The SUNSET Framework for Simulation, Emulation and at-sea Testing of Underwater Wireless Sensor Networks

Chiara Petrioli[a,b], Roberto Petroccia[a,b], John Potter[c], Daniele Spaccini[a]

[a]*Dipartimento di Informatica, Università di Roma "La Sapienza," Via Salaria, 113, 00198, Roma, Italy. E-mail: {petrioli,petroccia,spaccini}@di.uniroma1.it.*
[b]*WSENSE s.r.l., Rome, Italy.*
[c]*NATO STO Centre for Maritime Research and Experimentation, Viale San Bartolomeo 400, 19126 La Spezia, Italy. E-mail: Potter@cmre.nato.int*

Abstract

The Sapienza University Networking framework for underwater Simulation Emulation and real-life Testing (SUNSET) is a toolkit for the implementation and testing of protocols for underwater sensor networks. SUNSET enables a radical new way of performing experimental research on underwater communications. It allows protocol designers and implementors to easily realize their solutions and to evaluate their performance through simulation, in-lab emulation and trials at sea in a direct and transparent way, and independently of specific underwater hardware platforms. SUNSET provides a complete toolchain of pre-deployment and deployment time tools able to identify risks, malfunctioning and under-performing solutions before incurring the expense of going to sea. Novel underwater systems can therefore be rapidly and easily investigated. Heterogeneous underwater communication technologies from different vendors can be used, allowing the evaluation of the impact of different combinations of hardware and software on the overall system performance. Using SUNSET, underwater devices can be reconfigured and controlled remotely in real time, using acoustic links. This allows the performance investigation of underwater systems under different settings and configurations and significantly reduces the cost and complexity of at-sea trials. This paper describes the architectural concept of SUNSET and presents some exemplary results of its use in the field. The SUNSET framework has been extensively validated during more than fifteen at-sea experimental campaigns in the past four years. Several of these have been conducted jointly with the NATO STO Centre for Maritime Research and Experimentation (CMRE) under a collaboration between the University of Rome and CMRE.

Keywords: Underwater, Acoustic network, Simulation, Emulation, SUNSET, Wireless sensor, Internet of Things, Internet of Underwater Things.

1. Introduction

Underwater Wireless Sensor Networks (UWSNs) have become an important area of research with many potential applications, including the monitoring and discovery of the marine environment, offshore hydrocarbon surveying and extraction, underwater CO_2 storage, coastal protection, and the prediction of underwater seismic and volcanic events [1, 2]. A common feature in all these applications is the increasing need of underwater communications technologies connecting existing static and mobile platforms into cooperative underwater monitoring and control systems. The protocols that will support this underwater communications networking do not yet exist and so must be designed and tested. Several solutions have been proposed to implement new networking protocols for UWSNs [3, 4, 5, 6]. New protocols are usually evaluated by means of simulations with only a few being tested at sea [7, 8, 9, 10, 11]. Comparisons between simulation and experimental results have clearly shown that existing simulation tools often do not capture important features of the environment or hardware and this reduces confidence in their usefulness for solution validation, evaluation and benchmarking. The shortfalls of simulations arise from two major areas. Firstly, simulations can only capture a subset of the environmental variability, resulting in an approximate and generally simplified model of the acoustic channel and its dynamics [12]. Secondly, simulators also do not generally account for hardware features which can significantly impact system performance [13].

Even if a simulator captures the environment and hardware with sufficient fidelity, moving from simulation to real-life testing usually requires significant changes in the code needed to run on representative hardware embedded platforms. Significant effort is therefore spent in preparing expensive at-sea experiments, much of which is consumed solving problems identified only at the time of test. This results in significant inefficiencies in exploiting the limited at-sea testing opportunities. Research and development of innovative solutions for underwater monitoring systems would therefore greatly benefit from a tool that allows a seamless transition from simulation, where malfunctioning and incorrect code can be effectively detected and corrected, to an at-sea testing environment without code rewriting, avoiding delays and errors in code re-implementation. Such a tool should also enable tuning of the protocol parameters on the fly to efficiently explore the dependence of performance on parameter settings. This would be a key asset to speed up innovation in the field, allowing researchers to easily test, evaluate, and compare the performance of different protocols for UWSNs. To create this tool, we have developed and extensively validated a complete framework for Underwater Acoustic Sensor Networks (UASNs) simulation, emulation and real-life testing that we call SUNSET.

The first framework was originally presented and tested in [14] and it has since been extensively extended and improved [15] leading to the creation of SUNSET, the *Sapienza University Networking framework for underwater Simulation Emulation and real-life Testing*, recently released open source [16]. SUNSET version 1.0 was freely released in May 2012, following which it has been significantly improved and enhanced, adapting to the lessons learned during extensive at-sea experimentation, resulting in SUNSET version 2.0 [17], released in October 2013[1]. SUNSET is based on the well known network simulator ns-2 [18] (and its extension ns2-Miracle [19]). Both these simulation tools have been significantly extended to reduce the gap between simulation and at-sea results. More accurate channel and interference models have been implemented, with the possibility to replay the channel conditions measured at sea and to use real hardware models. Using SUNSET, researchers and developers can first implement, evaluate and improve their solutions in simulation, using ns-2. Developers can then seamlessly use the same code for emulation and at-sea testing, where real modems are used for data transmission and additional external devices can be integrated to the nodes for sensing and navigation. Five different acoustic modems, various sensing platforms, and several surface and underwater robots have been already interfaced and tested at sea.

The framework has been extensively validated and evaluated through several trials at different locations (at sea, in rivers, lakes, fjords) and considering different network configurations (mobile and static nodes placed according to different topologies). In particular we have tested MAC, routing, localization and synchronization solutions in network configurations ranging from small scale single-hop networks (4-5 nodes) to large scale multi-hop scenarios (up to 12 nodes). The collected results have shown that SUNSET is a powerful, reliable and flexible solution for both simulations and at-sea trials, leading to no impairment in terms of protocol performance. Moreover, SUNSET supports fast reconfiguration and fast porting of all the implemented solutions to different hardware assets. One of the main functionalities of the latest SUNSET version is a novel mechanism, called *back-seat driver*, to remotely control and operate the entire underwater network via acoustic links [20]. This module allows researchers to remotely operate the network in real time, reconfiguring the network (activating only the nodes needed for the specific tests), changing protocol and node parameters and test settings. Everything is done ’on the fly’, without the need to retrieve the underwater devices. This allows researchers to easily run different experiments without interruptions, thus saving time during at-sea campaigns.

[1]The main SUNSET components for simulation, emulation and at-sea testing are released as open source software. Additional modules and solutions, developed in collaboration with industrial partners or covered by a non disclosure agreement, are released in a binary format or are not released.

The rest of this paper is organized as follows; Section 2 outlines the state of the art of testbeds and emulation frameworks for underwater networks. A detailed description of the latest version of the proposed SUNSET framework is presented in Section 3. In Section 5 we report the results of tests we have conducted in the past years to validate and evaluate SUNSET during at-sea testing activity. Finally, concluding remarks are given in Section 6.

2. Related Works

Several underwater acoustic sensor network frameworks have been proposed to evaluate protocols for UWSNs [21] [22] [23] [24], [25].

Aqua-Sim [21] is a simulation tool developed on the ns-2 simulator [18], which however does not support the use of any ray tracing software to more accurately model the underwater acoustic channel. Additionally, it can be used only to run simulations without the possibility to test the implemented solutions on real platforms in at-sea experiments.

The Aqua-Lab testbed [22] consists of a water tank, a set of WHOI Micro-Modems and Application Programming Interfaces (APIs). Using the APIs, users can develop their own applications without knowing the exact mechanisms of the underlying acoustic physical layer. Using Aqua-Lab the user can also emulate different network topologies, propagation delays, and attenuation. Aqua-Lab with Micro-Modems has been used to conduct a set of experiments in both field and lab environments.

Aqua-Net [23] consists of a layered structure supporting cross-layer optimization. Different physical devices (e.g. Micro-Modems and Teledyne Benthos modems) can be supported at the physical layer. Aqua-Net has been implemented in a real system that consists of both hardware and software running on Gumstix computers [26].

SeaLinx [25] extends the Aqua-Net framework, enabling the user to run an entire protocol stack directly on the modem with the support of a more enhanced cross-layer communication. As for Aqua-Net, however, SeaLinx is designed to run only in emulation and real-life test modes with no support for simulations.

In [24] the authors propose a unified simulation and implementation software framework to support underwater MAC protocol development. The same C code is used in both the simulator and the modem to implement the logic of the selected MAC protocol. The simulator captures the essential behavior of the ARL OFDM modem selected for the framework and uses the same software interfaces as the modem. Since the same code is used for simulations and real-life experiments, performance comparisons between simulations and sea trials are possible.

An extension of [24] has been proposed in [27] where the UNET-2 framework is presented. The paper describes a software defined modem and a communication protocol stack based on the ARL modem, providing the possibility to investigate new coding schemes and to evaluate complete networking solutions by means of both simulations and at-sea trials. UNET-2 makes significant progresses in the direction of software defined acoustic modems, but it provides little support for the implementation and evaluation of adaptive, cross-layer optimized protocol stacks.

A limitation of all these solutions is that developers have to study new proprietary architectures and software to test new protocols. This limits accessibility to testing facilities and it is usually not trivial as it means that protocol code generally needs to be rewritten to test it in different scenarios.

In [28] the authors approach the task in a similar way to SUNSET, investigating the use of an open source platform to create a simulation/emulation tool for UWSNs. A new platform, named UANT, is proposed which is based on GNU radio, a software defined radio framework, interfaced with a PC running TinyOS (or TOSSIM in case of simulation/emulation). The use of the well known GNU radio concept allows researchers to test the impact on performance of a large library of supported modulation schemes including GMSK, PSK, QAM, OFDM etc. UANT is an interesting concept that adds flexibility to what can be tested at the lower layers of the stack while allowing researchers to interact with a well known simulation and deployment tool (TOSSIM/TinyOS). The main limitation is that UANT uses the computationally expensive GNU radio, so needs to run on a PC; an unlikely platform for real-life experiments due to housings and energy constraints. Moreover, running solutions on TinyOS means that protocol design has to follow rules that reflect features of IEEE 802.15.4 type devices and which do not always match features of underwater nodes and underwater communication devices. Additionally, TOSSIM is currently not able to support an underwater channel model. In [14] we presented and investigated, through lab and at-sea tests, the first simulation/emulation tool based on ns-2. We then improved this tool in [13], porting it to work on ns2-Miracle, thus making use of a more accurate model for the underwater acoustic channel. In [13] we have also performed a first investigation on the gap between actual at-sea and simulation networking performance, showing how to reduce this gap, improving simulation accuracy.

Recently, the DESERT framework has been proposed in [29]. DESERT is a simulation/emulation tool along the lines of that proposed in [14, 13]. DESERT implements a similar approach to SUNSET, with a set of libraries based on ns-2 and ns2-Miracle, focusing on the design of MAC and routing solutions. Since both SUNSET and DESERT are based on the same software, there is a high compatibility and interoperability between the protocol solutions and the modules developed for these two frameworks when running in simulation mode. However, as presented in [30], DESERT currently lacks several important features to efficiently move from simulations to real-life tests which can result in lower at-sea performance.

3. SUNSET

The first version of SUNSET was presented in 2011 [14], with enhancements released in 2012 [15]. It provided a complete toolkit to seamlessly simulate, emulate and test at sea novel communication protocols for Underwater Acoustic Sensor Networks (UASNs). SUNSET is based on the open source and well known network simulator ns-2 [18] (and its extension ns2-Miracle [19]), largely used by the research community. The reason for extending an open source architecture, rather than designing a new one from scratch, is to limit the effort needed by the networking research community that is already familiar with ns-2. This allows easy implementation and testing, without the need to learn a totally new software framework. The SUNSET framework has also been designed to separate the protocol stack implementation from the external modules controlling the use of real hardware when running in emulation and test mode. Developers and researchers can therefore easily implement new protocol solutions and run tests changing either the selected protocols (MAC, Routing, etc.) or some of their protocol parameters without any changes to the external devices code. At the same time, they can change the configuration parameters for the selected communication hardware without any change to the actual protocol stack.

Using SUNSET, protocol implementation can be first evaluated and tested in a controlled simulation environment, where a large set of configurations and settings can be considered, usually much larger than what can be tested at sea. Researchers and developers can then use the same code in emulation and field test mode, where real acoustic modems and additional external devices for sensing and navigation operations can be used. Additionally, SUNSET provides the possibility to remotely and wirelessly control in real time the network components and the tests to run using acoustic links. Given the costs of node deployment and at-sea operations, this is of paramount importance. Statistics and measurements regarding on-going experiments, and information on the status of the nodes, can be quickly collected to modify the protocol stack and its parameters, node settings, and network configuration. In particular, SUNSET allows one to change the network topology (remotely activating or deactivating a subset of the underwater nodes), tune the settings of the selected protocol and change the experiment parameters (e.g., policies to be followed by each device; which node has to act as collection point, if any; which nodes have to generate data; what traffic load has to be used, etc.), change the selected protocol stack, etc.

The architecture is flexible and open enough to allow the integration of any external device, once APIs are provided to control its operations. Using APIs, a driver to properly handle the device functionalities, data exchanges and interaction with the specific device can be easily implemented.

To the best of the authors' knowledge SUNSET is the first and only available tool to provide all the following functionalities and features in a single package:

- Simulations using different underwater acoustic channel models, such as empirical formulas (e.g., Urick's models [31]), the Bellhop [32] propagation simulator via the WOSS [33] interface (which includes real environmental parameters and provide more accurate results), and traces of the channel used for channel replay purposes [34].

- Reuse of the same code for simulation, emulation and at-sea test mode, introducing little additional delays (in the order of milliseconds on very low-power embedded devices) with limited additional overhead per packet (few bits).

- Support of five different commercial off-the-shelf acoustic modems for acoustic communication, namely, WHOI FSK and PSK Micro-modems, Evologics modems, Kongsberg modems, and Teledyne Benthos modems.

- Support of different sensing platforms and vehicles, including environmental underwater sensors for temperature, CO_2 and methane concentrations [35], pH, redox, pressure, salinity, conductivity, O_2 concentration, MARES Autonomous Underwater Vehicles (AUV) and Autonomous Surface Vehicles (ASV) developed by the INESC TEC/University of Porto, Folaga AUV developed by Graaltech, and Seacon AUV, developed by the LSTS group at University of Porto.

- Runs efficiently on small portable devices (e.g., Gumstix, PC104, IGEPv2 or other ARM-based systems), allowing the user to embed it inside modem or AUV housing.

- Combines acoustic communications, sensing and networking capabilities in a single unit, supporting simple interaction with static and mobile nodes in a remote and on-line way, through acoustic links.

SUNSET was selected as the standard platform for the FP7 STREP project CLAM [36], with the objective of developing innovative solutions for pipeline monitoring. During CLAM, SUNSET was further extended and used for both simulations and at-sea operations. SUNSET is also the standard platform for the new FP7 ICT project SUNRISE [37], to create a federation of internet-connected underwater testbeds in Europe and North America to test novel technologies for the Internet of Underwater Things.

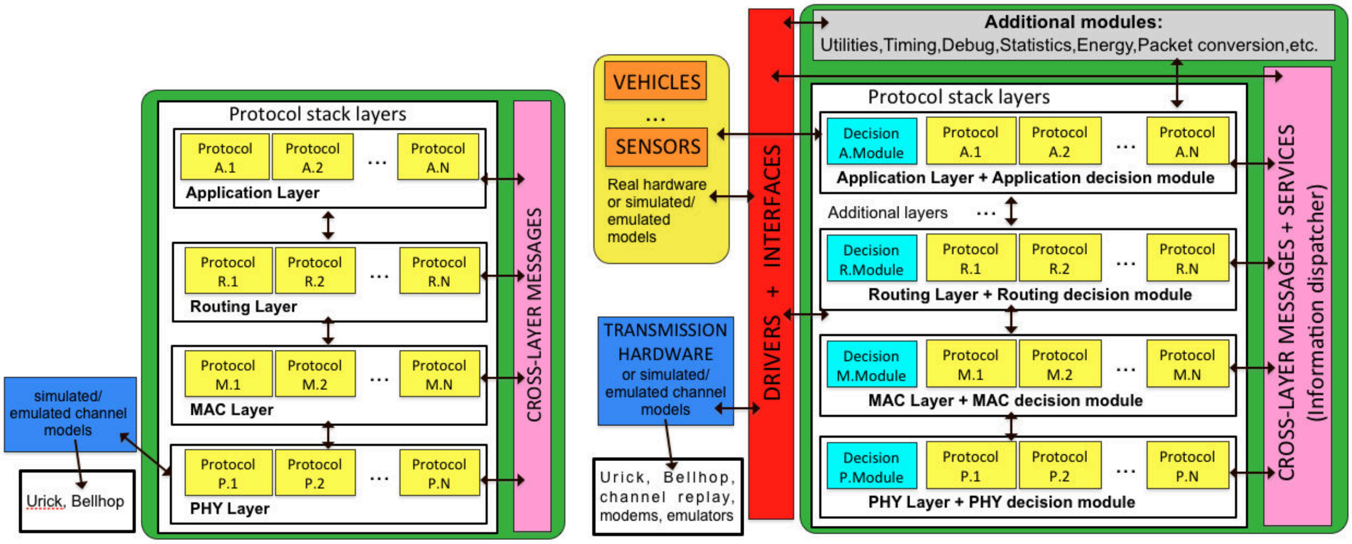
In the remainder of this section we describe the SUNSET functionalities in detail. We introduce the extensions to the ns-2 and ns2-Miracle simulation frameworks in Section 3.1. We present the simulation modules of SUNSET in Section 3.2. Modules for emulation and trials at sea are presented in Section 3.3. Finally, in Section 3.4 and Section 3.5 we describe the functionalities for the use of SUNSET over the Internet and for remote control of underwater acoustic devices, respectively.

3.1. Extending Ns-2 and Ns2-Miracle

The Network Simulator version 2 (ns-2 [18]) is a discrete event simulator for networking research. It is widely considered a reliable simulation tool for computer networks and it is used in over 600 institutions (academia and industry) scattered in 50 countries. The ns-2 simulator provides a very large number of applications, of protocols, of network types, of network elements and of traffic and mobility models. Ns-2 is based on C++ and uses *Tcl* as its scripting language. *OTcl*, an object-oriented dialect of Tcl, is used to execute command scripts, while *Tclcl* is used to link C++ and OTcl classes. Using the Tcl script, the users can define a particular network topology, select specific protocols (for each layer of the protocol stack), indicate the applications and settings to be simulated without the need to recompile the code in the case of changes.

Ns2-Miracle is an extension of ns-2 [19]. It uses the same event driven scheduler and scripting languages, adding a set of libraries to enhance some ns-2 functionalities, supporting the exchange of cross-layer messages and enabling the coexistence of multiple protocols in each layer of the stack. Additionally—and quite importantly for underwater research—ns2-Miracle implements the use of the Bellhop ray tracing software [32] to model the underwater acoustic channel via the World Ocean Simulation System (WOSS) interface [33]. A schematic representation of the original ns2-Miracle architecture is shown in Figure 1a.

Both ns-2 and ns2-Miracle are designed as simulation tools. They do not provide efficient support to run at-sea tests or interact with external hardware and software. With SUNSET we radically extend ns2-Miracle with a set of new functionalities allowing seamless code reuse for simulation, emulation and actual at-set experimentation. Figure 1 shows the wide range of the SUNSET extensions to ns2-Miracle. SUNSET inherits the ns2-Miracle layered structure for network simulation (including all protocol stack layers from physical to applications), including multi-protocol co-existence and support for cross-layer functionalities.



(a) The ns2-Miracle framework. (b) SUNSET.

Figure 1: SUNSET vs. ns2-Miracle: Beyond simulations to support for emulation and testing at sea.

However, to make it possible to experiment beyond simulation and provide support for emulation and actual testing at sea, SUNSET introduces a host of new modules, depicted in Figure 1b.

The new SUNSET core modules are described next.

Utilities module. The utilities module makes the use of the framework in simulation and emulation mode transparent to the user, providing functionalities to check if the framework is running in simulation or emulation mode and acting accordingly. For example, when scheduling an event, the utilities module is responsible for calling the specific scheduler, standard or real time (Section 3.3), which has to take care of event scheduling. It is also in charge of properly deallocating the memory allocated for a ns-2 packet. In particular, when running in simulation mode no actual memory is allocated for the packet payload. Instead, when running in emulation mode, additional memory is typically allocated for the packet that has to be erased when the packet is no longer needed.

Timing module. Simulators typically ignore delays in real devices such as: 1) Computational delay; 2) delay of internal operations of the devices (acoustic modems, embedded devices, sensors, etc.); 3) delays of the communication between hardware components on the same node; 4) delays of the actual message transmission, due to extra bytes added by the device. The timing module models these delays and the overhead when both in simulation and emulation mode, increasing fidelity.

Debug module. The debug module provides a tool to log and process debug information according to a given priority. In particular, a debug level can be assigned to each information to be logged. That is, when debug level k is chosen, only the information with an assigned debug level lower or equal to k is logged. Moreover, the debug level can be changed on demand, increasing or reducing the amount of logged information.

Statistics module. The statistics module provides a tool to collect measurements to easily evaluate protocol performance without the need to have advanced programming skills. Metrics include packet delivery ratio, protocol overhead, network throughput, packet latency, energy consumption, etc. When running in emulation or at-sea mode, each node stores its own information as an individual unit. The data from the specific nodes can be requested while the test is running or can be extracted at the end.

Information dispatcher module. The information dispatcher module implements a publish-subscribe paradigm. It allows the efficient sharing of information among different layers and among modules in the same layer. This is particularly important for networks with limited resources and/or deployed in challenging environments, like UWSNs [38]. Each module needs to register with the information dispatcher in order to provide information to or request information from other modules. Once the dispatcher is informed about the data each module can provide or request, it starts collecting the provided information and sending notifications to the requesting modules. All modules can request information on demand or ask for a periodic update. A timestamp is added to each information. The configuration of the information dispatcher can be performed by the user via simple TCL scripts. New cross-layer information can be defined, added and removed without the need to recompile the code.

3.2. Simulations with SUNSET

In this section we describe the new SUNSET modules designed to improve the environmental fidelity of ns-2 and ns2-Miracle in simulation mode. Extensions include support for more realistic underwater acoustic channel models.

Packet error model. The packet error model has been implemented to allow the user to customize specific network scenarios. Desired Packet Error Rates (PERs) can be configured at the network or link level, thus creating network topologies and link quality as needed for a given test. Three different error models are currently provided: BPSK-PER, FSK-PER and Static-PER. These packet error models can be used by each of the physical modules implemented in SUNSET. When the BPSK-PER or FSK-PER are used, the Bit Error Rate (BER) and the corresponding PER (depending on packet size and on the modulation scheme) are computed according to the signal-to-noise-plus-interference ratio (SNIR) value. When the Static-PER model is selected, the desired average packet error rate can be selected by the user, either network-wide or per link. Additionally, users can remove links from the simulated network and blacklist nodes. This feature can be useful not only when running simulations but also during lab and at-sea experiments. In particular, it can happen that when the network is deployed at-sea, all nodes are able to communicate with each other. In such a case no multi-hop routing test would be possible. Using the SUNSET packet error model, multi-hop communications can be forced via software (simply disabling reception over some links), in a way that is transparent to the protocol stack.

Channel replay. Two SUNSET modules have been designed to allow the use of channel replay techniques [34, 39]. The first module provides functionalities for collecting channel quality information by triggering periodic packet transmissions by each node. The second module is used to load the collected measurements and replays the underwater acoustic channel conditions collected at sea during simulations. For all the nodes in the network, correct message receptions and errors are used to create a channel matrix, which varies over time.[2] A propagation matrix is also created to capture the propagation delay on each link. This module is particularly useful to compare the performance of multiple protocols under the same environmental conditions, which is not possible when actually at sea. In fact, in case of a highly dynamic underwater acoustic channel, even if the tests of two different protocols occur close in time (e.g., one immediately after the other), the quality of the links in the network can vary quite quickly, making it problematic to distinguish between channel variability and inherent protocol performance.

Energy consumption model. The energy consumption model provides a way to compute the energy consumed by each node while transmitting, receiving or being idle. Differently from what is possible by using ns2-Miracle, real figures of specific devices, such as the power consumption of a real acoustic modem, can be easily loaded and used to compute energy consumption values. The model also supports the use of different transmission power levels when transmitting acoustic packets in the underwater channel.

3.3. Using SUNSET for in-lab emulation and at-sea testing

Emulation is a useful next step, after simulations, for the evaluation of the performance of underwater protocols. It allows accurate parameter checking and provides an opportunity to tune protocols by means of in-lab tests based on actual hardware before experiments at sea. When running in simulation mode an entire network with multiple nodes is simulated by a single SUNSET installation on the same PC. When running in emulation and at-sea test mode, each node of the network is implemented by a different SUNSET process,

[2] If the acoustic modem allows it, additional information on the link quality, e.g., SNIR, are also stored in the matrix.

each running on different hardware (including PCs and embedded devices). Significant extensions to ns2-Miracle are needed to provide support for emulation and actual testing at sea. In particular, new modules have been introduced for greater system efficiency, reliability and accuracy for use on actual hardware during in-lab and at-sea tests.

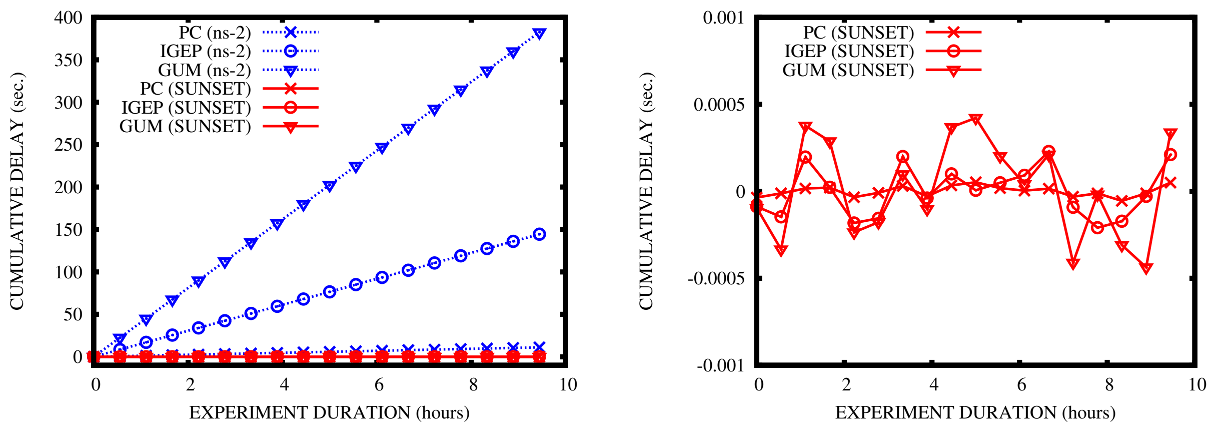
All these modules have been carefully designed and implemented to make the overall framework lightweight, thus enabling it to run on small, energy efficient, compact and inexpensive devices. The use of these devices is really important when planning at-sea tests, since they can be mounted in small housings (e.g., modem casings and on AUV and ASV), thus reducing costs and easing test preparation. SUNSET has been successfully ported to work on several different embedded and portable platforms, including Gumstix computers[3] [26], PC104, IGEPv2 and other ARM-based systems as well as MANTA portable systems [40]. Moreover, for devices supporting external memory cards (SD, MMC, etc.) SUNSET can be installed on the secondary memory. The user can just plug in the external memory and reboot to get SUNSET ready for use.

Real-Time scheduler. We have implemented a new real time scheduler to measure elapsed time that is more accurate than the one provided by ns-2. Our scheduler includes support for multi-threading, allowing for efficient interaction with any external device. Furthermore, the ns-2 real time scheduler does not take into account the computational delay of the specific platform used to run the software. Different platforms introduce different delays. For instance, we have estimated that the delay added to create an event, to add it to the scheduler, and to remove it from the scheduler at the proper time, is of 0.065 milliseconds on a desktop PC (PC), 0.85 milliseconds on the IGEPv2 board (IGEP), and 2.25 milliseconds on the Gumstix Verdex Pro (GUM).[4] Having a scheduler that is platform dependent means that if a heterogeneous set of devices is used in the network, even if these devices are all strictly synchronized, no synchronization in the operations performed by the different devices can be guaranteed, unless the user explicitly takes into account the introduced delays in its solutions. This, however, makes the code implementation more complex. To solve this problem, the SUNSET real time scheduler takes care of any delay added by the running platform making it transparent to users and developers. More specifically, it keeps track of the delays introduced to create a new event, to add it to the scheduler and to remove it from the scheduler at the proper time and adjusts the scheduling activities accordingly.

[3] We have run experiments using Basix, Connex, Verdex and the new Overo Gumstix boards.

[4] Desktop PC (CPU:Core(TM)2 Quad CPU Q6600 @2.40GHz, RAM: 4 GB); IGEPv2 board (CPU:ARM Cortex-A8 DM3730 @1Ghz, RAM: 512 MB); Gumstix Verdex Pro (CPU:Marvell PXA270 @600MHz, RAM:128MB low-power DDR).

Figure 2a shows the time difference between when the event is supposed to occur and when it actually occurs using the ns-2 and the SUNSET real time schedulers. We consider periodic events, with a period of 0.2 seconds, such as those generated by sensory data or from data from the navigation/telemetry system of a vehicle. We observe that when using the ns-2 real time scheduler the time difference increases over time (up to several tens of seconds) depending on the platform (PC vs. IGEP vs. Gumstix). This behavior is automatically accounted by SUNSET. Zooming into the SUNSET real time scheduler performance (Figure 2b), we observe a time difference that is always lower than 0.5 milliseconds per event, which does not increase over time, and that is stable irrespective of the platform. The SUNSET real time scheduler is clearly platform independent.



(a) Scheduling delays for periodic events.

(b) SUNSET scheduling delays (zoom)

Figure 2: Event scheduling delay-induced drifting due to computational delays.

Packet converter module. When running in emulation mode or when testing protocols at sea, it is important to reduce the packet header as much as possible for optimizing in-water transmissions. We have therefore designed a module for stripping ns-2 packets of information that is not necessary for transmission by actual devices. The module converts ns-2 packets into streams of bytes and vice versa, compressing the information in the header to the minimum possible size. If a new packet header is defined, the developer has to define the methods for packet conversion and reconversion. In this way the designer of the packet header is also responsible for its conversion. Our approach is flexible enough to allow the user to decide through the Tcl script what information from the packet header has to be considered or discarded in the conversion process and what is the size of the resulting information to be transmitted. In this way, the implemented code does not need to be recompiled and the user can make any changes easily and efficiently. This module can only be used in emulation and at-sea mode and it does not limit the use of regular ns-2 packets in simulation mode.

Application drivers. SUNSET has been designed to allow easy integration with any external device. Drivers are used to properly handle the device functionalities, for data exchange and interaction with the devices (Figure 1). Drivers for several devices are already provided with the current version of SUNSET. These devices are connected to the application layer of the protocol stack and can therefore make use of the networking features provided by the lower layers to exchange data and commands in the network over single-hop and multi-hop routes. Once a driver for a specific device is available, the new hardware can be simply plugged in the SUNSET framework. These drivers can be used with real hardware as well as on top of software emulating device operations. SUNSET currently supports different sensing platforms and vehicles (both surface and underwater robots) and for each of them a specific driver has been implemented and tested at sea.

Acoustic modem drivers. Specific drivers for commercial modems have been implemented for SUNSET to handle modem functionalities. Drivers extensively tested at sea include those for FSK and PSK Micro-Modems, Evologics modems (firmware versions from 1.4 to 1.7), Kongsberg modems, and Teledyne Benthos modems. Given the high flexibility of the framework, any modem with APIs to transmit and receive packets can be easily interfaced. These drivers can be used with actual hardware or when a modem emulator is used during the testing and preparation phases, e.g., the S2C Modem emulator [41].

Channel emulator. The SUNSET channel emulator is a powerful tool that enables the creation of an in-lab testbed for emulating an entire underwater network without the need of real acoustic modems. The network can be made up of several devices, running on the same PC and on different platforms (such as embedded devices), connected via Ethernet. The propagation delay for each link is computed by the channel emulator according to the position of the different devices assigned in 3D space. The position can be updated over time or when an event occurs, transparently emulating node mobility. Users can combine the SUNSET packet error model and blacklist features with the channel emulator to create scenarios and network topologies suitable to given applications. The use of the channel emulator allows researchers to test, correct and tune their protocols and protocol parameters before actual at-sea tests.

3.4. SUNSET over Internet

SUNSET can be also used to interconnect devices via Internet connections. This feature has been proven to be particularly useful when testing and preparing at-sea experiments remotely with collaborators. A first example of this distributed testing infrastructure was exploited in December 2012, where multiple components, deployed in three different countries, were interconnected [42]. A software emulating the

FEUP/INESC TEC vehicle dynamics was running in Porto, Portugal; SUNSET was running in Rome, Italy, providing the networking support for communications and making use of the packet error model to introduce acoustic channel dynamics, and, finally, a software emulating the Evologics acoustic modem [41] was running in Berlin, Germany. This distributed testbed has been used to test, in a controlled environment, the software controlling the vehicle and the modem, the protocol stack, and the driver implementations to be used for at-sea experiments. Several simulations have been performed with the twofold objective to a) validate the communication solutions and the performance of the coordination algorithm, and b) investigate the performance of the system under a wide set of possible scenarios, beyond that of the specific at-sea deployment. The experimental results have confirmed the validity and reliability of the proposed emulation approach, showing agreement between the results from emulation and at-sea trials. The delay of the Internet communication was in the order of few milliseconds, shortly delaying the exchange of data between SUNSET and both modem and vehicle emulators. When interacting with the modem emulator, each millisecond of delay was interpreted as if the link was 1.5m longer. Since the distance of the devices in the network was set to several tens of meters, this extra delay/length can be considered negligible. Similarly, when interacting with the vehicle emulator, the rate at which SUNSET collects data from and sends commands to the vehicle is in the order of hundreds of milliseconds or higher, and therefore the few milliseconds of delay over the Internet do not detrimentally affect the outcome of the experiment.

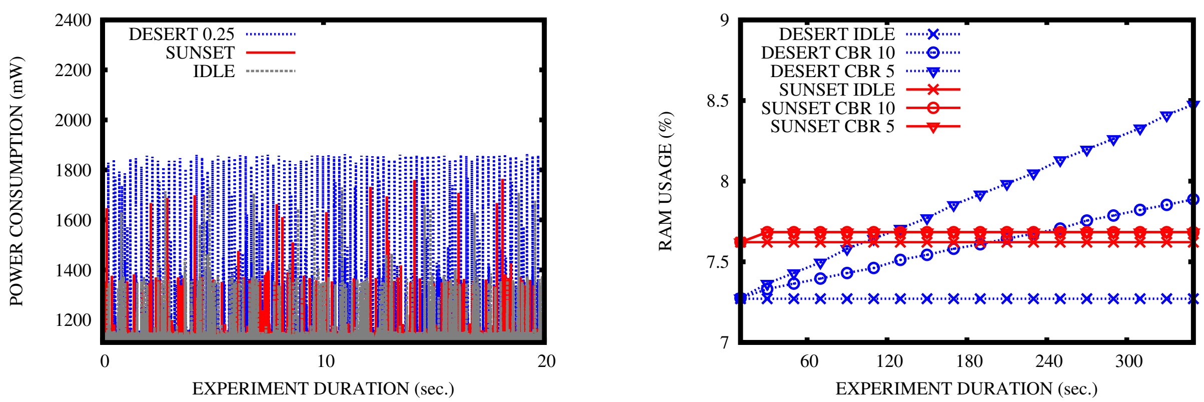
SUNSET can also be connected via Internet to existing testbed facilities, such as the Littoral Ocean Observatory Network (LOON) [43] hybrid communications testbed available at the Centre for Maritime Research and Experimentation (CMRE). The LOON consists of four seabed platforms equipped with modems and cabled to shore, where a PC acts as a server that is connected to the Internet. In this case, real devices deployed at sea and accessible over the Internet can be used for protocol testing instead of emulating the channel.

3.5. Back-seat driver

The SUNSET back-seat driver is a module specifically designed to remotely control and operate an underwater network via acoustic links [20]. Such a tool is of paramount importance for at-sea experiments. In particular, when an underwater network has to be reconfigured and there is no direct access to the nodes (for instance, via cable or via radio connection), the only option for reconfiguration is to physically recover the equipment, which is costly and may introduce large delays. Through the back-seat driver module, instead, SUNSET uses acoustic communications to start a new experiment with a selected protocol stack, to tune the parameters of a specific protocol, to stop a test and to control the status of an on-going experiment. Direct access to at least one node in the network is required through which all the the other nodes will be instructed. The kind of control enabled by the SUNSET back-seat driver protocol stack requires robust communications. Our implementation of the back-seat protocol stack achieves robustness, even if at the cost of a higher overhead.

4. SUNSET vs. DESERT: A comparison

In this section we compare the design and performance of SUNSET with that of a similar testing framework for UWSNs called DESERT [29]. DESERT follows the same approach of SUNSET in that it extends ns-2 and ns2-Miracle with functionalities for simulation, emulation and actual testing of solutions for UWSNs. A thorough performance comparison of SUNSET and DESERT version 1 is presented in [30]. The authors show how, being extensions of the same ns-2 and ns2-Miracle open-source software, both SUNSET and DESERT are highly compatible when running in simulation mode. The main differences between the two frameworks come in the way they support at-sea experiments. DESERT uses the real-time scheduler provided by ns-2, thus incurring in all the problems of scheduler accuracy listed in Section 3.3. Additionally, both SUNSET and DESERT use multiple threads when interacting with external devices. The ns-2 real-time scheduler, however, does not support interaction among multiple threads efficiently. More specifically, a main thread has to actively control the occurrence of any new event and data provided by secondary threads, thus incurring in a higher CPU, memory and energy consumption.



(a) Power consumption.

(b) RAM usage.

Figure 3: SUNSET and DESERT resources consumption on a Gumstix Verdex board.

Figure 3a shows a portion of 20 seconds from the power consumption traces collected using the Gumstix

Verdex board to run SUNSET and DESERT and when the board is in IDLE state (i.e., when no framework is running). The checking rate of the main thread for DESERT has been set to 0.25 seconds, which is a realistic value when performing periodic reading of data from a sensor or from the navigation system of a vehicle. We can clearly see that every 0.25 seconds there is a peak in the power consumption meaning that the main thread is checking for new events. DESERT power consumption is on average around 62% higher than when the system is IDLE. Since SUNSET does not introduce additional operations it shows always a power consumption close to that of IDLE. Figure 3b shows the percentage of RAM used by SUNSET and DESERT on the Gumstix Verdex board, when no data packets are generated (IDLE), and varying the packet generation rate. We observe that initially SUNSET requires more memory than DESERT (around 4.7% more) to load the additional supported libraries. When data packet generation starts, the RAM usage of both SUNSET and DESERT slightly increases, because the memory for those packets has to be actually allocated. The RAM usage of SUNSET is constant over time, which provides an indication that no memory leaks occur and that all data packets are correctly destroyed when they are not needed anymore. In the case of DESERT, instead, RAM usage increases over time and it increases faster for higher traffic. This suggests the presence of memory leaks related to packet memory allocation, as also noted by the DESERT authors [44]. Experiments on a computational board more powerful than the Gumstix, namely, the IGEPv2, show similar trends.

Another difference between SUNSET and DESERT concerns the mechanism for converting ns-2 packets into streams of bytes before transmissions in water. The SUNSET Packet Converter module allows to define at run time the packet header fields that the user wants to convert and the number of bits to use for each field. This approach allows the user to compress as much as possible the actual amount of information transmitted in water. DESERT version 1, instead, supports a simpler, but less flexible, approach. Fields and sizes of the packet header are hard coded and cannot be easily changed by the user. Any change requires to recompile the code and to upload the updated libraries to all the underwater devices. Most of these fields, however, depend on the actual network deployment, network topology and protocol configuration, which introduces a packet conversion overhead higher than actually needed.[5]

[5] We are aware of a recently released new version of DESERT, enhancing packet conversion and some of the simulation modules along the lines of what done in SUNSET. At this time we have not been able to compare the performance of the new DESERT conversion module against that of SUNSET.

5. In-field Testing with SUNSET

SUNSET has been extensively used in more than fifteen at-sea campaigns over the past four years, including several trials in collaboration with the NATO STO Centre for Maritime Research and Experimentation (CMRE). In this section we provide details on these testing activities. Experimental campaigns are reported chronologically, which is also functional to show the evolution of the SUNSET framework over time. Further details on the various tests can be found on the SUNSET webpage [16].

5.1. SUNSET for protocol testing

SUNSET has proven effective for testing at sea a number of protocols, whose performance was previously tested via simulations, by running the same code on real underwater hardware with minimal changes. We start by describing tests and lesson learned from MAC protocols, and then we show the use of SUNSET for testing multi-hop underwater routing.

MAC. A first set of experiments on MAC solutions were conducted in September 2010 during the CMRE ACommsNet10 campaign. At the time we used SUNSET version 1.0. We chose three solutions of increasing complexity, namely, CSMA [11], T-Lohi [45] and DACAP [46]. The network was deployed around Pianosa, an island of a protected marine park on the west coast of Italy. The test included a total of 6 nodes. In particular, 3 static nodes were deployed on the sea bottom as the vertices of a triangle with a side 1km long; 1 static node was floating at surface level, in the middle of the triangle, and finally 2 were mobile: One was attached to the Folaga AUV and the other to the CMRE CRV Leonardo. Each node was equipped with the WHOI FSK Micro-Modem. One major feature of SUNSET that turned extremely useful for at-sea testing is that it allowed us to pass from one test to another by just changing few lines of the configuration script of a protocol. For instance, evaluating DACAP with different packet sizes required just changing the value of the corresponding parameter of the configuration script for that protocol, without need to recompile any code.

Key measurements allowed by SUNSET include protocol packet delivery ratio and latency. As an example, in Table 1 we show the protocol performance for the Pianosa scenario described above. The node on the surface was acting as the sink while the other nodes were competing to reserve the channel and send data. Data packets were generated according to a Poisson process to mimic a statistical event-driven data flow. Metrics collected via SUNSET included the maximal packets per packet time (pppt) supported by the different protocols to obtain a packet delivery ratio (PDR) higher than a threshold (set to either 0.95 or 0.85). The table also displays the corresponding packet end-to-end latency.

Table 1: Static scenario.

Metrics/Protocols	CSMA	T-Lohi	DACAP
PDR > 0.95 (Latency)	0.12 pppt (17.5 sec.)	0.08 pppt (13 sec.)	0.11 pppt (32 sec.)
PDR > 0.85 (Latency)	0.22 pppt (46.5 sec.)	0.16 pppt (19.3 sec.)	0.13 pppt (42 sec.)

During the tests at Pianosa SUNSET was also used to collect channel quality information, which was then used to simulate the same scenarios where the real acoustic channel information were fed to the Bellhop ray tracer. Simulation results and at-sea tests shows similar performance, which confirms the use of SUNSET as a pre-deployment tool for protocol performance evaluation.

An important lesson learned from the ACommsNet10 campaign was that a tool for acoustically controlling the various network components remotely and in real-time, was necessary. In fact, changes to the scripts were possible only via cabled/radio connections to the devices, if available. Otherwise their retrieval was needed. This prompted us to add such a tool, the back-seat driver, to SUNSET.

Routing. During the CMRE CommsNet'12 and CommsNet'13 at-sea campaigns and during the experimental activities of the EU-funded project CLAM project we used SUNSET to test various routing protocols. During these campaigns we tested SUNSET in networks with a higher number of nodes. Experiments refer to configurations of up to twelve, static and mobile, nodes deployed in the waters surrounding Palmaria, an island in the gulf of La Spezia, in Italy, in October 2012 and September 2013. Nodes were located distant enough that multi-hop routing was a necessity. Different routing solutions were implemented in SUNSET and evaluated at sea: Flooding-based, a channel-aware cross-layer routing, termed CARP [6], and an improved version of the solution described in [10]. Before the at-sea tests, all these solutions were first evaluated and tuned by means of simulations. A new feature introduced in SUNSET was used in the CommsNet'13, namely, the channel emulator. This allowed us to consider the overhead and delays introduced by the actual computational platforms before deployment at sea.

In this campaigns of experiments we tested SUNSET on different acoustic modems, i.e., the Evologics S2C R 18/34, WHOI FSK Micro-Modems and Teledyne Benthos. SUNSET allowed us flexible testing at sea, in that by just changing a few lines in the configuration script, we were able to vary the traffic load, the packet size, the number of nodes used in each test, the ID of the sink node, the transmission power of the modems, etc. In this case, we used SUNSET version 2.0, with the newly introduced back-seat driver mechanism, which allowed us to reconfigure the network as needed, and to start, stop, and monitor each test from remote, just via acoustic connections.

Sample results obtained through SUNSET in the CommsNet'12 campaign are shown in Table 2. Those figures refer to the testing of CARP in a scenario with 7 nodes. The two nodes most further apart from each other (about 3km) were used as sink and data source. The source node was set to generate one packet every 30s.

Table 2: Subset of the CARP results during CommsNet'12

Metrics →	PDR [%]	Latency [s]	Route length [hops]
Results →	89	67	1.6

Additionally, SUNSET was also used to collect channel measurements about the quality of the links between pairs of nodes, which were then used to evaluate the performance of the different routing protocols by means of channel replay-based simulations [34].

The SUNSET flexibility of use has been demonstrated by using it for the final demonstration of the EU FP7 CLAM project [36] conducted in the Norwegian fjord of Trondheim in May 2013. The objective of the experiment was to mimic a pipeline monitoring scenario. Seven underwater nodes were deployed in the fjord: Five of them at a depth ranging from 170 to 190 meters, accessible only through acoustic links, and two on the surface with a radio link to the control station on shore. Three of the bottom nodes were deployed following a line, as they were along a pipeline, and the other two spread across the oil well area. One of the surface nodes was deployed on the side of the Gunnerus ship, working as a mobile platform, again showing how SUNSET features can be transparently used also in mobile nodes. The remaining node was attached to a surface buoy at a depth of 15 meters. This buoy was used as a gateway to interact with the underwater network: To transmit the commands coming from the shore station and to report all measurements to shore. All the nodes were equipped with Kongsberg cNODE acoustic modems [47].

Three different routing protocols (Flooding, CARP [6] and SUN [48]) were investigated. The trial lasted for two weeks and all SUNSET functionalities were explored. The first week was mainly devoted to on shore testing to fix and tune all the protocol solutions according to the scenarios then tested at sea. An intensive use of the channel emulator allowed us to evaluate the implemented software when running on the same platform (Gumstix Overo) and connected to the same sensors (CO_2, temperature, pressure, vibration, etc.) then used in the fjord. During the second week all the nodes were deployed at sea. The SUNSET blacklisting capability was also exploited to create a multi-hop topology. The back-seat driver was extensively used to monitor and configure the different tests. The network operators were able to remotely operate the entire network from the control station on shore (or on board of the ship) continuously, day and night, with no unnecessary interruptions and without the need to retrieve or to access the nodes directly. More than 50.000 packets were transmitted during the trial and on average 30 seconds were needed to reconfigure the entire network. The back-seat driver was also used to start external processes, such as for remotely activating the acoustic release and for acquiring additional sensor measurements.

Table 3 shows some of the results obtained using an improved version of the CARP protocol, with respect to the one designed for CommsNet'12. Six nodes were used during this test: The surface buoy working as sink; three nodes generating data (the ship, a node on the pipe and another on the oil well); the remaining two nodes working as relays. One data packet was generated every 60 seconds at each data source resulting in one data packet injected in the network every 20 seconds. Two scenarios were considered: One with the mobile nodes on top of the pipeline (scenario 1) and another one with the mobile node on top of the oil well (scenario 2).

Table 3: Subset of the CARP results during Trondheim tests, May 2013

Scenarios ↓ / Metrics →	PDR [%]	Latency [s]	Route length [hops]
Scenario 1	82	84	1.3
Scenario 2	89	55	1.32

5.2. SUNSET back-seat driver

Following the lessons learned from the several campaigns performed in 2012 and 2013, the SUNSET back-seat driver mechanism has been significantly improved and extended. The current version of this fundamental SUNSET module was first tested during the NATO CMRE CommsNet13 experiment, the first at-sea experiment for the EU FP7 SUNRISE project [37]. It was conducted in the period September 9 to 22 off the coast of the Palmaria island. SUNSET was used to investigate the performance of five different MAC and five different routing solutions, making full use of the nodes available in the network. The use of the back-seat driver allowed us to remotely change the network configuration and switch from one test to another quick and easy. More than 70.000 packets were transmitted during the trial, switching among 55 different experimental setups. All reconfiguration commands were sent over acoustic links, without the need of any other form of connections, or of equipment retrieval.

Table 4 shows the performance of the SUNSET back-seat driver used to start/stop experiments with different modems on three statically deployed nodes. In the table, EVO, MM-1 and BEN refer to the use of Evologics modems, WHOI FSK Micro-Modems and Teledyne Benthos modems, respectively. The protocol stack included a CSMA MAC and an advanced flooding-based solution. MM-2, instead, refers to the use of the WHOI FSK Micro-Modems with a TDMA protocol at the MAC layer.

Table 4: Back-seat driver performance: 3 modems on 3 nodes.

	EVO	MM-1	MM-2	BEN
Delay [s]	12.79	48.12	23.5	24.84
Request attempts	1	2	1	1.7

The overhead added by the Micro-Modem to each data packet (a header of 2.8 seconds) results in quite long transmission delays and a larger collisions window. This translates into poor performance when a flooding-based or a random access solution are used. We therefore considered the use of a TDMA protocol for the WHOI Micro-Modem (MM-2) which resulted in much better performance with the need, however, of a fully synchronized network. The extra overhead introduced by Teledyne Benthos modems (a header of about 1 second) still impairs the use of a solution combining flooding-based routing and random access MAC, but leads to reasonable results. The Evologics modem showed the best performance in the considered scenario.

These results confirm the ability of the back-seat driver to control nodes and show the significant potential impact of hardware/software optimization on the overall system performance.

Table 5 shows the back-seat driver performance on a network composed by 9 nodes in a scenario in which an increasing number of nodes (from 1 to 8) was remotely activated and configured. The extra node in the network is the control station. In this case, Evologics S2CR 18/34 modems were used by the back-seat driver to instruct the nodes.

Table 5: Back-seat driver performance: 1 modem on 9 nodes.

	1	2	3	4	5	6	7	8
Delay [s]	7.6	19.7	22.4	22.5	36.3	50.8	45.4	55
Request attempts	1	1	1.5	2	2	2.5	2.5	2.75

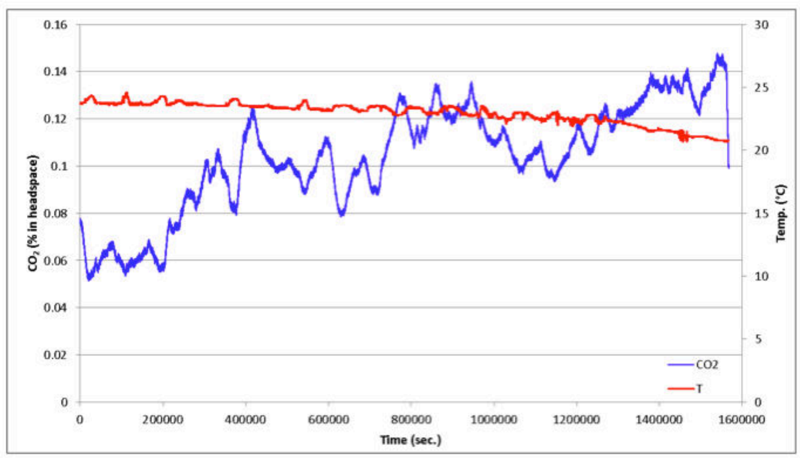
As expected, increasing the number of nodes to be activated results in longer delays and a higher number of attempts to instruct the network. In particular, more responses have to be received by the control station incurring in a higher traffic load and a higher number of packet collisions. However, the back-seat driver was successfully able to reconfigure all the nodes in less than a minute.

5.3. SUNSET for real-time remote control

SUNSET has functionalities that allow the user to control any hardware, e.g., sensors as well as mobile robots, remotely and in real time via acoustic communications. In this section we report two examples of these functionalities, concerning a static network of sensors for environmental monitoring and the control of underwater and surface vehicles.

CO2Net. SUNSET has been interfaced with a novel sensing platform, developed by the Geochemistry Department of the Università di Roma "La Sapienza," for monitoring the temperature and the concentration of methane and CO_2 dissolved in water. The combination of sensing, networking and communication capabilities (supported by the use of an acoustic modem), allowed the realization of a prototype remotely-operated sensing device, CO2Probe (Figure 4a), which can be used to perform accurate real-life monitoring of underwater CO_2 storage infrastructures. During the CMRE ACommsNet11 at-sea campaign, we investigated the

(a) CO2Probe and acoustic modem.



(b) Subset of the data monitored in the harbor of La Spezia, Summer/Fall 2011.

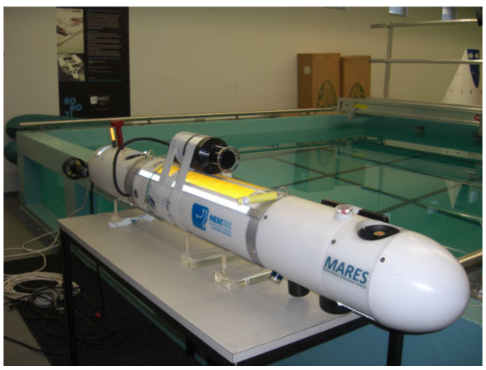
Figure 4: CO2Probe and measurement from ACommsNet11.

possibility to remotely instruct the CO2Probe and to collect the measurements in real time. The CO2Probe, locally running SUNSET on an embedded device, was connected to one of the underwater nodes in a network of 4 nodes. The other nodes were used as relays since, according to the acoustic channel condition, a direct acoustic link to the sensing device was not always available. For these three nodes, SUNSET was running on a PC on the shore station, which was used as control station to instruct the probe and collect the data.[6] A network of cooperative devices (CO2Net) was created. Using this system, we were able to easily collect the measurements (see Figure 4) and to configure and change the monitoring parameters in real time

[6]The protocol stack at each node was running CSMA [11] and flooding.

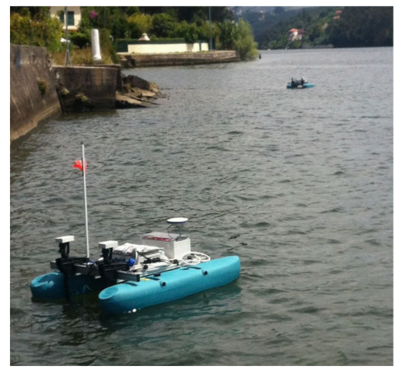
via acoustic links. The same approach can be used for any other sensing device or monitoring scenario, thus making SUNSET a key enabler for innovative monitoring applications that can cover large areas.(Details about the CO2Probe, CO2Net and further results can be found in [35].)

AUV and ASV control. During the first months of 2012, SUNSET was interfaced with the MARES AUV, produced by INESC TEC/University of Porto (Figure 5a). A new SUNSET driver was designed and implemented allowing to remotely operate the vehicle underwater using acoustic communications. Preliminary tests were performed in the water tank at the Oceansys laboratory. A Gumstix device was used to run SUNSET inside the vehicle and Evologics modems S2C R 18/34 were used for acoustic transmissions.



(a) MARES AUV with acoustic modem.

(b) NIB.

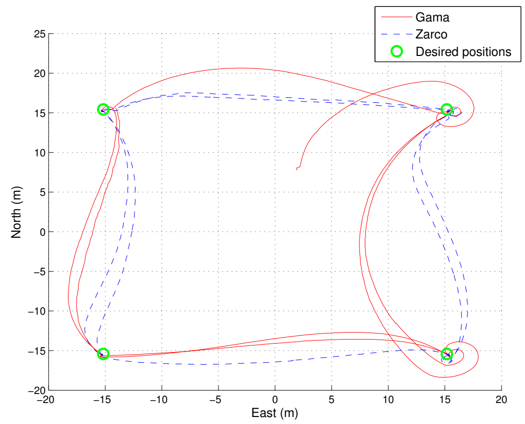


(c) Autonomous Surface Vehicle in water.

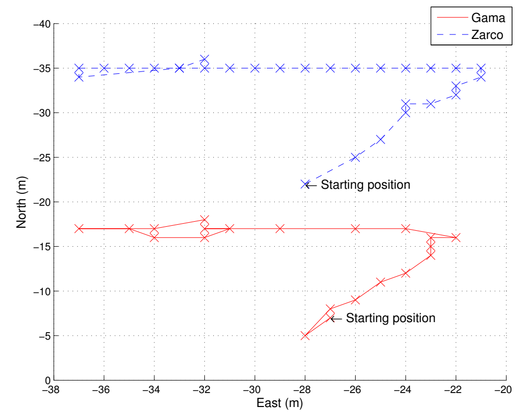
Figure 5: SUNSET integration with INESC TEC/University of Porto ASV.

In Summer 2012 SUNSET was integrated with INESC TEC/University of Porto ASV (Figure 5c) and with the Navigation and Instrumentation Buoy (NIB, Figure 5b). Initial in-field tests were conducted at the Douro river in Porto where four stations were deployed: A control station on shore with an acoustic modem in water, a NIB station and two ASVs. Gumstix devices were installed in the different nodes to run SUNSET and Evologics modems S2C R 18/34 were used for the acoustic transmissions.

Using SUNSET, the nodes cooperated to support the vehicle movements according to a predefined path or to a given formation control algorithm (Figure 6). Figure 6a shows the case where the two surface vehicles (Zarco and Gama) are exchanging acoustic messages to coordinate their moment along the corners of a square. Figure 6b shows the actual positions of Zarco and Gama when instructed by the control station to move in formation. Commands are sent by the control station to the two vehicles and position feedbacks are transmitted by the vehicles to the control station. All transmission are acoustic, with no use of radio communications. A Slotted CSMA protocol specifically modified for this scenario was implemented in SUNSET. (Additional details can be found in [42].)



(a) Zarco and Gama moving along the corners of a square.



(b) Zarco and Gama trajectories while coordinating. The crosses indicate the reception of commands from the control station.

Figure 6: Vehicle control and coordination via acoustic links.

6. Conclusions

We have presented SUNSET, a new ns-2 based open source framework for underwater sensor network simulation, emulation and at-sea testing. We described the overall SUNSET framework and its main modules together with some of the most interesting results from SUNSET-enabled at-sea experiments conducted in recent years. The results obtained at sea and all the lessons learned over the years have driven the design of additional, improved and enhanced functionalities. This led to the creation of a more complete, reliable and efficient platform, SUNSET version 2.0, to be shared with the underwater community to simulate, emulate and test at sea underwater protocol solutions. Using SUNSET, acoustic communications and networking capabilities can be combined with sensing, navigation and control operations in order to provide a complete framework for real-life testing of complete underwater monitoring solutions. The latest version of SUNSET introduces an extremely limited overhead when running on small devices, which can be energy efficient, compact, and inexpensive. An additional delay on the order of milliseconds and a few extra bits of data added to the message packet header are the only toll to pay, making SUNSET a truly effective, stable and proven solution for at-sea testing. SUNSET has been interfaced with several commercial acoustic modems, sensing platforms, surface and underwater mobile robots. Using SUNSET, researchers and developers are able to quickly identify underperforming or malfunctioning communication modules, before and during at-sea tests. Additionally, remote and real time control on the underwater devices after deployment is provided by a "back-seat driver" feature, reducing the cost, delay and logistic complexity of at-sea campaigns.

Acknowledgments

This work was supported in part by the EU FP 7 ICT project SUNRISE “Sensing, monitoring and actuating on the UNderwater world through a federated Research InfraStructure Extending the Future Internet.” The authors gratefully acknowledge (in alphabetical order) the CLAM project, Evologics GmbH, the Geochemistry Department of the Università di Roma “La Sapienza,” Graaltech, the INESC TEC/University of Porto group, the LSTS group at University of Porto, the NATO STO Centre for Maritime Research and Experimentation, the SUNRISE project, the Wireless Networks and Embedded Systems Laboratory at SUNY Buffalo and the Woods Hole Oceanographic Institution for their valuable feedback and for their support and help during tests and experimentation.

References

[1] L. Lanbo, Z. Shengli, C. Jun-Hong, Prospects and problems of wireless communication for underwater sensor networks, Wireless Communications and Mobile Computing, Special Issue on Underwater Sensor Networks 8 (8) (2008) 977–994. doi:http://dx.doi.org/10.1002/wcm.v8:8.

[2] J. Heidemann, M. Stojanovic, M. Zorzi, Underwater sensor networks: Applications, advances and challenges, Philosophical Transactions of the Royal Society A 370 (2012) 158–175.

[3] D. Pompili, I. F. Akyildiz, Overview of networking protocols for underwater wireless communications, IEEE Communications Magazine 47 (1) (2009) 97–102.

[4] M. Chitre, S. Shahabudeen, M. Stojanovic, Underwater acoustic communications and networking: Recent advances and future challenges, Marine Technology Society Journal 42 (2008) 103–116.

[5] P. Casari, M. Zorzi, Protocol design issues in underwater acoustic networks, Elsevier Computer Communications 34 (17) (2011) 2013–2025.

[6] S. Basagni, C. Petrioli, R. Petroccia, D. Spaccini, Channel-aware routing for underwater wireless networks, in: Proceedings of MTS/IEEE OCEANS 2012, Yeosu, Korea, 2012, pp. 1–9.

[7] C. Petrioli, R. Petroccia, M. Stojanovic, A comparative performance evaluation of MAC protocols for underwater sensor networks., in: Proceedings of MTS/IEEE OCEANS 2008, Quebec City, Quebec, Canada, 2008, pp. 1–10.

[8] P. Casari, B. Tomasi, M. Zorzi, A comparison between the tone-lohi and slotted fama mac protocols for underwater networks., in: Proceedings of MTS/IEEE OCEANS 2008, Quebec City, Quebec, Canada, 2008, pp. 1–8.

[9] R. Diamant, L. Lampe, A hybrid spatial reuse mac protocol for ad-hoc underwater acoustic communication networks, in: Communications Workshops (ICC), 2010 IEEE International Conference on, Cape Town, South Africa, 2010, pp. 1–5.

[10] J. ALves, G. Zappa, Low overhead routing for underwater acoustic networks, in: Proceedings of IEEE OCEANS 2011, Santander, Spain, 2011, pp. 1–10.

[11] S. Basagni, C. Petrioli, R. Petroccia, M. Stojanovic, Optimized packet size selection in underwater WSN communications, IEEE Journal of Oceanic Engineering 37 (3) (2012) 321–337.

[12] B. Tomasi, G. Zappa, K. McCoy, P. Casari, M. Zorzi, Experimental study of the space-time properties of acoustic channels for underwater communications, in: Proceedings of IEEE OCEANS 2010, Sydney, Australia, 2010, pp. 1–9.

[13] C. Petrioli, R. Petroccia, J. Potter, Performance evaluation of underwater mac protocols: From simulation to at-sea testing, in: Proceedings of IEEE OCEANS 2011, Santander, Spain, 2011, pp. 1–10.

[14] C. Petrioli, R. Petroccia, J. Shusta, L. Freitag, From underwater simulation to at-sea testing using the ns-2 network simulator, in: Proceedings of IEEE OCEANS 2011, Santander, Spain, 2011, pp. 1–9.

[15] C. Petrioli, R. Petroccia, SUNSET: Simulation, emulation and real-life testing of underwater wireless sensor networks, in: Proceedings of IEEE UComms 2012, Sestri Levante, Italy, 2012, pp. 1–12.

[16] SENSES Lab, SUNSET: Sapienza university networking framework for underwater simulation, emulation and real-life testing.
URL `http://reti.dsi.uniroma1.it/UWSN_Group/index.php?page=sunset`

[17] C. Petrioli, R. Petroccia, D. Spaccini, SUNSET version 2.0: Enhanced framework for simulation, emulation and real-life testing of underwater wireless sensor networks, in: Proceedings of ACM WUWNet 2013, Kaohsiung, Taiwan, 2013, pp. 1–8.

[18] The VINT Project, The ns Manual, http://www.isi.edu/nsnam/ns/, 2002.

[19] N. Baldo, F. Maguolo, M. Miozzo, M. Rossi, M. Zorzi, ns2-MIRACLE: A modular framework for multi-technology and cross-layer support in network simulator 2, in: Proceedings of the 2nd International Conference on Performance Evaluation Methodologies and Tools, ValueTools 2007, Nantes, France, 2007, pp. 1–8.

[20] R. Petroccia, D. Spaccini, Implementing a back-seat driver to remotely control the experiments in an underwater acoustic sensor network, in: Proceedings of MTS/IEEE OCEANS 2013, Bergen, Norway, 2013, pp. 1–9.

[21] P. Xie, Z. Zhou, Z. Peng, H. Yan, T. Hu, J.-H. Cui, Z. Shi, Y. Fei, S. Zhou, Aqua-sim: An ns-2 based simulator for underwater sensor networks, in: PProceedings of MTS/IEEE OCEANS 2009, Biloxi, Mississippi, USA, October, 2009, pp. 1–7.

[22] Z. Peng, J.-H. Cui, B. Wang, K. Ball, L. Freitag, An underwater network testbed: Design, implementation and measurement, in: Proceedings of ACM WUWNet 2007, Montréal, Quebec, Canada, 2007, pp. 1–8.

[23] Z. Peng, Z. Zhou, J.-H. Cui, A. J. Shi, Aqua-Net: An underwater sensor network architecture: Design, implementation, and initial testing, in: Proceedings of MTS/IEEE OCEANS 2009, Biloxi, MS, 2009, pp. 1–8.

[24] S. Shahabudeen, M. Chitre, M. Motani, A. L. Y. Siah, Unified simulation and implementation software framework for underwater MAC protocol development, in: Proceedings of MTS/IEEE OCEANS 2009, Biloxi, MS, 2009, pp. 1–9.

[25] S. N. Le, Z. Peng, J.-H. Cui, H. Zhou, J. Liao, Sealinx: A multi-instance protocol stack architecture for underwater networking, in: Proceedings of ACM WUWNet 2013, Kaohsiung, Taiwan, 2013, pp. 1–5.

[26] Gumstix, Inc., http://www.gumstix.com/.

[27] M. Chitre, I. Topor, T.-B. Koay, The UNET-2 modem - an extensible tool for underwater networking research, in: Proceedings of MTS/IEEE OCEANS 2012, Yeosu, Korea, 2012, pp. 1–7.

[28] D. Torres, J. Friedman, T. Schmid, M. B. Srivastava, Software-defined underwater acoustic networking platform, in: Proceedings of ACM WUWNet 2009, Berkeley, CA, 2009, pp. 7:1–7:8.

[29] R. Masiero, S. Azad, F. Favaro, M. Petrani, G. Toso, F. Guerra, P. Casari, M. Zorzi, DESERT underwater: an NSmiracle-based framework to DEsign, simulate, emulate and realize test-beds for underwater network protocols, in: Proceedings of IEEE OCEANS 2012, Yeosu, Korea, 2012, pp. 1–10.

[30] R. Petroccia, D. Spaccini, Comparing SUNSET and DESERT frameworks for in field experiments in an underwater acoustic

network, in: Proceedings of MTS/IEEE OCEANS 2013, Bergen, Norway, 2013, pp. 1–10.

[31] R. Urick, Principles of Underwater Sound, McGraw-Hill, 1983.

[32] M. Porter *et al.*, Bellhop code, URL: http://oalib.hlsresearch.com/Rays/index.html.

[33] F. Guerra, P. Casari, M. Zorzi, World ocean simulation system (WOSS): A simulation tool for underwater networks with realistic propagation modeling, in: Proceedings of ACM WUWNet 2009, Berkeley, CA, 2009, pp. 1–8.

[34] S. Basagni, C. Petrioli, R. Petroccia, D. Spaccini, Channel replay-based performance evaluation of protocols for underwater routing, in: Proceedings of IEEE MTS/OCEANS 2014, Saint John, NL, Canada, 2014, pp. 1–7.

[35] A. Annunziatellis, S. Graziani, S. Lombardi, C. Petrioli, Petroccia, CO2Net: A marine monitoring system for CO_2 leakage detection, in: Proceedings of IEEE OCEANS 2012, Yeosu, Korea, 2012, pp. 1–7.

[36] CLAM: Collaborative embedded networks for submarine surveillance.
URL `http://clam.ewi.utwente.nl/`

[37] SUNRISE: Sensing, monitoring and actuating on the UNderwater world through a federated Research InfraStructure Extending the Future Internet, http://fp7-sunrise.eu/.

[38] R. Jurdak, A. G. Ruzzelli, G. M. P. O'Hare, Design considerations for deploying underwater sensor networks, in: Proceedings of SensorComm 2007, Valencia, Spain, 2007, pp. 227–232.

[39] R. Otnes, P. van Walree, T. Jenserud, Validation of replay-based underwater acoustic communication channel simulation., IEEE Journal on Oceanic Engineering 38 (4) (2013) 689–700.

[40] LSTS, University of Porto, Manta communications gateway.
URL `http://lsts.fe.up.pt/vehicles/manta`

[41] O. Kebkal, K. Kebkal, M. Komar, Development of upper-layer protocols with S2CR acoustic modems emulator, in: Proceedings of MTS/IEEE UComms 2012, Sestri Levante, Italy, 2012, pp. 1–6.

[42] N. A. Cruz, B. M. Ferreira, O. Kebkal, A. C. Matos, C. Petrioli, R. Petroccia, D. Spaccini, Investigation of underwater acoustic networking enabling the cooperative operation of multiple heterogeneous vehicles, Marine Technology Society Journal 47 (2013) 43–58.

[43] J. Alves, J. Potter, G. Zappa, P. Guerrini, R. Been, A testbed for collaborative development of underwater communications and networking, in: In Proceedings of IEEE Military Communications Conference (MILCOM'12), 2012, pp. 1–8.

[44] I. Calabrese, R. Masiero, P. Casari, L. Vangelista, M. Zorzi, Embedded systems for prototyping underwater acoustic networks: The DESERT underwater libraries on board the PandaBoard and NetDCU, in: Proceedings of MTS/IEEE OCEANS 2012, Hampton Roads, Virginia, 2012, pp. 1–8.

[45] A. Syed, W. Ye, J. Heidemann, Comparison and evaluation of the T-Lohi MAC for underwater acoustic sensor networks, IEEE Journal on Selected Areas in Communications 26 (9) (2008) 1731–1743.

[46] B. Peleato, M. Stojanovic, Distance aware collision avoidance protocol for ad-hoc underwater acoustic sensor networks., IEEE Communications Letters 11 (12) (2007) 1025–1027.

[47] Kongsberg maritime, Instruction manual cNode transponder.
URL `http://www.km.kongsberg.com/ks/web/nokbg0397.nsf/AllWeb/4ADD212486A1B94EC125780000355234/`

[48] G. Toso, R. Masiero, P. Casari, O. Kebkal, M. Komar, M. Zorzi, Field experiments for dynamic source routing: S2C EvoLogics modems run the SUN protocol using the DESERT underwater libraries, in: Proceedings of MTS/IEEE OCEANS 2012, 2012, pp. 1–10.