Researchers commonly use simulation to investigate device overall performance prior to physical design and implementation or to evaluate multiple alternatives over an extensive variety of conditions (Zia, Otero, Siddiqui, & Poncela, 2021). The errors in the simulation models or inconsistent data or facts analysis produces incorrect or misleading results. Even though there exist various simulation tools for UWSN it is impractical to have all-in-one integrated tool that simultaneously support simulation, emulation and test -bed implementations. Each simulation tool exhibits its own features, advantages and limitations as given in table I. These UWSN simulators are most effective and suitable in some unique conditions. So, the researchers have to choose simulators carefully that is best suited for the intended application.

Table 1. Features and Limitations of Underwater simulators

Simulator Name	Features					Limitations
	Programming Language	***Specification Support***	***Emulation Support***	***GUI Interface***	***Open Source***	
NS-2 (NS, 2021)	C, C++, oTcl	IEEE802.15.4, IEEE802.11, IEEE802.3	No	Simple	Yes	Does not support GUI. Complicated scripting language
Aqua Lab (Luo, et al., 2017)	NS-2, PHP, Java, Ajax	IEEE805.11	No	Good	NA	Does not support complex simulation environment
Aqua-Sim (Raj. et al., 2015)	NS-2	IEEE802.15.4	No	Good	Yes	Advanced channel models with real experiments data required for simulation

Aqua-net/mate (Peng, et al., 2009 & Zhu, et al., 2013),	NS-2	IEEE802.11	Yes	Simple	Yes	Allows only one protocol suite per modem
Aqua Tools (Sehgal, et al., 2010)	NS-2	IEEE805.11	NA	Good	Yes	Allows only one protocol suite per modem
NS-Miracle Baldo, et al., 2010)	C, C++, oTcl	IEEE802.15.4	No	Good	Yes	Node depth affects the channel performance
SUNSET (Petrioli, et al., 2013)	NS-2, NS-Miracle	IEEE802.11	Yes	Good	Yes	Difficult to achive time synchronization and also to identify the accuracy-related problems
DESERT (Masiero, et al., 2012)	C, C++, NS-2, NS-Miracle	IEEE802.15.4	Yes	Good	Yes	Less efficient solution in real time applications compared to SUNSET
SUNRISE (Petrioli, et al., 2014)	NS-2, NS Miracle	IEEE802.11	Yes	Good	Yes	Limited battery life for the Underwater Robotic applications
UANT (Demirors et al., 2015)	C++,Tiny OS,TOSSIM	IEEE805.11	Yes	Simple	Yes	It requires the prerequisite library packages (SimPy, Numpy, MatplotLib) before installing and running
WOSS (Guerra et al.,2009)	C++	IEEE802.15.4	No	Good	Yes	Currently Multi-path Propagation is not considered
USNeT (Ovaliadis et al., 2013)	C	NA	No	Simple	NA	Simulation and emulation results does not match accurately
UwSim (Dhurandher et al., 2008)	C#,.NET framework	NA	Underwater robotic research	Good	Yes	Supports limited functionalities
Aqua3D (Tran et al., 2012)	C++	NA	NA	Simple	Yes	Tested with only Aqua-sim trace files, Need to check with other simulators

OPNET (Chang, 1999)	C, C++	IEEE805.11	No	Good	Yes	Suitable for real time implementation

5. Challenges in Simulators

Simulation is a fascinating process for assessing the overall performance of UWSN protocols because designers can use it to assess analytical frameworks. However, the simulation environment must consider real-time perspectives to obtain the results. Otherwise, it can't give accurate results with the working attributes of the framework the developers are studying. Standardization and lack of accuracy can lead to inaccurate data, which can result in incorrect conclusions or inappropriate implementation decisions.
Some of the common challenges faced by the simulators are (i) The constant factors used to set the network environment may cause inaccuracies in simulation. (ii) It is difficult for the developer what parameter need to vary, how much to vary and which combination of parameters to consider.

6. Conclusion

The model related to the underwater environment is realized through simulation tools by studying their behavior based on physical characteristics. Simulation enables the researcher to test their models in par with real-time implementation. This paper mainly discusses various UWSN simulators, their classification based on the programming languages, and credibility of simulators. Further, the research challenges and comparison of UWSN simulation tools are presented to enable the researchers to select the appropriate tools suitable for the defined environment to test and validate the proposed model.

References

1. Cui, J. H., Kong, J., Gerla, M., & Zhou, S. (2006). The challenges of building mobile underwater wireless networks for aquatic applications. *IEEE Network*, 20(3), 12-18.
2. Akyildiz, I. F., Pompili, D., & Melodia, T. (2005). Underwater acoustic sensor networks: research challenges. *Ad hoc networks*, 3(3), 257-279.
3. Manjula, R. B., & Manvi, S. S. (2011). Issues in underwater acoustic sensor networks. *International Journal of Computer and Electrical Engineering*, 3(1), 101.
4. Dhviya, V. P., & Arthi, R. (2014). Analysis of simulation tools for underwater wireless sensor networks. *International Journal of Computer Science & Engineering Technology (IJCSET).*
5. Das, A. P., & Thampi, S. M. (2016). Simulation tools for underwater sensor networks: a survey. *Network protocols and Algorithms*, 8(4).
6. Demirors and Emrecan, (2015). SEANet: A software-defined acoustic networking framework for reconfigurable underwater networking, *Proceedings of the 10th International Conference on Underwater Networks & Systems*.
7. Guerra, F., Casari, P., & Zorzi, M. (2009, November). World ocean simulation system (WOSS) a simulation tool for underwater networks with realistic propagation modeling. *In Proceedings of the Fourth ACM International Workshop on UnderWater Networks* (pp. 1-8).
8. Ovaliadis, K., & Savage, N. (2013). Underwater sensor network simulation tool (USNeT). *International Journal of Computer Applications*, 71(22).
9. Dhurandher, S. K., Misra, S., Obaidat, M. S., & Khairwal, S. (2008). UWSim: A simulator for underwater sensor networks. *Simulation*, 84(7), 327-338.
10. Tran, M., Zuba, M., Le, S., Zhu, Y., Peng, Z., & Cui, J. H. (2012, October). Aqua-3D: An underwater network animator. *In* 2012 Oceans (pp. 1-5). IEEE.
11. Chang, X. (1999, December). Network simulations with OPNET. *In WSC'99. 1999 Winter Simulation Conference Proceedings*. 'Simulation-A Bridge to the Future'(Cat. No. 99CH37038) (Vol. 1, pp. 307-314). IEEE.
12. NS-3[2021], *A Discrete Event Network Simulator*, retrieved from https://www.nsnam.org/docs/tutorial/html/

13. 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.
14. Raj, C., & Sukumaran, R. (2015). Modeling UWSN simulators–a taxonomy. *International Journal of Computer and Information Engineering*, 9(2), 585-592.
15. 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.
16. 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.
17. Zhu, Y., Le, S., Pu, L., Lu, X., Peng, Z., Cui, J. H., & Zuba, M. (2013, June). Aqua-Net Mate: A real-time virtual channel/modem simulator for Aqua-Net. *In 2013 MTS/IEEE OCEANS-Bergen* (pp. 1-6). IEEE.
18. Sehgal, A., Tumar, I., & Schönwälder, J. (2010, May). AquaTools: An underwater acoustic networking simulation toolkit. *In IEEE, Oceans*.
19. Baldo, N., Miozzo, M., Guerra, F., Rossi, M., & Zorzi, M. (2010). Miracle: the multi-interface cross-layer extension of ns2. *EURASIP Journal on Wireless Communications and Networking, 2010*, 1-16.
20. Petrioli, C., Petroccia, R., & Spaccini, D. (2013, November). SUNSET version 2.0: Enhanced framework for simulation, emulation and real-life testing of underwater wireless sensor networks. *In Proceedings of the Eighth ACM International Conference on Underwater Networks and Systems* (pp. 1-8).
21. Masiero, R., Azad, S., Favaro, F., Petrani, M., Toso, G., Guerra, F., ... & Zorzi, M. (2012, May). DESERT Underwater: an NS-Miracle-based framework to DEsign, Simulate, Emulate and Realize Test-beds for Underwater network protocols. *In 2012 Oceans-Yeosu* (pp. 1-10). IEEE.
22. Petrioli, C., Petroccia, R., Spaccini, D., Vitaletti, A., Arzilli, T., Lamanna, D., ... & Renzi, E. (2014, September). The SUNRISE GATE: Accessing the SUNRISE federation of facilities to test solutions for the Internet of Underwater Things. *In 2014 Underwater Communications and Networking (UComms)* (pp. 1-4). IEEE.
23. Peng, Z., Cui, J. H., Wang, B., Ball, K., & Freitag, L. (2007, September). An underwater network testbed: design, implementation and measurement. *In Proceedings of the second workshop on Underwater networks* (pp. 65-72).
24. Thumpi, R., Manjula, R. B., & Sunilkumar, S. M. (2013). A survey on routing protocols for underwater acoustic sensor networks. *Int. J. Recent Technol. Eng.(IJRTE)*, 2(2), 170-175.
25. Bharamagoudra, M. R., & Manvi, S. S. (2017). Agent-based secure routing for underwater acoustic sensor networks. *International Journal of Communication Systems*, 30(13), e3281.
26. Montana, J. M. J. (2008). AUVNetSim: A simulator for underwater acoustics networks. *Massachusetts Institute of Technology. Sea Grant College Program*.
27. Chitre, M., Bhatnagar, R., & Soh, W. S. (2014, September). UnetStack: An agent-based software stack and simulator for underwater networks. *In 2014 Oceans-St. John's* (pp. 1-10). IEEE.
28. Fang, S., Da Xu, L., Zhu, Y., Ahati, J., Pei, H., Yan, J., & Liu, Z. (2014). An integrated system for regional environmental monitoring and management based on internet of things. *IEEE Transactions on Industrial Informatics*, 10(2), 1596-1605.
29. Zhou, Z., Yao, B., Xing, R., Shu, L., & Bu, S. (2015). E-CARP: An energy efficient routing protocol for UWSNs in the internet of underwater things. *IEEE Sensors Journal*, 16(11), 4072-4082.
30. Zia, M. Y. I., Otero, P., Siddiqui, A., & Poncela, J. (2021). Design of a web based underwater acoustic communication testbed and simulation platform. *Wireless Personal Communications*, 116(2), 1171-1193.