# SUNSET version 2.0: Enhanced Framework for Simulation, Emulation and Real-life Testing of Underwater Wireless Sensor Networks

Chiara Petrioli
Sapienza University of Rome
113, Via Salaria
Rome, Italy
CINI, Roma, Italy
WSENSE s.r.l., Rome, Italy
petrioli@di.uniroma1.it

Roberto Petroccia
Sapienza University of Rome
113, Via Salaria
Rome, Italy
CINI, Roma, Italy
WSENSE s.r.l., Rome, Italy
petroccia@di.uniroma1.it

Daniele Spaccini
Sapienza University of Rome
113, Via Salaria
Rome, Italy
CINI, Roma, Italy
spaccini@di.uniroma1.it

## ABSTRACT

We describe the newest design and development of the Sapienza University Networking framework for underwater Simulation Emulation and real-life Testing (SUNSET), a powerful toolkit for implementation and testing of novel protocol solutions for underwater sensor networks. SUNSET enables testing and enhancement of underwater solutions using a controlled simulation environment. The very simulation code can be then transparently ported to various real hardware and underwater platforms for protocol emulation and actual in field testing. SUNSET was the first open source framework for seamless simulation, emulation and actual at sea testing of novel underwater systems. The first SUNSET was presented in 2011 [1], improved in 2012 [2] and freely released to the research community in May 2012. Since then, SUNSET has been significantly improved and validated through several trials at sea. New modules have been created and the functionalities of existing ones have been extended and enhanced. The new SUNSET version 2.0 has been recently released to the community (October 2013 [3]). This paper discusses in details the SUNSET 2.0 features and modules, and report about the latest in field tests where SUNSET 2.0 has been used.

## Keywords

Underwater acoustic networks, simulation, emulation, ns-2, at sea testing, SUNSET.

## 1. INTRODUCTION

Underwater wireless sensor networks have become an important area of research and the enabling technology of a wide range of emerging applications: Monitoring and discovery of the marine environment, maintenance and conservation of marine archaeological sites, remote control of submarine oil extraction, underwater safe $CO_2$ storage, coastline protection, and prediction of underwater seismic and volcanic events, etc. [4]. The need of underwater communications solutions turning existing static and mobile platforms into cooperative underwater monitoring and control systems is therefore fast increasing. Despite the growing number of contributions on the design of protocols for underwater networks, research is still not moving at the speed required by emerging applications. The cost and logistic complexity of in field experiments, as well as the low accuracy and heterogeneity of existing simulators, pose a major barrier for solution validation, evaluation and benchmarking. Major changes in the code are typically needed to bring solutions to new platforms; long times and significant effort are spent in preparing expensive in field experiments, without a toolchain of pre-deployment and deployment tools able to identify in advance risks, malfunctioning and underperforming solutions. As a result, experimental campaigns are often devoted to solving problems identified at the time of test. To address these issues, we have developed and extensively validated in field a new complete framework for Underwater Acoustic Sensor Networks (UASNs) simulation, emulation and real-life testing. The framework has been originally presented and tested in [1, 5] and it has been significantly extended and improved in the recent past [2], leading to the creation of SUNSET, the *Sapienza University Networking framework for underwater Simulation Emulation and real-life Testing*. SUNSET version 1.0 has been freely released in May 2012 as the first framework based on open source software enabling seamless simulation, emulation and real-life testing of communication protocols and underwater systems. SUNSET builds on the open source and well known network simulator ns-2 [6] and on its extension ns2-Miracle [7]. Using SUNSET, researchers and developers can implement, evaluate and ameliorate their solutions by means of simulations. More important, they can use the very same code used for simulations for emulation and in field testing (e.g., at sea trials), where real modems are used for data transmission and additional external devices can be integrated to the nodes for sensing and navigation. SUNSET has been created with the objective of providing a flexible, reliable and efficient standard framework for running protocol stacks that can be easily shared, used and modified by the underwater research community. Such a framework was


*WUWNet'13*, November 11-13, 2013, Kaohsiung, Taiwan

http://dx.doi.org/10.1145/2532378.2532424

the missing key asset to speed up innovation in the field. SUNSET has been extensively validated and evaluated in the recent past through several trials at locations of different nature (sea, river, lake, fjord), and considering different network configurations. In field tests have shown opportunities for performance improvements and desirable additional functionalities for the core architecture. New SUNSET modules have therefore been designed and implemented and the existing functionalities have been extended and enhanced. The resulting SUNSET version 2.0 [3] has been recently released open source (October 2013). This is the version described in details in this paper.

Among new core functionalities, the new SUNSET version includes a novel mechanism, called *back-seat driver*, to remotely control and operate the entire underwater network via acoustic links [8]. Developers and researchers can therefore more easily run different experiments. They can reconfigure the network and change protocol parameter and test settings in real-time, without the need to retrieve underwater devices after their deployment. The new back-seat driver module allows the developers to remotely operate the network and to run the experiments without interruptions thus saving time and money during in field campaigns.

The rest of the paper is organized as follows. Section 2 describes the state-of-the-art of testbeds and emulation frameworks for underwater networks. A detailed description of the innovations introduced by SUNSET version 2.0 with respect to version 1.0 is presented in Section 3. In Section 4 we report recent results from in field SUNSET 2.0-based experiments. Concluding remarks are provided in Section 5.

## 2. RELATED WORKS

Investigation and in field testing of the performance of communication protocols for UASNs has been supported by multiple frameworks [9, 10, 11, 12]. These frameworks are paving the way for a new paradigm [13] where software defined communication protocol stacks are combined with commercial or software defined acoustic modems to provide adaptive and reliable networking for all the different classes of underwater application domains. More specifically, in [9] the Aqua-Lab testbed is presented. Software Application Programming Interfaces (APIs) are defined as a middleware for the interaction with the WHOI Micro-Modem. Aqua-Lab supports different network topologies, propagation delays, and signal attenuation and it has been used to conduct a set of experiments in both field and lab environments. In [10], the authors present the Aqua-Net architecture. It assumes a layered structure with the possibility for cross-layer optimization and supports the use of different acoustic modems at the physical layer (i.e., WHOI Micro-Modems and Teledyne Benthos modems). The Aqua-Net architecture has been used in field by running it on an embedded system (Gumstix device [14]) and by using WHOI Micro-Modems for acoustic transmissions. A similar approach is described in [11], where the authors propose a unified simulation and emulation architecture for underwater MAC protocol development. It allows researchers and testers to use the same C code implementation of a MAC protocol both for simulations and on the real modem without the need of any code re-writing. ARL OFDM modems have been considered for acoustic transmissions.

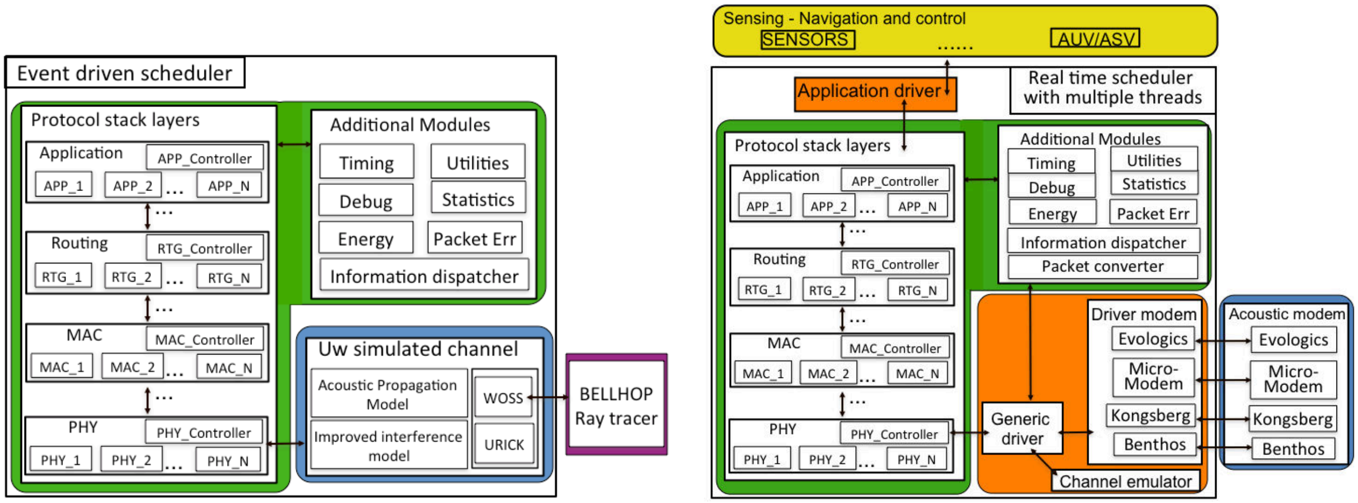
A fundamental limit of all these solutions is that proprietary architectures and software have to be studied by the developers to use these systems, which is usually non trivial. In addition, such architectures are not open enough to support the heterogeneous hardware platforms typically used in field, limiting their use for in field experiments. An extension of [11] has been proposed in [12] where the UNET-2 framework is presented. It 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 in field tests. UNET-2 makes significant progresses in the direction of software defined acoustic modems, but it provides little support for the implementation and evaluation of software defined protocol stacks.

The first work considering the use of an open-source platform to create a simulation/emulation tool for UASNs has been presented in [15]. A new platform, named UANT, is proposed that is based on a GNU radio interfaced with a PC running TinyOS (or TOSSIM in case of emulation/simulation). The GNU radio is an open-source software development toolkit that provides signal processing blocks to implement software radios. It has been used by the authors to transmit the waveform on the acoustic channel using an acoustic transducer. TinyOS and TOSSIM are generally used to implement the communication protocol stack for terrestrial sensor networks. TinyOS is used when running on real wireless sensor devices, TOSSIM instead when running simulations. The same code simulated by TOSSIM can be ported to work on TinyOs. Although the proposed UANT architecture is interesting, it is difficult to use it for real in field experimentation. In fact, the GNU radio software is computationally expensive and it needs to run on a PC, which is unlikely to be used during real-life experiments due to housing constraints. Moreover, to run on TinyOS, the protocol design has to follow rules which reflect features of IEEE 802.15.4 devices and that differ from features and constraints of underwater nodes and underwater communication devices.

Recently, the DESERT framework has been proposed in [16], which is a simulation/emulation tool based on the idea we presented in [1, 5]. DESERT is also based on ns-2 and ns2-Miracle, and implements an approach similar to that proposed by SUNSET focusing on a large library of MAC and routing solutions. Being based on the same base software, there is a high compatibility and interoperability between the protocol solutions and the modules developed in SUNSET and DESERT when running in simulation mode. However, as presented in [17], DESERT lacks several critical features to efficiently pass from simulations to real-life tests, resulting in lower in field performance and test support.

## 3. SUNSET VERSION 2.0

The first version of the Sapienza University Networking framework for underwater Simulation, Emulation and real-life Testing, also known as SUNSET 1.0, was presented in 2011 providing a complete toolkit to seamlessly simulate, emulate and test at sea communication protocols for UASNs. SUNSET is based on the open source and well known network simulator ns-2 [6] and on its extension ns2-Miracle [7], both




(a) SUNSET modules in simulation mode.

(b) SUNSET modules in emulation mode.

Figure 1: SUNSET architecture: Simulation and emulation modes.

largely used by the research community. We decided to extend these software tools rather than implementing a new simulation framework from scratch to limit the effort needed by researchers and developers already familiar with ns-2 for implementing and testing their underwater solutions in SUNSET. The SUNSET framework has been also designed to separate the protocol stack implementation from the external modules controlling the use of real hardware when running in emulation and test mode. Using our framework, anyone implementing his/her solutions for simulations can eventually use the very same code in emulation and field test mode, with the option of using real acoustic modems for data transmission and additional external devices for sensing and navigation operations. Moreover, in SUNSET, researchers and developers can remotely control in real-time the operations of the entire network using acoustic links. Statistics and measurements regarding the on-going experiments, and information on the status of the nodes are readily available to the users. This allows to remotely monitor the running test and to adjust and reconfigure it according to the user needs. In particular, the network topology can be modified (remotely activating or deactivating only a subset of the underwater nodes), the settings of the selected protocol and experiment parameters can be tuned (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.), the selected protocol stack can be changed, etc.

SUNSET version 2.0 has all the functionalities of SUNSET 1.0 and enhances them to perform accurate simulations, at the same time significantly improving support for emulation and real-life testing (Figure 1). The new version of SUNSET maintains all the Utilities, Debug and Statistics modules of SUNSET 1.0, and the Timing module that models delays introduced by the use of real hardware. It also maintains and enhances the additional modules to convert the ns-2 data packets in streams of bytes compressed for actual in water transmissions as well as the modules that abstract the interaction with external hardware, i.e., the acoustic modems, sensing platforms and vehicle navigation systems.

SUNSET version 2.0 code is more flexible and better organized, resulting in a less steeper learning curve. All the modules have been grouped in three main folders: **Core components**, which contains all the main modules needed for both simulations and emulation tests; **network protocols**, implementing all the different protocol stack solutions together with additional components to evaluate the protocol solutions, and **emulation components**, providing the additional support to seamlessly run the network solutions in emulation and field test mode.[1] If the user wants to use SUNSET only to run simulations, none of the modules and libraries provided in the Emulation components folder needs to be compiled and loaded.

With respect to the previous version, new modules have been implemented and several of the old ones have been improved to provide a more robust and flexible toolkit. The main changes introduced with the new release of SUNSET include:

- An improved real-time scheduler.
- A publish-subscribe pattern to exchange information among different modules of the protocol stack or between the stack and external devices.
- New functionalities added to the PHY layer providing the possibility to remove direct links from the network (node blacklist).
- Different packet error models: Static PER and PER related to BPSK and FSK modulations.

[1]Emulation components are currently provided as compiled libraries for the X86 architecture. Additional libraries to run SUNSET on different architectures can be requested to SUNSET developers.

- A novel channel emulator allowing to debug and to investigate the overall system performance when running in emulation mode, without the need of any real communication device.
- A novel accurate energy consumption model.
- A new MAC protocol (CSMA with no acknowledgments).
- A novel trace module.
- A statistic module to collect key metrics for the evaluation of the performance of protocol solutions developed for the SUNSET framework.
- New classes to share position information among the different modules (including the channel emulator).
- A new Transport module supporting the use in SUNSET of the ns2-Miracle CBR module when running in both simulation and emulation mode.
- A packet converter for the CBR packet header to be used when running in emulation mode.
- New connection classes implementing the separation between client and server modules.

In preparing the release of the new SUNSET, great effort has been devoted to provide detailed documentation and guidelines on how to use it, and on how to develop new solutions within SUNSET. Several scripts and examples have been prepared to use the system in both simulation and emulation mode, considering different network and protocol stack configurations. A detailed description of how to download and install SUNSET and of how to implement new modules is also provided.

To download, install and use SUNSET version 2.0 two different options are currently provided:

- **Virtual machine**. A complete virtual machine that already includes SUNSET and all the required software. It is based on a light and minimal Linux Debian distribution to reduce the overall size of the virtual environment ($\sim$ 1.5GB). This solution is strongly recommended by the authors.
- **SVN repository**. It allows to download the SUNSET code, libraries, scripts and examples. All the additional software (ns-2, ns2-Miracle, cross-compile environment, etc.) have to be manually installed by the user.

Moreover, since the new version of ns2-Miracle cannot properly work with the latest version of ns-2 (ns-2.35), due to a problem when loading the dynamic libraries, a patch to fix this issue has been prepared by the SENSES Lab underwater group, in collaboration with Evologics GmbH [18], and it is available together with the novel version of our framework on the SUNSET web page [3]. It is now possible to use ns2-Miracle and therefore SUNSET with the latest version of ns-2.

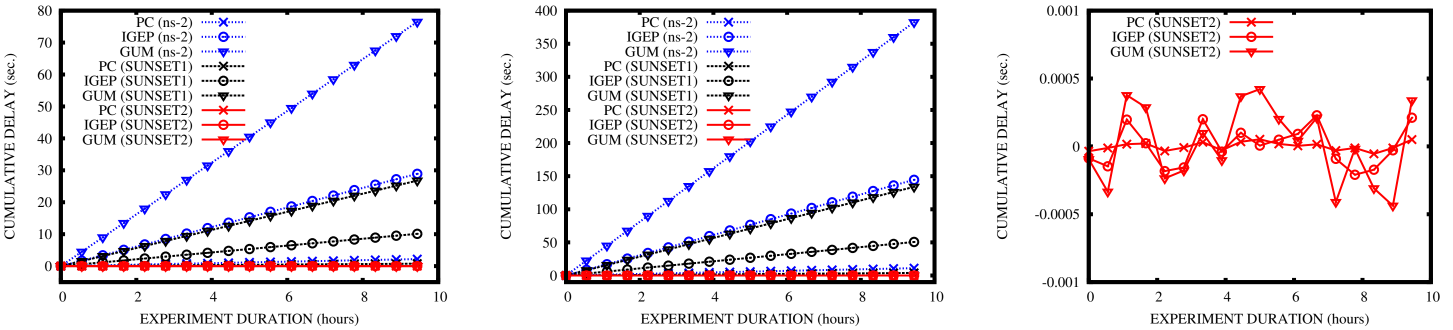
## Additional modules and improvements

In this section we describe the main additional modules and improvements introduced by SUNSET 2.0.

### Real-time scheduler

The first version of SUNSET real-time scheduler was an improvement of the basic ns-2 real-time scheduler to support interaction among multiple threads and to reduce the system CPU usage. In the new version of SUNSET, this real-time scheduler has been further enhanced to compensate for any computational delay added by the specific platform used to run the software and to increase the scheduler accuracy, as presented in [17]. Each time an event has to be scheduled a computational delay is introduced by the hardware platform that executes the event scheduling operations. We have estimated that using the ns-2 real-time scheduler the delay added to create the event, to add it to the ordered structure of the scheduler, and to remove it from the scheduler at the proper time is about: 0.065 milliseconds on the Desktop PC (CPU:Core(TM)2 Quad CPU Q6600 @2.40GHz, RAM: 4 GB), 0.75 milliseconds on the IGEPv2 board (CPU:ARM Cortex-A8 DM3730 @1Ghz, RAM: 512 MB) and 1.95 milliseconds on the Gumstix board (CPU:Marvell PXA270 @600MHz, RAM:128MB low-power DDR2). Using the first version of the SUNSET real-time scheduler these delays had already been reduced by 65%. Both ns-2 and the first SUNSET real-time scheduler, however, fail to compensate for these delays that are cumulatively added to the system clock. This means that if a periodic timer is used by a protocol implementation and if such timer is restarted whenever it expires, each restart cumulatively adds a HW-dependent delay to the system clock. Having the system running on different platforms, even if the devices are all equipped with an accurate clock and are all synchronized, the different computational delays desynchronize event scheduling in the network. Such desynchronization increases over time. The user has to actively compensate for these delays, which could make the software development complex. The new SUNSET real-time scheduler has been improved to automatically compensate for these delays, without requiring any additional effort by the user.

Figure 2 shows the effect of the computational delay on the three considered platforms when using the basic ns-2 real-time scheduler and the previous and the new version of the SUNSET real-time scheduler, named in the following as SUNSET1 and SUNSET2, respectively. Figure 2a shows the case where one event is periodically scheduled every 1 second and Figure 2b the case where one event is periodically scheduled every 0.2 seconds. The latter can mimic a possible periodic reading of data from a sensor or from the navigation/telemetry system of a vehicle. We can see that using the ns-2 and SUNSET1 real-time schedulers the cumulative delay introduced by the platform operations increases over time (up to several tens of seconds) depending on the platform computational power. SUNSET2 instead compensates for the additional delays, resulting in negligible (i.e., near 0) clock difference, not increasing over time. Figure 2c zooms into the case where SUNSET2 schedules a periodic event every 0.2 seconds. We can clearly see that the clock difference for each event is always smaller than 0.5 milliseconds.




(a) One event generated every 1 sec.

(b) One event generated every 0.2 sec.

(c) Zoom into one event generated every 0.2 sec.

Figure 2: Cumulative computational delay added by the different platforms.

**Information dispatcher**

The information dispatcher module implements the publish-subscribe pattern. It allows to share information among the different layers and modules of the protocol stack while keeping the high degree of flexibility and modularity given by the layered architecture. Each module providing information to or requesting information from other modules can register itself with the information dispatcher. It then informs the dispatcher about the data it can provide and the data it is interested in. The dispatcher collects the information from the different components and notifies any update to all the modules that are interested in them. All modules can also request information on demand or ask for periodic updates. Each information is timestamped in order to check its freshness. Moreover, if multiple components are providing the same information or if there are modules requesting information requiring some processing on the data provided by the other components, the information dispatcher can take care of that. This flexible mechanism is particularly useful when different external modules need to interact to provide a more complete picture of what is happening in the system, in order to decide the actions to perform. The information dispatcher can be completely configured via TCL scripts adding new resources to be shared by the different modules without any need to recompile the implemented code.

**Enhanced PHY layer**

New features have been added to the physical layer to support the design of desired network topologies when investigating network protocol performance. The novel SUNSET PHY module implements the possibility to remove undesired links from the network via software, i.e., nodes can blacklist each other. If two nodes blacklist each other then the link connecting these nodes is removed from the network, i.e., all the packets coming from the blacklisted nodes will be discarded at the physical layer. If only one node blacklists the other, an asymmetric link is introduced in the network. This new feature is particularly useful to properly set the desired network topology and link quality. It can be used not only when running simulation and in lab emulation tests but also when running real in field experiments. Consider for instance the case in which users want to test a routing protocol in a scenario where all the nodes deployed at sea are able to talk to each other, either because a good acoustic underwater channel is temporarily available or because it is not possible to deploy the nodes further apart from each other. In such a scenario, a multiple-hop logical topology can be enforced by blacklisting unwanted links. Using the new SUNSET PHY, packets coming from blacklisted nodes will be discarded and will be considered as additional noise and interference at the PHY layer. Everything is done in a transparent way to the MAC and routing layers. Additionally, the PHY layer supports the use of multiple transmission power levels when running simulations to more easily develop and test power management solutions.

**Packet error model**

A new packet error model has been implemented in SUNSET that supports different Packet Error Rates (PERs) and can be easily extended to implement novel error models at the network or link level. The proposed model allows to easily configure the network and link quality according to the selected scenarios or based on channel traces collected during in field activities. This module is mainly meant for simulations but it could also be used when the system is running in emulation mode if the user wants to reproduce a specific scenario. 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 PHY modules provided in SUNSET. When the first two models are selected, the SNIR value is considered to compute the BER and the corresponding PER (depending on packet size) related to the selected modulation scheme. When the Static model is chosen, the user can set the desired average packet error rate in the network or per link. Each time a packet is received by a node a random value is generated and compared with the error probability selected by the user to decide if the packet has to be dropped or not.

**Channel emulator**

The channel emulator is a powerful tool provided by SUNSET that allows us to emulate an underwater network made up of several devices without the need of real acoustic modems (Figure 3). Several node instances can run on the same PC and on different platforms (PCs and embedded devices) connected via Ethernet to create an in lab testbed. The channel emulator computes the propagation delay associated to each link according to the position assigned to the different nodes

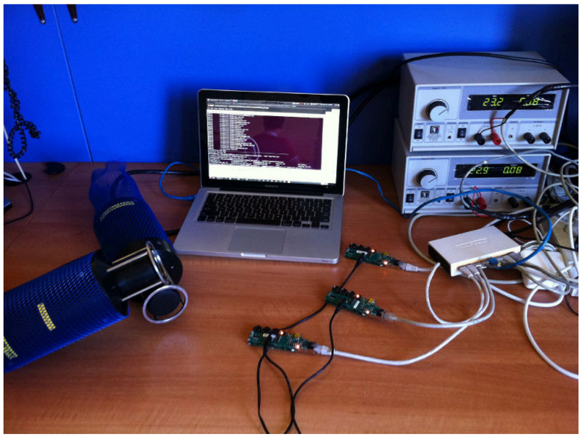


Figure 3: Laboratory testbed using the channel emulator and SUNSET running on different hardware platforms.

in 3D space. Using the channel emulator it is also possible to emulate node mobility updating the node position over time or when changes occur. Everything is done transparently to the user/developer. Moreover, the channel emulator can be used together with the packet error model and blacklist capabilities described above to design the desired network topology and link quality. It therefore allows to investigate, using the lab testbed, scenarios close to the ones expected during in field tests for an accurate tuning and fixing of the protocol solutions before the actual deployment of the nodes at sea.

**Energy consumption model**

A new energy consumption model has been developed in SUNSET that provides a more accurate system to compute the energy consumed by each node while transmitting, receiving or being idle. The proposed model supports also the possibility to have nodes using different transmission power levels when transmitting acoustic packets in the underwater channel. Moreover, it allows to load and use the real figures of a specific devices, e.g., the power consumption of a real acoustic modem based on measurements or on its data sheet.

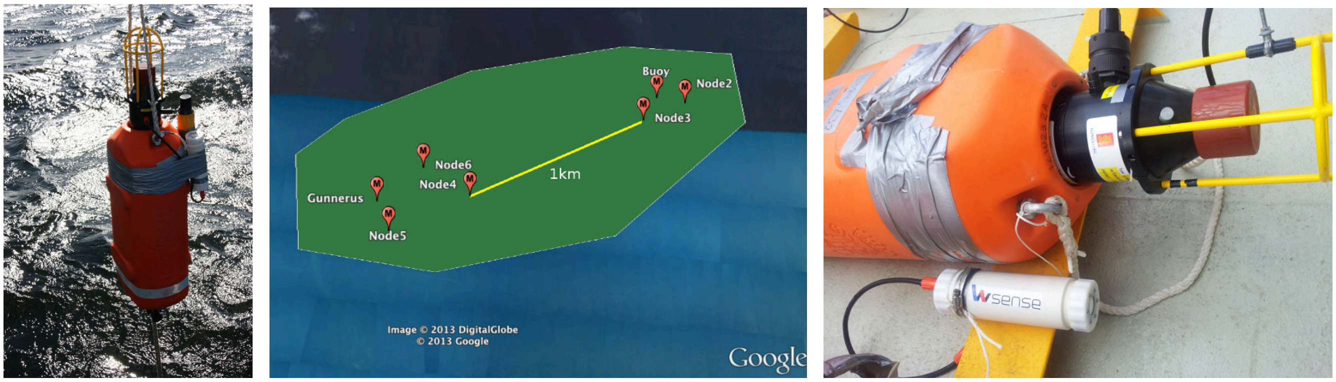
# 4. SUNSET EVALUATION AND VALIDATION

SUNSET has been extensively validated and evaluated during in field testing activities. Several tests have been conducted to verify the feasibility of the proposed emulation framework and its capability to work with different external devices. In what follows we describe the latest in field experiments we have conducted where the novel features introduced by SUNSET version 2.0 have been fully exploited. These experiments were part of the final demonstration tests for the EU FP7 CLAM project [19] conducted in May 2013 in the Trondheim fjord, Norway.[2] Two entire weeks of experiments have been conducted. The first week was mostly devoted to hardware integration, networking solution tuning, in water tests close to shore to prepare the actual experiments that would be conducted in the fjord during the second week. The CLAM cNode modem (Figure 4a) has been used as communication device. It is an innovative underwater node developed within the CLAM project that extends the functionalities provided by the Kongsberg cNode acoustic modem. During the second week seven CLAM cNode modems have been deployed (see Figure 4b): Five have been positioned at a depth of about 200 meters; one connected to a surface station (gateway buoy) acting as a bridge between the control station on shore and the underwater network; one deployed on the side of a ship (Gunnerus) acting as a mobile node. All nodes were battery powered and the ones deployed on the sea floor were equipped with acoustic releases for node recovery. Each of these nodes was also equipped with a floating collar to bring the nodes to the surface after the activation of the release. Inside each node a Gumstix Overo embedded device was installed and it was used to run the SUNSET framework and to interact with the acoustic modem. SUNSET version 2.0 has been used during all these activities: Preparation phase and final tests. The performance of all the networking solutions used within the CLAM project have been investigated: Two MAC protocols (Underwater Polling [20] and CSMA [21]) and three routing solutions (SUN [22], CARP [23] and Flooding). Ranging estimations among the nodes in the network have been collected to then investigate localization approaches in post processing.

## 4.1 Preparation phase

During the preparation phase, the SUNSET channel emulator has been used to test the performance of the implemented protocol solutions and to tune the protocol parameters (timing, delays, etc.) emulating a network like the one considered during the second week. Seven nodes have been emulated using the same positions of the 3D final deployment, introducing propagation delays similar to those experienced in the actual fjord tests. The estimated delays introduced by the real hardware (Gumstix and modems) have also been considered during this phase according to the computational delay and transmission bit rate of the selected platforms. To model different possible configurations different packet error rates have been tested, exploiting blacklisting to evaluate the impact of adding and removing direct links and to evaluate performance under different topology scenarios. The different SUNSET instances were run on PCs and on the same Gumstix boards eventually used in the CLAM nodes. A large set of tests has been investigated in a completely controlled environment considering different traffic loads, data sources and network configurations. Heavy traffic loads have also been investigated to test the system performance under duress in order to identify possible protocol misbehaving. These tests have been particularly important since they have allowed us to address and discover possible issues related to the use of the Gumstix Overo boards when running the software, loading the libraries and performing computations. Additionally, they have allowed us to better tune protocol parameters (timeouts and expected delays) when running on real hardware. The novel SUNSET modules have therefore allowed to speed up this preparation phase and identify and fix possible issues before the actual in field deployment, thus saving time, effort, and money.

[2] SUNSET has been the tool of choice used during the entire project to investigate the performance of different networking solutions during in field experiments.




(a) CLAM cNode modem during the deployment phase.

(b) Trondheim fjord network deployment.

(c) CLAM cNode equipped with the WSENSE CO2 sensor.

Figure 4: CLAM project experiments.

## 4.2 Final tests

The very code used and tested during the preparation phase has been used for the final experiments conducted in the second week. The performance of the different MAC and routing solutions have been investigated under different traffic loads and network configurations. The gateway buoy was used as sink node collecting all the information generated by the data sources on the seafloor. A CO2 Probe [24], developed by WSENSE [25], in collaboration with he Geochemistry group at the University of Rome "La Sapienza," was used as sensing device and data generator. This probe is an environmental underwater sensor for measuring temperature and $CO_2$ concentration. Using the communication and networking functionalities provided by SUNSET, the sensing device was remotely controlled and instructed by the control station (running on the ship) to collect and report the requested measurements to the user periodically or in case of specific events (e.g., if an alarm is detected). During the final tests, the blacklist functionalities were used again. Upon nodes deployment we were able to verify that all the data sources were able to communicate using a single link to the gateway buoy. Multi-hop communication has then been forced via software to investigate the performance of the different routing solutions. To control the operation of the nodes and to reconfigure the different devices in the network when switching from one test to the next one (changing the protocol stack, traffic load, etc.) the SUNSET back-seat driver has been used [8]. It has allowed to remotely operate and control the underwater devices in real-time using acoustic links for the entire week of experiments, avoiding the need of retrieving the underwater nodes for protocol stack reconfiguration or parameter tuning after their deployment. Using this system, it has been possible to perform several overnight tests considering different protocol stacks and network configurations. More than 50.000 packets have been transmitted during the trial. The node on the gateway buoy and the node on the ship have been used as control points to instruct the other nodes in the network. The back-seat driver has also been used to start external processes: Remotely activating the acoustic release and additional sensor measurements acquisition.

# 5. CONCLUSIONS

We have presented SUNSET version 2.0, describing its new features and capabilities with respect to those of the previous version. The new version of SUNSET extends and enhances the flexibility and reliability of SUNSET 1.0 when simulating, emulating and testing in field underwater protocols. New modules have been implemented to support all the different steps of protocol design and validation: Simulation, emulation and real-life testing. A more detailed documentation and a larger set of examples and configuration scripts are provided to help and support researchers and developers using SUNSET for their research. SUNSET features, old and new ones, significantly speed up the design and test of novel protocol stacks and allow to transparently test the developed solutions in a variety of different scenarios and settings, using different platforms. SUNSET also enables researchers and developers to fast identify underperforming or malfunctioning communication modules, before and during in field tests. This saves a lot of time and resources during experimental campaigns, significantly increasing the results that can be obtained in field.

## Acknowledgments

This work was supported in part by the EU FP 7 IP project SUNRISE "Sensing, monitoring and actuating on the UNderwater world through a federated Research InfraStructure Extending the Future Internet" and by the EU FP 7 STREP project CLAM "CoLlAborative EMbedded Networks for Submarine Surveillance." We thank Maksym Komar for the help provided for creating the patch to support the use of ns-2.35 with ns2-Miracle.

# 6. REFERENCES

[1] C. Petrioli, R. Petroccia, J. Shusta, and L. Freitag, "From underwater simulation to at-sea testing using the ns-2 network simulator," in *Proceedings of IEEE/OES OCEANS 2011*, Santander, Spain, June, 6–9 2011.

[2] C. Petrioli and R. Petroccia, "SUNSET: Simulation, emulation and real-life testing of underwater wireless sensor networks," in *Proceedings of IEEE UComms 2012*, Sestri Levante, Italy, September, 12–14 2012.

[3] SENSES Lab, "SUNSET: Sapienza University Networking framework for underwater Simulation, Emulation and real-life Testing," Last time accessed: October 2013. [Online]. Available: http://reti.dsi.uniroma1.it/UWSN_Group/index.php?page=sunset

[4] J. Heidemann, M. Stojanovic, and M. Zorzi, "Underwater sensor networks: Applications, advances, and challenges," *Royal Society*, vol. 370, no. 1958, pp. 158–175, May 2012.

[5] C. Petrioli, R. Petroccia, and J. Potter, "Performance evaluation of underwater mac protocols: From simulation to at-sea testing," in *Proceedings of IEEE/OES OCEANS 2011*, Santander, Spain, June, 6–9 2011.

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

[7] N. Baldo, F. Maguolo, M. Miozzo, M. Rossi, and 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, October 23–25 2007, pp. 1–8.

[8] R. Petroccia and 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, June, 10–13 2013.

[9] Z. Peng, J.-H. Cui, B. Wang, K. Ball, and L. Freitag, "An underwater network testbed: Design, implementation and measurement," in *Proceedings of the third ACM International Workshop on UnderWater Networks (WUWNet '07)*, Montréal, Quebec, Canada, September 14 2007.

[10] Z. Peng, Z. Zhou, J.-H. Cui, and A. Shi, "Aqua-Net: An underwater sensor network architecture: Design,implementation, and initial testing," in *Proceedings of MTS/IEEE OCEANS 2009*, Biloxi, Mississippi, USA, October, 26–29 2009.

[11] S. Shahabudeen, M. Chitre, M. Motani, and A. L. Y. Siah, "Unified simulation and implementation software framework for underwater MAC protocol development," in *Proceedings of MTS/IEEE OCEANS 2009*, Biloxi, Mississippi, USA, October, October, 26–29 2009.

[12] M. Chitre, I. Topor, and T.-B. Koay, "The UNET-2 modem - An extensible tool for underwater networking research," in *Proceedings of MTS/IEEE OCEANS 2012*, Yeosu, Korea, May, 21–24 2012.

[13] "SUNRISE: Sensing, monitoring and actuating on the UNderwater world through a federated Research InfraStructure Extending the Future Internet," Last time accessed: July 2013. [Online]. Available: http://sunrise.di.uniroma1.it

[14] "Gumstix inc." Last time accessed: March 2013. [Online]. Available: http://www.gumstix.com

[15] D. Torres, J. Friedman, T. Schmid, and M. B. Srivastava, "Software-defined underwater acoustic networking platform," in *Proceedings of the Fourth ACM International Workshop on UnderWater Networks (WUWNet '09)*, Berkeley, California, USA, November 3 2009, pp. 7:1–7:8.

[16] R. Masiero, S. Azad, F. Favaro, M. Petrani, G. Toso, F. Guerra, P. Casari, and M. Zorzi, "DESERT underwater: an NS-miracle-based framework to DEsign, simulate, emulate and realize test-beds for underwater network protocols," in *Proceedings of MTS/IEEE OCEANS 2012*, Yeosu, Korea, May, 21–24 2012.

[17] R. Petroccia and 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, June, 10–13 2013.

[18] Evologics, "Evologics S2C acoustic modems," Last time accessed: March 2013. [Online]. Available: http://www.evologics.de/

[19] "CLAM: Collaborative embedded networks for submarine surveillance," Last time accessed: March 2013. [Online]. Available: http://clam.ewi.utwente.nl/

[20] F. Favaro, F. Guerra, and M. Zorzi, "Data upload from a static underwater network to an AUV: Polling or random access?" in *Proceedings of MTS/IEEE OCEANS 2012*, Yeosu, Korea, May, 21–24 2012.

[21] S. Basagni, C. Petrioli, R. Petroccia, and M. Stojanovic, "Optimized packet size selection in underwater WSN communications," *IEEE Journal of Oceanic Engineering*, vol. 37, no. 3, pp. 321–337, July 2012.

[22] G. Toso, R. Masiero, P. Casari, O. Kebkal, M. Komar, and M. Zorzi, "Field experiments for dynamic source routing: S2C EvoLogics modems run the SUN protocol using the DESERT Underwater libraries," in *Proc. of MTS/IEEE OCEANS*, Hampton Roads, VA, Oct. 2012.

[23] S. Basagni, C. Petrioli, R. Petroccia, and D. Spaccini, "Channel-aware routing for underwater wireless networks," in *Proceedings of MTS/IEEE OCEANS 2012*, Yeosu, Korea, May, 21–24 2012.

[24] A. Annunziatellis, S. Graziani, S. Lombardi, C. Petrioli, and Petroccia, "CO2Net: A marine monitoring system for $CO_2$ leakage detection," in *Proceedings of MTS/IEEE OCEANS 2012*, Yeosu, Korea, May, 21–24 2012.

[25] "WSENSE s.r.l." Last time accessed: October 2013. [Online]. Available: http://www.wsense.it/