Exploring Simulation of Software-Defined Underwater Wireless Networks

Li Wei
Michigan Technological University
Houghton, Michigan
liwei@mtu.edu

Yuxin Tang
Shanghai Jiao Tong University
Shanghai, China
wdb_jd@sjtu.edu.cn

Yuching Cao
University of California, Los Angeles
Los Angeles, California
yuching.cao@engineering.ucla.edu

Zhaohui Wang
Michigan Technological University
Houghton, Michigan
zhaohuiw@mtu.edu

Mario Gerla
University of California, Los Angeles
Los Angeles, California
gerla@cs.ucla.edu

ABSTRACT

Multi-modal communication methods have been proposed for underwater wireless networks (UWNs) to tackle the challenging physical characteristics of underwater wireless channels. These include the use of acoustic and optic technology for range-dependent transmissions. Software-defined networking (SDN) is an appealing choice for managing these networks with multi-modal communication capabilities, allowing for increased adaptability in the UWN design. In this work, we develop a simulation platform for software-defined underwater wireless networks (SDUWNs). Similar to OpenNet, this platform integrates Mininet with ns-3 via TapBridge modules. The multi-modal communication is implemented by equipping each ns-3 node with multiple net devices. Multiple channel modules connecting corresponding net devices are configured to reflect the channel characteristics. The proposed simulation platform is validated in a case study for oceanographic data collection.

CCS CONCEPTS

• **Networks** → Network performance evaluation;

KEYWORDS

Software-Defined Networks, Underwater Wireless Networks, Simulation Platform

ACM Reference Format:
Li Wei, Yuxin Tang, Yuching Cao, Zhaohui Wang, and Mario Gerla. 2017. Exploring Simulation of Software-Defined Underwater Wireless Networks. In *WUWNET'17: WUWNET'17: International Conference on Underwater Networks & Systems, November 6–8, 2017, Halifax, NS, Canada.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3148675.3148720

1 INTRODUCTION

Underwater wireless networks (UWNs) have drawn much attention for their extensive applications in scientific research, ocean environment monitoring, offshore drilling, and naval operations. A UWN system may consist of heterogeneous nodes with an on-shore or off-shore control center, and off-shore surface or underwater nodes that are mobile or stationary. Moreover, the off-shore mobile nodes, especially those representing autonomous or unmanned underwater vehicles (AUV/UUV) and unmanned surface vehicles (USV), are playing an increasingly important role in marine data collection operations such as underwater searching and ocean floor mapping. With a capable UWN system, mobile nodes will be able to share timely information, make collaborative decisions, and enhance the efficiency of operations by modifying their trajectories in real time.

Since wireless communications in radio frequencies cannot work effectively in underwater environments, novel underwater wireless communication techniques, i.e. magneto-inductive, acoustic, and optical communications, have been developed to provide a means of wireless communication among UWN nodes. Due to the challenging channel characteristics, practical underwater wireless devices are usually only capable of either long-range/low-data-rate transmissions (e.g. several Kbps over several kilometers in acoustic communications [16], [2]) or short-range/high-data-rate transmissions (e.g. several Mbps over several meters using magnetic inductive, optical or ultra-sonic communications [2], [12]). Considering the high cost of underwater equipment, it is more efficient to equip UWN nodes with multiple types of communication devices and deploy nodes sparsely in a vast underwater area [2].

In addition, long-range underwater acoustic communication suffers from long propagation delays and limited bandwidth of acoustic signals. The unpredictable environment also results in intermittent connectivity for most underwater communication techniques. To account for those factors, the system architecture of a UWN node should be highly flexible to support novel communication devices and networking protocol designs. In the last two decades, a plethora of UWN-oriented protocols and system architecture have been proposed [16], [3], [11]. The hierarchical architectures proposed in [3], [11], [10] allow the user to install multiple different protocols at each layer of the protocol stack to coordinate multiple UWN communication devices. Sealinx [10] also implements a core module, thereby supporting more comprehensive protocol designs across network layers.

WUWNET'17, November 6–8, 2017, Halifax, NS, Canada

ACM ISBN 978-1-4503-5561-2/17/11…$15.00
https://doi.org/10.1145/3148675.3148720

Due to the current abstractions of UWN protocol stack layers, cross-layer design of a protocol is still fairly limited. It can currently be accomplished by implementing either multiple interactive modules at different layers, or a single protocol module with functions interacting with different layers. Given its flexible and virtualizable nature, SDN provides an ideal platform for implementing underwater communications technologies [2]. In addition to pioneering work in software-defined underwater acoustic communications [5], [13], several SDUWN architectures have been recently proposed [2], [6]. The separation of the control plane from the data plane and an abstracted network representation are key features that make SDUWN a promising design for future UWN systems [2].

In this paper, we implement an SDUWN simulation platform based on OpenNet [4] and ns-3 underwater acoustic network (UAN) module[7]. To validate the simulation platform, a case study of using an SDUWN system to monitor the ocean current is discussed. The OpenFlow protocol is employed to configure the routing topology of the SDUWN system.

The rest of this paper is organized as follows. Related works about SDN simulations are reviewed in Section II. Section III focus on the design of the SDUWN simulation. The ocean current monitoring case study is discussed in Section IV. Finally, Section V concludes our work, and Section VI discusses the future possibilities in the field.

2 RELATED WORKS

As the next-generation networking paradigm, SDN aims to improve network resource utilization, reduce network management cost, increase interoperability between heterogeneous devices, and accelerate innovation and evolution in the UWN field[2]. A typical SDN system consists of network nodes such as switches with open and standardized interfaces and a network controller which defines the behavior and operation of those switches[2]. A data plane exists among the nodes for processing packets, with the control plane consisting of a central controller configuring the behavior of the nodes and connections between the nodes.

By using a lightweight OS-level virtualization approach, Mininet provides a rapid solution for prototyping SDN systems[9]. However, both the control plane and data plane in Mininet are assumed to be wired networks. In [14], OPNET is employed to simulate a multi-hop wireless network using SDN to control routing configurations.

Some SDN simulators integrate multiple software packages to simulate both the network and the physical channel. For example, OpenNet[4] uses Mininet for the control/data layers combined with ns-3 to simulate the physical layer of a Wi-Fi channel. It does so by connecting an ns-3 Tap Bridge module with a Tap device created in Mininet. In [8], application modules that mimic SDN switches supporting OpenFlow 1.0 were developed, and the Direct Code Execution (DCE) ns-3 module was used to connect ns-3 with the POX SDN controller. The DCE SDN framework proposed in [8] performed better than OpenNet in terms of memory usage and real-time performance. However, OpenNet is more flexible in running applications and controllers since Mininet runs terminals in each host node's namespace with the same kernel used by operating system.

In this article, we implement a simulation platform for SDUWN using OpenNet and the ns-3 UAN module. The underwater wireless channels for both the control plane and data plane networks are simulated by ns-3. SDN switches and controllers are emulated in Mininet, and are connected to ns-3 nodes with UAN net devices via Tap devices and Tap bridges.

3 A SIMULATION SYSTEM FOR SDUWN

This section will focus on design of the proposed SDUWN simulation system. The overall architecture and motivations for detailed implementations will be introduced in this section.

The architecture of the proposed SDUWN system is shown in Fig. 1. The framework is similar to OpenNet, as it uses Tap bridge modules to interface Mininet with ns-3. Applications run on host nodes of Mininet with all capabilities of the operating system which runs the simulation. Data packages will be processed on switch nodes of Mininet following the employed protocols with the help of Open vSwitch. The SDN controller can run on a host node of Mininet which connects to Tap bridges of a ns-3 node. The physical status of the SDUWN nodes, i.e. locations, available energy and communication capabilities, are simulated in ns-3, as well as the communication channels. When the network is constructed, a SDUWN node will be simulated as the combination of a host node with a switch node of Mininet and a ns-3 node. The host node and switch node in Mininet are connected with negligible delay and zero error rate. A packet sent out from a specific port of switch can be relay to a net device of ns-3 node via a Tap bridge module and finally transmitted to the wireless channel simulated in ns-3. Then, the packets are transmitted via ns-3 channels and received by a corresponding ns-3 node. The received packets are sent up via the Tap Bridge and handled by network programs in Mininet.

Each node in our system consists of a host-and-switch pair in Mininet, as well as a matching node in ns-3. Similar to common adhoc network nodes, most SDUWN nodes have full network function capabilities.

On the Mininet side, hosts run the applications, which allows programs developed for the simulation platform to be conveniently migrated to other hardware platforms. The switches (e.g. Open vSwitch) handle interactions and protocols between each node and the rest of the SDN. The multi-modal communication devices of a host are simulated by corresponding Tap devices on a switch, and each device can be accessed by the host as a network port.

Each ns-3 node can possess multiple net devices. Physical states such as location coordinates, mobility patterns, and battery, are simulated by ns-3 modules. Long-range low-data-rate and short-range high-data-rate devices can be simulated as different net devices connected to corresponding channel modules in ns-3. Using other wireless models provided by ns-3, it is possible to build a hybrid network with SDUWN nodes connected to surface nodes, which can communicate via Wi-Fi or other RF networks.

Since OpenNet was originally designed for Wi-Fi networks, it is reasonable to assume that there is a reliable out-of-band control plane network available for OpenFlow packets [4]. Thus, control plane connections between an SDN controller and switches do not

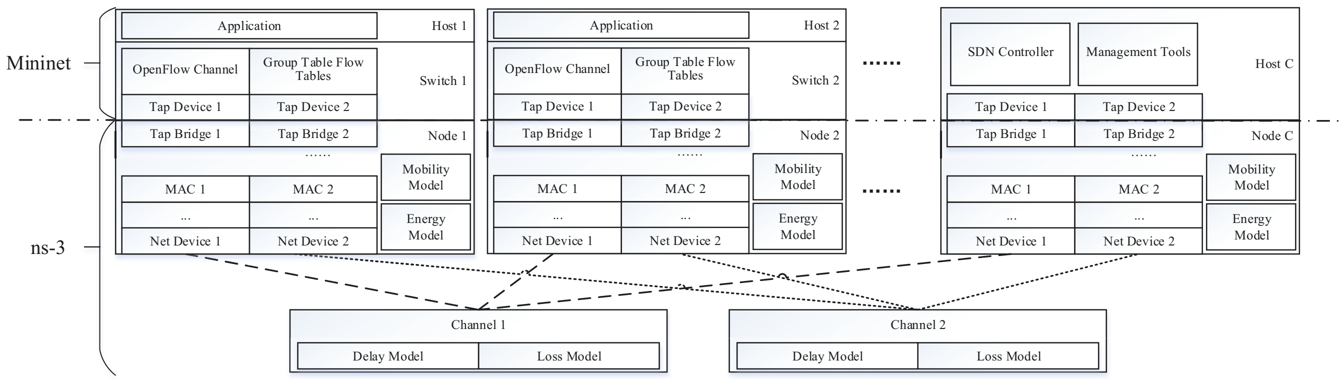

Figure 1: Simulation of SDUWN with Mininet and ns-3. Channel 1 and Channel 2 can, respectively, model a long-range low-data-rate and a short-range high-data-rate underwater channel

pass through channels simulated in ns-3. However, a real SDUWN system will need to employ a control plane network which is similar to the hybrid in-band and out-of-band control proposed in SoftWater [2]. As a result, control plane network will also suffer from long delays and intermittent connectivity and needs to be simulated in ns-3. To simulate the control plane, the SDN controller runs on a Mininet host, and Open vSwitches in Mininet are configured with the in-band control mode. The control host connects to an ns-3 node via a Tap connection, and all packets between the controller and switches then travel through the ns-3 channels.

Existing SDN controllers need to be modified to fulfill SDUWN requirements and adapt to the long delay and unreliable control plane channel characteristics. At least the TCP timeouts need to be changed to establish the OpenFlow connection between SDN controller and each SDUWN node.

Since Mininet links are primarily designed for wired connections, OpenNet implemented segment classes for different types of wireless networks in Mininet[9]. For each segment class, there exists a corresponding channel helper module created in ns-3. The ns-3 channel characteristics, MAC settings and physical layer parameters of net devices are all configured by the segment class. A network interface between a Tap bridge with ns-3 is installed by adding hosts or switches to a segment in Mininet. Thus, wireless links may not show up when running the net command in Mininet. Multiple segment classes constructed in a Mininet script will construct multiple independent channel modules in ns-3. Collisions and interferences only happen between packets transmitted in the same channel module.

4 A CASE STUDY

To test our simulation platform and explore challenges of SDUWN protocol design, a simple example is studied in this section to illustrate challenges of using OpenFlow 1.0 in SDUWN.

4.1 SDUWN Application Scenario Example

The front in oceanography refers to the boundary between two distinct water masses moving in different directions [1]. Usually, a conductivity temperature depth (CTD) sensor and an acoustic Doppler current profiler (ADCP) will be equipped on each node to measure the oceanographic data of the front area. Data measured by an ADCP is a distribution graph of current velocity, which can be as large as MB files, while the CTD data usually vary on a far slower timescale and have a lower level of variety than ADCP data.

Assume that a SDUWN system is deployed to measure the current front. Each node is equipped with one modem for long-range low-data-rate communications, and another for short-range high-data-rate communications. Then, there could be a controller node equipped with a long-range low-data-rate modem that runs POX to configure the SDN switch on each SDUWN node. The long-range low-data-rate acoustic modems can establish one-hop control plane communications between the controller node and any of the SDUWN nodes. The short-range high-data-rate acoustic modems can be used to establish data plane communications to transmit data generated by sensors on each node.

Based on the measurements from [15], [17], we assume that the long-range low-data-rate acoustic modems communicate with a center frequency of 24KHz, 6KHz of bandwidth, a throughput up to 10Kbps, and a communication range within 1.5Km [15]. The short-range high-data-rate acoustic modem can communicate at 1Mbps within 100m [17]. Since the modulation algorithms for both acoustic modems are not available, our simulation models them as ns-3 UAN net devices with equivalent communication ranges, center frequencies, bandwidths, and data rates. Thus, interferences and collisions can be simulated in the experiments.

4.2 Running Original OpenFlow 1.0 in SDUWN

To explore challenges of SDUWN protocol design, we first try to run the original OpenFlow 1.0 with the aforementioned SDUWN configurations. Since controller is using underwater communication for control plane message transmission, the estimated Round Trip Time (RTT) for the controller to a node 100m away can be calculated as $RTT = 2T_{prop} + T_{tx1} + T_{tx2} + T_{proc}$, where T_{prop} denotes the propagation delay, which is about 66.67ms in this example. T_{tx1} and T_{tx2} are transmission delays for packet sent and received, respectively. Taking the 74B long OFPT_HELLO message as an example, $T_{tx} = 74B/10Kbps = 59.2ms$, while for a 242B OFPT_FEATURES_REPLY message which is also a necessary message for the handshaking process T_{tx} can also be as long as 193.6ms

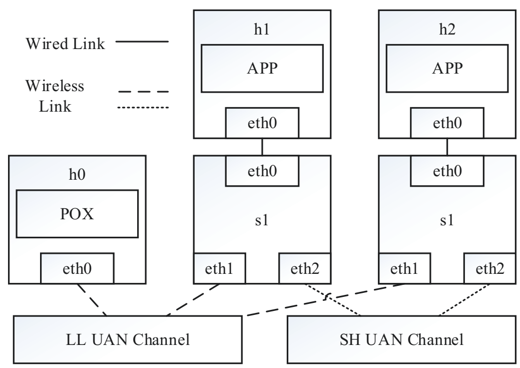

Figure 2: A simple example of SDUWN. Two SDUWN nodes (h1-s1 and h2-s2) are equipped with both Short-range High-data-rate underwater acoustic modem and Long-range Low-data-rate underwater acoustic modem. The controller is on node h0 equipped with long-range low-data-rate underwater acoustic modem. The SDN switches on 2 nodes are configured by the POX running on h0. Application program is running on h1 and h2.

. Even with the assumption that the processing delay denoted by Tproc is negligible, the RTT is surely larger than 200ms for SDUWN control plane. Since the original control plane of OpenFlow is designed to work in secured communication based on standard TCP protocol. Regardless the intermittent connectivity, the long delay for underwater communication will make handshaking between the controller and a switch hardly succeed. We used a ns-3 CSMA channel to replace the long-range low-data-rate channel and tried different delay values. The simulation results show that OpenFlow connection between the controller with a single switch cannot be established if RTT is more than 400ms.

An example of configuration time for RTT equal to 0, 100ms, 200ms and 300ms is shown in Fig. 3. The configuration time is calculated as the time between the first OFPT_HELLO packet in the channel and the last OFPT_BARRIER_REPLY that shows the configuration of both nodes are completed. The topology for these experiments are similar to Fig. 2. To set precise RTT, we used ns-3 CSMA channel to replace the long-range low-data-rate UAN control plane channel. Both nodes and the controller are connected to the same CSMA bus to model a wireless channel with contention.

Fig. 4 shows the traffic of a 2-Node SDUWN simulation. The Y-axis denotes the size of data in Bytes. Solid line shows the OpenFlow packets, while the dot dash line denotes traffic for Node 1 and the deshed line denotes the traffic for Node 2. Since the flow table is very simple in this example, there will be only one OFPT_SET_CONFIG packet needed to set all configurations for one node. However, due to contention and congestion, the configuration process can only be finished after several rounds of retransmission. The flow tables have been configured around 380s. The controller will broadcast an OFPT_ECHO_REQUEST packet every 5s to check if every node is still online, and each node will reply an OFPT_ECHO_REPLY to acknowledge the controller, which is shown in Fig. 4 as the periodic small peaks. We started to transmit a file from Node

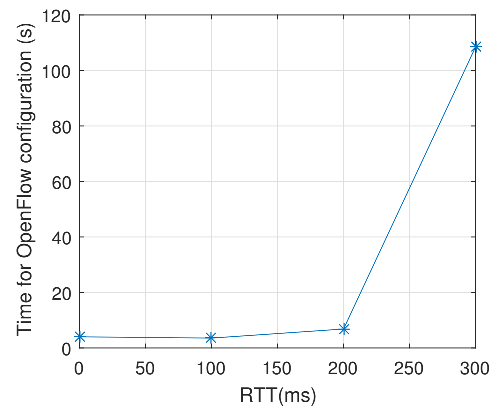

Figure 3: Configuration time for different control plane channel RTT.

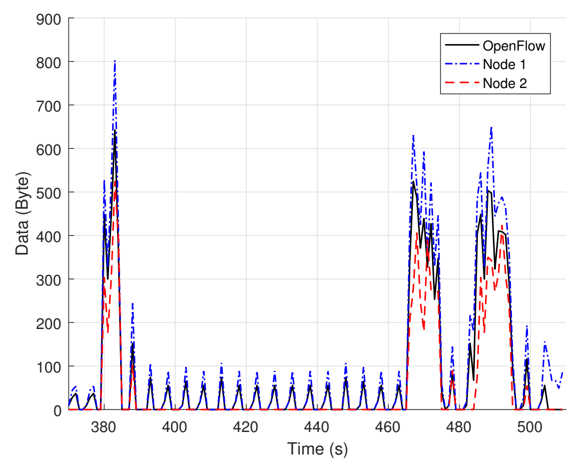

Figure 4: 2-Node SDUWN example with ns-3 CSMA control plane channle. The delay for CSMA channel is 100ms.

1 to Node 2 around 470s to validate that the network configuration is working.

Different from most existing wireless SDN scenarios, the control plane network of SDUWN is a wireless network with congestions and contentions. The original OpenFlow control plane communication is designed for secured wired connection, thus there is no mechanism to handle control plane contentions. We set the bandwidth of a CSMA control plane network to 10Mbps and simulated the handshaking process between the controller and multiple nodes. The simulation results show that with original TCP and OpenFlow design, the handshaking process cannot be finished if there are more than 4 nodes within the network.

4.3 Example with UAN Control Plane Channel

To illustrate that our simulation system can simulate the underwater environment, we configured a long-range low-data-rate channel as the control plane network with the parameters equivalent to

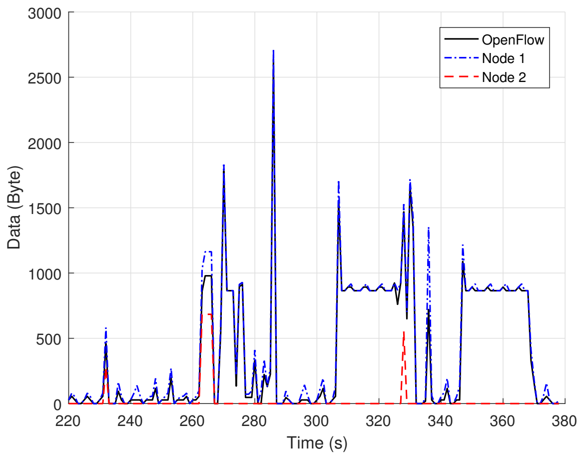

Figure 5: 2-Node SDUWN example with UAN control plane channle.

[15]. There are 2 SDUWN nodes and 1 controller in this example. Since the UAN net device module of ns-3 introduces extra delays, the distance between two nodes and the controller is set as close as 10m to lower the RTT. However, the RTT is still too long for the controller to finish the configuration.

Fig. 5 shows the OpenFlow packets (solid line) and data packets for Node 1 (dot dash line) and Node 2 (dashed line). The Y-axis is the size of packet in Bytes. The previous flow tables on Node 1 and 2 expired around 260s and the controller tries to update the flow tables during 260s to 290s. Then, we try to use Node 1 to transmit a file to Node 2 with the UDP protocol at 305s, since there were no valid flow tables for this service, the Open vSwith running on Node 1 sent these files to the controller as PACKET_IN packets. The controller tried to use PACKET_OUT to relay the packet to Node 2, but due to limited bandwidth and the lack of contention handling mechanism, it did not succeed.

5 CONCLUSIONS AND FUTURE WORK

This paper presented a SDUWN simulation system based on OpenNet and ns-3. The case study shows that the propagation delay, the limited bandwidth and the intermittent connectivity of underwater channel are major challenges for SDUWN protocol design, especially for existing OpenFlow protocol who employs TCP to establish the secured control plane connection. The wireless control plane network also requires radical modifications on SDN protocols, i.e. a contention and congestion handling mechanism for control plane is needed. Moreover, a hybrid in-band and out-of-band control plane configuration needs to be developed for existing SDN controller softwares. A SDN protocol for SDUWN will need to be designed since OpenFlow cannot be directly used.

ACKNOWLEDGMENTS

The work is supported by the National Science Foundation under Grant No.: ECCS-1651135, CNS-1551067 and CNS-1205757

REFERENCES

[1] Segar D. A. 1998. *Introduction to Ocean Sciences*. USA: Wadsworth Publishing Company. http://113.160.249.209:8080/xmlui/handle/123456789/8025

[2] Ian F. Akyildiz, Pu Wang, and Shih-Chun Lin. 2016. SoftWater: Software-defined networking for next-generation underwater communication systems. *Ad Hoc Networks* 46 (Aug. 2016), 1–11. https://doi.org/10.1016/j.adhoc.2016.02.016

[3] P. Casari, C. Tapparello, F. Guerra, F. Favaro, I. Calabrese, G. Toso, S. Azad, R. Masiero, and M. Zorzi. 2014. Open source suites for underwater networking: WOSS and DESERT underwater. *IEEE Network* 28, 5 (Sept. 2014), 38–46. https://doi.org/10.1109/MNET.2014.6915438

[4] M. C. Chan, C. Chen, J. X. Huang, T. Kuo, L. H. Yen, and C. C. Tseng. 2014. OpenNet: A simulator for software-defined wireless local area network. In *2014 IEEE Wireless Communications and Networking Conference (WCNC)*. 3332–3336. https://doi.org/10.1109/WCNC.2014.6953088

[5] Emrecan Demirors, Jiacheng Shi, Raffaele Guida, and Tommaso Melodia. 2016. SEANet G2: Toward a High-data-rate Software-defined Underwater Acoustic Networking Platform. In *Proceedings of the 11th ACM International Conference on Underwater Networks & Systems (WUWNet '16)*. ACM, New York, NY, USA, 12:1–12:8. https://doi.org/10.1145/2999504.3001112

[6] R. Fan, L. Wei, Pengyuan Du, C. M. Goldrick, and M. Gerla. 2016. A SDN-controlled underwater MAC and routing testbed. In *MILCOM 2016 - 2016 IEEE Military Communications Conference*. 1071–1076. https://doi.org/10.1109/MILCOM.2016.7795472

[7] Albert F. Harris, III and Michele Zorzi. 2007. Modeling the Underwater Acoustic Channel in Ns2. In *Proceedings of the 2Nd International Conference on Performance Evaluation Methodologies and Tools (ValueTools '07)*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), ICST, Brussels, Belgium, Belgium, 18:1–18:8. http://dl.acm.org/citation.cfm?id=1345263.1345286

[8] Jared Ivey, Hemin Yang, Chuanji Zhang, and George Riley. 2016. Comparing a Scalable SDN Simulation Framework Built on Ns-3 and DCE with Existing SDN Simulators and Emulators. In *Proceedings of the 2016 Annual ACM Conference on SIGSIM Principles of Advanced Discrete Simulation (SIGSIM-PADS '16)*. ACM, New York, NY, USA, 153–164. https://doi.org/10.1145/2901378.2901391

[9] Bob Lantz, Brandon Heller, and Nick McKeown. 2010. A Network in a Laptop: Rapid Prototyping for Software-defined Networks. In *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks (Hotnets-IX)*. ACM, New York, NY, USA, 19:1–19:6. https://doi.org/10.1145/1868447.1868466

[10] Son N. Le, Zheng Peng, Jun-Hong Cui, Hao Zhou, and Janny Liao. 2013. SeaLinx: A Multi-instance Protocol Stack Architecture for Underwater Networking. In *Proceedings of the Eighth ACM International Conference on Underwater Networks and Systems (WUWNet '13)*. ACM, New York, NY, USA, 46:1–46:5. https://doi.org/10.1145/2532378.2533868

[11] Chiara Petrioli, Roberto Petroccia, John R. Potter, and Daniele Spaccini. 2015. The SUNSET framework for simulation, emulation and at-sea testing of underwater wireless sensor networks. *Ad Hoc Networks* 34 (Nov. 2015), 224–238. https://doi.org/10.1016/j.adhoc.2014.08.012

[12] T. Riedl and A. Singer. 2014. Towards a video-capable wireless underwater modem: Doppler tolerant broadband acoustic communication. In *2014 Underwater Communications and Networking (UComms)*. 1–5. https://doi.org/10.1109/UComms.2014.7017122

[13] Dustin Torres, Jonathan Friedman, Thomas Schmid, Mani B. Srivastava, Youngtae Noh, and Mario Gerla. 2015. Software-defined underwater acoustic networking platform and its applications. *Ad Hoc Networks* 34 (Nov. 2015), 252–264. https://doi.org/10.1016/j.adhoc.2015.01.010

[14] Junfeng Wang, Yiming Miao, Ping Zhou, M. Shamim Hossain, and Sk Md Mizanur Rahman. 2017. A software defined network routing in wireless multihop network. *Journal of Network and Computer Applications* 85 (May 2017), 76–83. https://doi.org/10.1016/j.jnca.2016.12.007

[15] Li Wei, Zheng Peng, Hao Zhou, Jun-Hong Cui, Shengli Zhou, Zhijie Shi, and J. O'Donnell. 2013. Long Island Sound testbed and experiments. In *2013 OCEANS - San Diego*. 1–6. https://doi.org/10.23919/OCEANS.2013.6741276

[16] L. Wei, Z. Wang, J. Liu, Z. Peng, and J. H. Cui. 2016. Power Efficient Deployment Planning for Wireless Oceanographic Systems. *IEEE Systems Journal* PP, 99 (2016), 1–11. https://doi.org/10.1109/JSYST.2016.2533396

[17] J. Younce, A. Singer, T. Riedl, B. Landry, A. Bean, and T. Arikan. 2015. Experimental results with HF underwater acoustic modem for high bandwidth applications. In *2015 49th Asilomar Conference on Signals, Systems and Computers*. 248–252. https://doi.org/10.1109/ACSSC.2015.7421124