

Cooperation and Networking in an Underwater Network composed by Heterogeneous Assets

José Braga[†], Ricardo Martins[†], Chiara Petrioli^{*§}, Roberto Petrocchia^{‡¶} and Luigi Picari^{*§}

^{*} Computer Science Department, University of Rome “La Sapienza”, Rome, Italy

[†] Underwater Systems and Technology Laboratory Faculty of Engineering, Porto University, Portugal

[‡] NATO STO CMRE - Centre for Maritime Research and Experimentation, La Spezia, Italy

[§] WSENSE s.r.l., Rome, Italy

[¶] The work has been performed while the author was at University of Rome “La Sapienza” and WSENSE s.r.l.

Abstract—Underwater communication and robot technologies have grown rapidly in the last decade. Systems made of underwater unmanned vehicles have moved from single vehicle deployments to systems comprising teams of assets. As of today the possibility to support cooperation and interoperability of heterogeneous platforms is a key issue. The objective of this paper is to present the SUNRISE [1] approach to overcome this limitation through the development of an abstraction layer to support the interoperability of different vehicle control software. SUNRISE also aims at integrating control software with the underwater communication and networking components to create more flexible, modular and capable underwater systems. The developed SUNRISE infrastructure has been evaluated and validated through in lab tests and at-sea experiments, in the Porto harbour, during the REP14 sea trial in the Atlantic Ocean and in the Mediterranean coast of Italy in June 2015, showing promising results. The authors gratefully acknowledge CMRE team, Sapienza team and the Portuguese Navy for their valuable feedback and for their support and help during the REP14 sea trial.

Index Terms—Underwater Wireless Sensor Networks, Autonomous Underwater Vehicle, Remote Control, DUNE, ROS, MOOS, SUNSET, S-SDCS, SUNRISE.

I. INTRODUCTION

The interest in Underwater Wireless Networks (UWNs) has largely increased in the past decade to support a wide range of emerging applications, including monitoring of the environment and critical infrastructure, coastline protection, and prediction of underwater seismic and volcanic events [2], [3]. To support these applications the technology of Unmanned Maritime Vehicles (UMVs) has evolved significantly in the past five years. Autonomous Underwater Vehicles (AUVs), Unmanned Surface Vehicles (USVs), Remotely Operated Vehicles (ROVs), gliders, buoys, vessels, and fixed subsea or surface equipment are increasingly working together [4], [5]. New types of UMVs have been developed by both Research Institutions and Industry with increasing capabilities. As stated in [4], however, the status of technol-

ogy and user maturity needs to be differentiated between what is being done in research and what is being employed operationally. For instance, cooperative control of many UMVs has been researched and experimented for more than a decade. However, UMVs working together autonomously in an operational/commercial setting is yet to be realized.

One of the main reasons for this reduced speed in the development and deployment of UWNs is the absence of standards and common interfaces for underwater digital communication and information sharing among heterogeneous network nodes. A first initiative to define a common language to support initial contact and emergency message exchange between nodes has been initiated by the NATO STO Centre for Maritime Research and Experimentation (CMRE) together with Academia and Industry. The proposed physical coding scheme, named JANUS [6], [7], [8], [9], is currently in the process of becoming a NATO standard. However, even if the heterogeneous mobile nodes in the network support a common physical coding scheme, they need to encode and decode messages in the same way. Without this level of understanding any interaction between two underwater robots, using different control software, would therefore not be possible.

In this paper we present the SUNRISE project [1] activities to overcome this limitation and to create the possibility for a heterogeneous network of mobile assets that are able to communicate and cooperate. In particular, in SUNRISE an abstraction layer has been defined to support the interaction between the communication and networking components and the control software of the UMV. This interaction protocol, named SSC for Software-to-Software Communication, can be used by any networking and control software and an XML document is used to describe the structure and semantics of commands. The current version of SSC supports a core set of commands, described in detail in Section III,

and it will be extended and improved according to the user and operator needs in the next months. The SSC protocol has been fully integrated with the DUNE [10], MOOS [11], ROS [12] and SUNSET [13] software defined communication stack (S-SDCS in the following) which have been selected as the standard SUNRISE solutions to control the UMV operations and to provide the networking functionalities, respectively. Several in lab and at sea experiments have been conducted to evaluate and validate the SSC protocol and its integration with DUNE and SUNSET. Regarding MOOS and ROS, right now only in lab experiments have been conducted. Sea trial were carried out as well as in the Porto harbour also in the Atlantic Ocean in July 2014 and in the Mediterranean sea in June 2015. A network consisting of 8 heterogeneous nodes was deployed in the Atlantic Ocean during the REP14 sea trial. REP14 was conducted in collaboration with CMRE and the Portuguese Navy under a collaboration agreement with the University of Porto and the University of Rome “La Sapienza”. The developed infrastructure was used to provide to the system operator the capability to interact with and to instruct remotely and in real time the submerged vehicles on the tasks to perform. The developed solution was also used in the Mediterranean sea, off the coast of Marzamemi (Sicily, south of Italy), during an archeological survey mission part of a collaboration between University of Rome “La Sapienza”, University of Porto and the Sicily Region Authority for the Sea. In June 2015 we deployed at sea in Marzamemi the SUNRISE *re-deployable testing facility*. The SUNRISE re-deployable testing facility is a cable-less testbed developed by the University of Rome “La Sapienza” composed by multiple underwater sensor nodes that can be easily deployed and recovered by SUNRISE users. The SUNRISE re-deployable testing facility has been designed to be dynamic, easy to deploy and use, and highly adaptable to different application scenarios. Each node of the testbed can be easily customized with additional hardware (e.g., sensor(s), battery pack(s), modem(s), external storage drives) based to the user’s needs. Nodes of the SUNRISE re-deployable testing facility run La Sapienza S-SDCS creating a network among each other and with possible multivendor vehicles integrated in the system. In Marzamemi in addition to 4 underwater sensor nodes the re-deployable testing facility included also three Light AUVs which were also running the S-SDCS. In sea trials all the remote commands were correctly delivered to the vehicles using acoustic communications and networking capabilities provided by the S-SDCS, thus resulting in a complete success.

The rest of the paper is organized as follows. Previous work on proposed solutions for underwater vehicle control, cooperation and networking is summarized in

Section II. In Section III we describe the SSC protocol in detail. Section IV illustrates the integration of SSC with DUNE and SUNSET software defined communication stack (S-SDCS). Experimental activities are described in Section V. Finally, Section VI concludes the paper.

II. RELATED WORK

Many papers have been published in the past presenting solutions for the navigation and cooperation of mobile assets and networking approaches. As of today, all these different components have been mainly addressed separately or using ad-hoc solutions [14], [15], [16], [17], [18]. Different middleware platforms for robotics research have been developed, extended, and largely used by the robotic research community. The current de facto standard software suites for the control of unmanned underwater vehicles include the Mission Oriented Operating Suite (MOOS) [11], the Robot Operating System (ROS) [12] and DUNE (DUNE: Uniform Navigation Environment) [10]. These control software suites are in general responsible for collecting information from vehicles on-board sensors, processing the data according to the selected algorithms (navigation, vehicle cooperation, etc.), and determining the next course of action. For instance, if an algorithm aims at enabling cooperation among different vehicles, such as moving them in a predefined formation, the information about the other vehicles has to be collected and the vehicle’s position has to be adjusted accordingly, to keep the formation. Similarly to what described for vehicle control suites, different software to support networking capabilities when deploying assets underwater have been designed and implemented. The most used ones are currently SUNSET [13], DESERT [19], UnetStack [20], SeaLinx [21]. These software solutions provide open or proprietary architecture to implement various networking protocols supporting the delivery of messages in the network over multiple hops. In particular, in SUNRISE we have adopted and extended SUNSET due to its flexibility, performance at sea and support for adaptive networking. Different approaches have been investigated at the different layers of the protocol stack, including random access, time division, cross-layering, etc. These solutions have been also used and tested via simulations, in lab testing and in-field experiments. In SUNRISE SUNSET has been extended into SUNSET software defined communication stack (S-SDCS) [22]d which now supports multi-modal communications and multi-stack operation: the network dynamically selects the best communication technology to use, the best protocol and protocol parameters depending on decisions of policy engines e.g., to account for varying channel conditions or application quality of service constraints. S-SDCS has been used for enabling communication among the

underwater assets used in the experiments reported in this paper. In [16], [18] an ad-hoc solution is investigated by the authors to combine navigation, communication and networking capabilities. SUNSET has been used for networking and communication support and the proprietary INESC/TEC software to control the surface and underwater vehicle operations. According to the knowledge of the authors, however, no approach has been proposed so far to combine networking, communication and navigation capabilities in a general, flexible and portable way that can be used by the various control, communication and networking software. In the rest of the paper we describe the SUNRISE approach which aims at designing a general communication protocol to interconnect middleware platforms to control robots with networking and communication solutions. This protocol, named SSC, has been defined and it has been integrated with the DUNE and the SUNSET S-SDCS. Various in lab and in field tests have been conducted to evaluate the flexibility and portability of the proposed approach showing promising results. Recently, University of Rome La Sapienza team has also extended the work to interconnect the S-SDCS with multi-vendor vehicles running MOOS and ROS.

III. SSC PROTOCOL

When considering a network made of different and heterogeneous mobile assets, various control software (MOOS, ROS, DUNE, etc.) may be used within the same network. In order to make possible a cooperation between these heterogenous vehicles, they should all be able to understand each others instructions. An abstraction layer has to be designed to enable the interaction and interoperability of different systems. SUNRISE project [1] aims at building such a networking system where heterogeneous underwater nodes, both static and mobile, are able to communicate and cooperate ¹. Additionally, to reach a high level of efficiency and performance in the deployed network, SUNRISE aims at combining and integrating communication, networking, sensing and navigation capabilities. This combination and integration would make possible to create more capable and smart devices, which are able to react to external stimulus and adapt to dynamic environments.

To achieve this task the SSC protocol has been designed to create a general abstraction layer to interconnect with any networking and control software.

The main objectives in designing the SSC protocol have been:

- Simplicity

¹We assume that the considered nodes are able to decode the received acoustic messages, i.e. they use the same physical coding schemes and protocol stack solutions.

- Ease of implementation
- Extendibility
- Expressiveness

Platform independence has been another important aspect guiding the design choices. The created communication protocol should in fact make little or no assumptions about implementation details of the parties involved in exchanging commands. An XML document has been defined describing the structure and semantics of the commands.

The current version of the SSC protocol defines and supports a core set of commands to: 1) Collect information on the capabilities of the UMV; 2) Collect status information on the vehicle; 3) Define and delete missions; 4) Start and stop missions; 5) Abort the current mission (it is executed in case of emergency); 6) Support the interaction of the vehicle with the communication and networking component. Additionally, SSC enables to set and get parameters on the modem or the protocol stack and to estimate the range to the other nodes in the network. Each command can be transmitted by the vehicle or the system operator and it can address the local mobile node or any remote node (with the support for multicast or broadcast) making use of the networking support. Commands are defined as a comma separated list of strings with the following basic structure:

- **Header:**
 - *Version*: The version of the command, at the present stage all commands have version 0.
 - *Flags*: Bit field with behaviour modifiers.
 - *Time-to-live*: The number of seconds after which the command will expire if not successfully acted upon.
 - *Priority*: The priority of the command. The higher the number the higher the priority.
 - *Source*: Source node identifier.
 - *Destinations*: List of identifiers of destination nodes.
 - *Name*: the name of the command.
- **Arguments:** An arbitrarily long list of fields specific to a given command name.
- **Footer:**
 - *CRC*: The cyclic redundancy check of all preceding octets.
 - *Command terminator*: The commands terminator octet.

The list of the commands supported by the SSC protocol version 0 is defined in Table I.

The SSC protocol enables the use of the networking features by the vehicle control software (data exchange,

²This message allows the networking system to inform the vehicle about the beginning and the end of an acoustic transmission which may interfere with measurement collected by the on board acoustic sensors (echo sounder, multi beam, side scan sonar, etc.).

Table I: SSC commands.

MANOEUVRES	
Goto	It encodes a manoeuvre specifying a desired movement of a system to a target waypoint.
StationKeeping	It makes the vehicle enter a given circular perimeter around a waypoint coordinates for a certain amount of time.
Rows	It instructs the vehicle about performing a lawn-mower type manoeuvre at a given depth/altitude with a given speed.
Dock	It instructs the vehicle to dock (undock) to (from) a predefined docking station.
PLAN MANAGEMENT	
PlanListGet	It instructs the destination system(s) to return the list of plans that can be executed on the target system(s).
PlanGet	It instructs the destination system(s) to return the list of manoeuvres for the requested plan.
PlanStart	It instructs the destination system(s) to start the requested plan.
PlanStop	It instructs the destination system(s) to stop the plan currently running on the vehicle.
PlanAdd	It instructs the destination system(s) to add the requested plan, expressed as a list of manoeuvres.
PlanDelete	It instructs the destination system(s) to delete the requested plan.
Abort	It instructs the destination system(s) to cancel any type of automatic control and to enter the standby mode.
SENSOR MANAGEMENT	
SensorListGet	It instructs the destination system(s) to return the list of measurements available at the target system(s).
SensorInfoGet	It instructs the destination system(s) to return the list of sensors supporting the requested measurement.
SensorSampleGet	It instructs the destination system(s) to return measurements from the given target sensor(s).
POSITION, RANGING, AND BEARING	
PositionGet	It instructs the destination system(s) to return its(their) position in the world.
RangeGet	It instructs the destination system(s) to return the estimated travel time (and bearing, if available) to the given target system(s).
LOW-LEVEL COMMANDS	
PacketTx	It requests transmission of a packet with user-defined data.
PacketTxStatus ²	It conveys the status of a transmission request.
PacketRx	It is dispatched when a data packet is received.
ParamGet	It instructs the destination system(s) to return the value for the given setting (parameter).
ParamSet	It instructs the destination system(s) to set the target parameter with the given value.

parameter setting, information collection, etc.) and the collection of data available at the vehicle that can be used by the protocol stack solutions. The overall infrastructure makes it possible to control vehicle operations remotely and in real time despite the specific control software running on each vehicle. Commands to the vehicle control software (generated by another vehicle or a remote operator) can be encoded in the SSC format and passed to the networking component. The networking component translates the SSC command (depending on the considered networking software) in order to compress and encode it before the in-water acoustic transmission. The needed protocol stack data, if any, are also added in order to deliver the packet to the intended destination. On the receiver side the opposite steps are performed. The SSC command is extracted from the received acoustic message and it is passed to the control software. The on-board software then translates the SSC command, according to the specific language and format used by the on-board control software, in order to execute the requested action. Specific drivers are therefore needed for the conversion between the control software and the SSC format and between the networking component and the SSC protocol. These conversions make possible the use and interoperability of any control and networking software using the SSC protocol. They can be easily implemented by the developers of the control and networking software which know the details of the specific systems.

IV. SSC PROTOCOL INTEGRATION

The SSC protocol has been fully integrated with the DUNE [10] and SUNSET [13] software defined communications stack which have been selected as the standard SUNRISE solutions to control the UMV operations and to provide the networking functionalities, respectively. In what follows we briefly describe the DUNE and SUNSET S-SDCS systems and their capabilities to run in emulation and real-life mode, which has been quite useful to test and debug the implemented system before going at sea.

A. DUNE

DUNE [10] is the on-board software used in the autonomous vehicles developed by the LSTS group at the University of Porto. DUNE provides an operating-system and architecture independent C++ programming environment for writing efficient and modular real time reactive tasks. DUNE uses the publish/subscribe pattern to provide loose coupling between modules, which in DUNE are called tasks. DUNE tasks publish and subscribe messages without knowing any details about the other tasks. Effectively, message passing is the only mechanism available to exchange information between

tasks. For example, a task that interacts with a sensor produces a message of a given type with a sensor reading (e.g., acceleration), that can later on be consumed by another task which integrates that information and produces a state estimate (e.g., estimated position), another task can consume that estimate and produce commands to actuators (e.g., increase thrust). Figure 1 illustrates this concept.

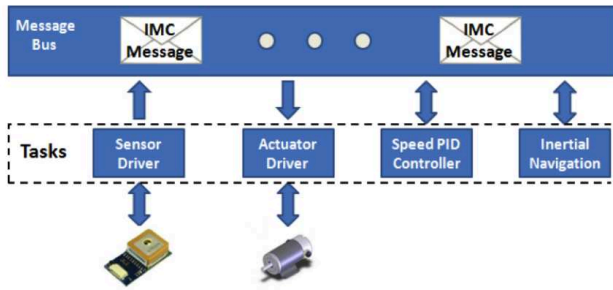


Figure 1: Simplified representation of DUNE's message passing concept.

This modularity and loose coupling between tasks facilitates incorporating new sensors and functionalities without having to change the existing code. The new functionality is implemented by writing a new task that only needs to listen to the necessary inputs and dispatch the outputs. In most cases a simple configuration change is enough to support new functionality. This fact eases not only the life of the everyday developer, but also of new, or temporary developers that will only be working with some specific module of the software, being shielded from the complexity of the remaining tasks of the framework. Communication between tasks is made exclusively using IMC message [23]. IMC is a message-oriented protocol for autonomous vehicles and sensor networks. IMC defines a set of common messages used for communication between the different vehicle components. Additionally, DUNE provides a simulation engine. In this case, all tasks that interact with real sensors and actuators are replaced by analogous simulation tasks (simulation engines for sensors and actuators are based on historical data collected during in-field operations). Hardware-in-the-loop simulation can be achieved using the same method. A driver has been added to DUNE to convert IMC messages into SSC commands and vice versa when providing instructions to and receiving messages from the networking component, respectively.

B. SUNSET

SUNSET, for Sapienza University Networking framework for underwater Simulation Emulation and real-life Testing [13] is a software module for creating

networks of underwater sensor nodes and underwater vehicles. SUNSET has been created with the objective of providing a flexible, reliable, and efficient standard framework for running protocol stacks where the same implemented code can be used to run simulation, emulation and in-field experiments. No code rewriting is needed when porting the simulated/emulated solutions to work with real hardware, which allows to fast move from solution design to field testing. In SUNRISE SUNSET has been extended by University of Rome La Sapienza's group into SUNSET Software Defined Communications Stack (S-SDCS) [22], and by adding a framework for secure communication [24]. In the S-SDCS multiple protocol stacks (possibly for multiple communication technologies) are implemented. Based on application requirements, traffic, channel conditions and additional network information, policy engines dynamically select the communication technology, protocols and protocol parameters to use to optimize system performance.

The S-SDCS has been designed by separating the protocol stack from the additional components handling the communications with external devices. This provides the possibility to implement new protocol solutions in an easy and fast way, without affecting external hardware modules (see Figure 2) and makes very easy to interconnect the S-SDCS with different commercial probes and with different navigation software.

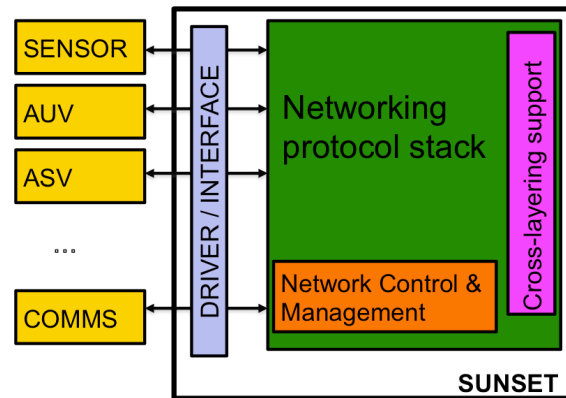


Figure 2: Simplified representation of SUNSET architecture.

Additionally, through the use of its network control and management module the S-SDCS allows to remotely operate the devices in 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 performed “on the fly”, without the need to retrieve the underwater devices. This makes it possible to easily run different experiments without interruptions, thus saving time and money during at-sea campaigns. As for DUNE, a specific

driver has been developed for the S-SDCS to support the use of the SSC protocol. When a message in the SSC format is provided to SUNSET S-SDCS, the intended destination of the message is considered. If the message is addressing the local node, the request (ranging request, parameter setting, etc.) is executed and the corresponding response is provided to the vehicle. If the intended destination of the message is instead a remote node or a set of remote nodes, the message is compressed and encoded to be transmitted in water in an efficient way, according to the considered protocol stack. When the remote node receives the acoustic request, the message is decoded, converted in the SSC format, and executed (on the S-SDCS side or DUNE side). Similarly, if a response has to be delivered, it is compressed, encoded and transmitted in water.

V. PERFORMANCE EVALUATION

Leveraging on the SUNSET S-SDCS and DUNE capabilities to run in simulation/emulation mode, the development and preliminary testing of the SSC protocol was first done using DUNE's and SUNSET's simulation engines. The software was then tested in the Porto harbour with three AUVs and one shore modem. The system operator was using the shore modem to access the underwater network via SUNSET S-SDCS and to instruct the submerged AUVs on the actions to perform via acoustic messages. The system operator was able to remotely start, stop and abort different mission plans loaded on the vehicles. Information about the positions and the sensor measurements collected by the nodes were provided in real time to the system operator. During the harbour tests, the integration of SUNSET S-SDCS with the SUNRISE web interface (SUNRISE GATE [25]) was also used to provide to the system operator access via Internet to the deployed network with a graphical user friendly interface (see Figure 3).

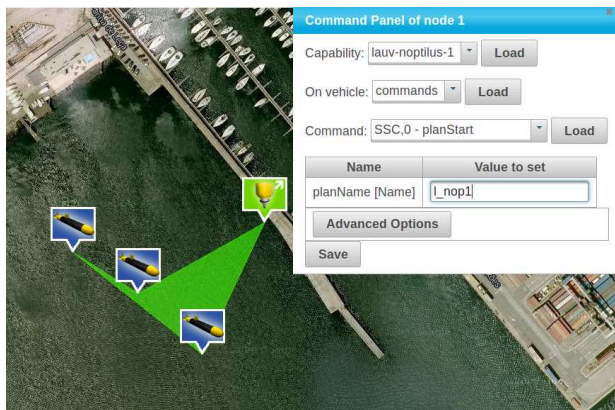


Figure 3: SUNRISE GATE Web GUI.

The first at sea tests were then performed during the REP14 sea trial, conducted in collaboration with CMRE and the Portuguese Navy. Figure 4 shows the network deployed for the REP14 tests consisting of 8 heterogeneous nodes, deployed in the Atlantic Ocean in July 2014: Two mobile surface vehicles - Lisa and Carol; 4 mobile underwater vehicles - LSTS' LAUV Noptilus 1 and LAUV Noptilus 2 AUVs [26], and two eFolaga vehicles [27]³; one moored gateway buoy - GW; and the NATO Research Vessel Alliance.

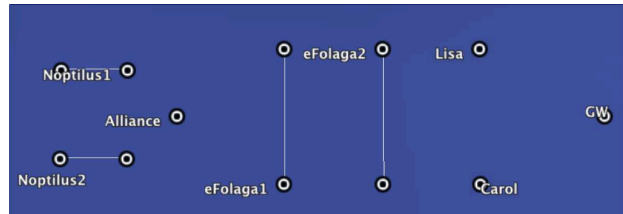


Figure 4: REP14 Network deployment.

DUNE was running on-board of two AUVs (LAUV Noptilus 1 and LAUV Noptilus 2) while the S-SDCS was installed in all the nodes. A system operator interacted remotely and in real time with the underwater network using the gateway buoy (GW) and multi-hop acoustic communications. LAUV Noptilus 1 and LAUV Noptilus 2 were the final destination of the commands and all the other nodes were used as relays to forward acoustic requests and responses in the network.

Using SUNSET S-SDCS and the control interface running on the gateway buoy, the system operator was able to remotely send commands to the LAUV Noptilus vehicles containing the actions to perform (plan start, stop and abort). During the various tests the same infrastructure was used to give the operator real time updates on the vehicles' positions and on the measurements collected by the on-board sensors (including Temperature, as shown in Figure 5). All the remote commands were correctly delivered enabling remote control on the submerged vehicle by the system operator. On average 2.5 hops were used to deliver the information from the system operator to the vehicle and vice versa, and a maximum of 5 hops was experienced. The SSC protocol was also used to provide the vehicle the possibility to control the transmission power of the acoustic modem. The vehicle was able to configure the use of a lower transmission power value when navigating at the surface⁴ and increasing this value when navigating underwater.

³The eFolaga vehicles were using MOOS as on board control software executing pre-planned missions.

⁴When navigating at the surface, it is possible to have the transducers temporary out of the water due to waves motion. Transmitting with a high power while the transducer is out of the water can damage it.

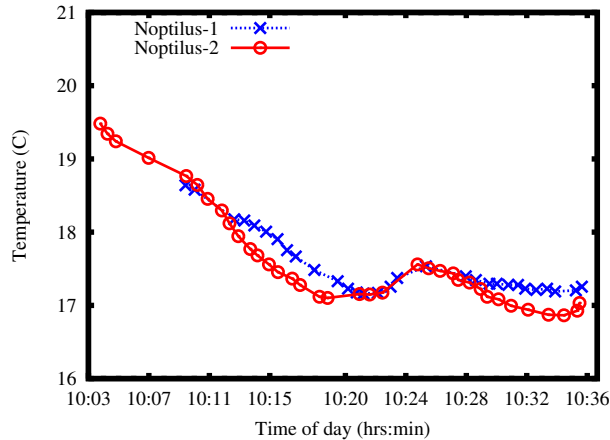


Figure 5: Subset of temperature measurements collected by Noptilus vehicles.

In REP14, the SSC RangeGet command was also used and tested. The DUNE software requested the S-SDCS to estimate ranges to the other nodes in a cooperative way, thus providing these estimations to the onboard LBL position update algorithm. The S-SDCS was using a two-way message exchange with multi-cast requests (single request and multiple replies) [28]. One vehicle (LAUV Noptilus 3 moving following a rectangular shape) and four beacons were used (Lisa, Carol, GW and Alliance) as displayed in Figure 6. The use of a cooperative approach made it possible to reduce the number of messages transmitted in water and the time needed to collect all the estimated ranges. Additionally, using a non cooperative approach, the vehicle would have needed to poll one by one the beacon nodes following a given order. If multiple consecutive beacons in that order are not reachable, the AUV will not be able to collect any update, even if other beacons could have replied and provided feedback to the LBL system. The use of a cooperative approach makes it possible to constantly collect range updates from all the beacons in the communication range. This can be clearly seen in Figure 6 where, while the vehicle is moving along the rectangular path, it is able to reach only a subset of the beacons (GW and Lisa, the other beacons are too far away) and it is not wasting time trying to reach nodes out of the communication range.

Using the implemented system the vehicle was following the rectangular plan with a good accuracy, the average error while following the track was 1.9m, maximum error was 6.4m.

Via the combination of the networking, communication and localization capabilities the SUNRISE approach also supports the use of mobile beacons. The two surface vehicles (Carol and Lisa) could have been moving following the AUV to be always able to provide position

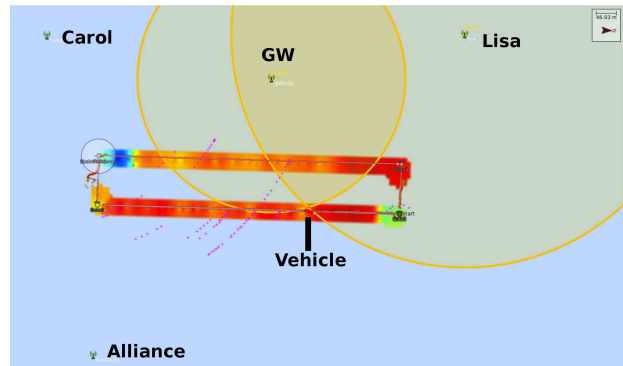


Figure 6: Vehicle ranging the deployed beacons using SUNSET cooperative approach.

updates. The position of the surface vehicle, obtained using the on board GPS, can be in fact embedded in the acoustic message to provide correct ranging estimation. In this way a more accurate position estimation can be achieved. Additionally, the use of a mobile infrastructure to support the AUV navigation would make it possible to significantly extend the operational area. The second at sea tests were during an archeological survey mission held in June 2015 in Marzamemi (Sicily, south of Italy). The aim of the mission, born from a partnership between University of Rome “La Sapienza”, University of Porto and the Sicily Region Authority for the Sea, was to perform a pilot exploration of relics from Roman times. Figure 7 shows the network topology used during the archeological survey mission. The network was composed by 8 heterogeneous nodes: 4 static underwater sensor nodes, 3 AUVs (LAUV Noptilus) and the control room (running the SUNRISE GATE) located on a Rigid-hulled inflatable boat (RIB) and connected to the underwater network through an acoustic modem dropped from the RIB at a depth of about 3m.

The underwater network has been deployed in a shallow water environment with a maximum depth of 7m over a squared area of about 200×200 m. Static nodes and the acoustic modem dropped from the RIB have been deployed at a maximum depth of 4m. AUVs sailed at different depths ranging from 1 to 5m. The underwater sensor nodes were measuring and transmitting geochemical parameters on the water quality while AUVs were used to perform a mission that consisted in creating an image of the sea floor. Figure 8 shows an image of the debris of three roman columns obtained from the side scan sonar.

We tested the capability to get information in real time from the sensors and AUVs on their measured data and position. We also tested the ability to control in real-time the underwater nodes and vehicles, adding, starting and stopping tasks and changing the mission

plan of the AUVs while they were underwater. All tests and command/information exchange were successful, proving proper behaviour of the SSC protocol as well as robustness of the networking module and excellent usability of SUNRISE re-deployable testing facility in real underwater applications.

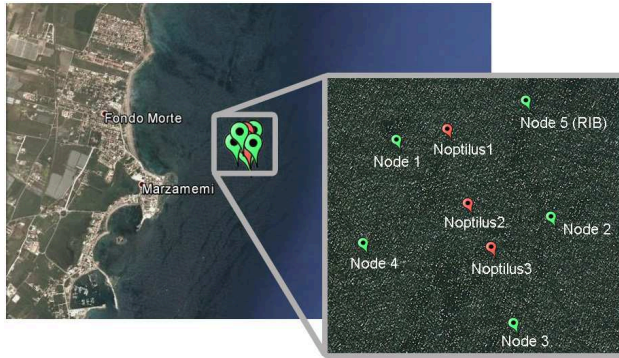


Figure 7: Network deployment used in Sicily (south of Italy).



Figure 8: Image of Roman columns obtained from the side scan sonar.

On going SUNRISE at-sea experiments are testing new functionalities implemented in the SSC protocol, as well as drivers implemented by La Sapienza team for MOOS and ROS in scenarios with cooperating heterogeneous mobile nodes operating different control software.

VI. CONCLUSIONS

In this paper we have presented the SUNRISE achievements on creating an underwater network supporting the communication and cooperation of heterogeneous static and mobile assets. More specifically, a flexible, general and modular protocol, named SSC, has been defined to support the interaction between any vehicle control software with the support for communication and networking components. The SSC protocol has been fully integrated with the two software selected in SUNRISE as the standard solutions to use, DUNE to control the UMV operations and S-SDCS to provide

the networking functionalities. The developed infrastructure has been largely tested via in lab tests and at sea experiments held in the Atlantic Ocean and in the Mediterranean sea showing promising results. Indeed, during the REP14 at sea trial, 8 heterogeneous nodes (static nodes, surface and underwater vehicles) were deployed in the Atlantic Ocean, while in the summer 2015, during an archeological survey mission held in Sicily (south of Italy), 8 heterogeneous nodes (static nodes and AUVs) were deployed in the Mediterranean sea. The system operator was able to interact with and instruct remotely and in real time the submerged vehicles on the tasks to perform. It was also able to collect feedbacks on the status of the vehicle and measurements from the on board sensors. New approaches to combine communication, networking and localization were also tested.

On-going tests in SUNRISE are evaluating extensions of the SSC protocol. MOOS and ROS drivers have also been implemented by La Sapienza team to support the use of the SSC protocol in a scenario comprising multivendor underwater assets. This will make it possible to have in the next SUNRISE sea trial vehicles running different control software able to exchange data and cooperate, e.g., LAUV Noptilus running DUNE, eFolaga running MOOS and MARTA [29] running ROS.

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 CMRE team, Sapienza team and the Portuguese Navy for their valuable feedback and for their support and help during the REP14 sea trial. The authors also gratefully acknowledge the “Soprintendenza del Mare della regione Sicilia” that is the Sicily Region Authority for the Sea.

REFERENCES

- [1] “SUNRISE: Sensing, monitoring and actuating on the Underwater world through a federated Research Infrastructure Extending the Future Internet,” <http://fp7-sunrise.eu>.
- [2] J. Heidemann, M. Stojanovic, and M. Zorzi, “Underwater sensor networks: Applications, advances and challenges,” *Philosophical Transactions of the Royal Society A*, vol. 370, pp. 158–175, August, 2 2012.
- [3] L. Lanbo, Z. Shengli, and C. Jun-Hong, “Prospects and problems of wireless communication for underwater sensor networks,” *Wireless Communications and Mobile Computing, Special Issue on Underwater Sensor Networks*, vol. 8, no. 8, pp. 977–994, August 2008.
- [4] A. Y. Martin, “Unmanned Maritime Vehicles: Technology Evolution and Implications,” *Journal of Maritime Technology Society*, vol. 47, no. 5, pp. 72–83, September/October 2013.
- [5] S. L. Wood and C. E. Mierzwa, “State of Technology in Autonomous Underwater Gliders,” *Journal of Maritime Technology Society*, vol. 47, no. 5, pp. 84–96, September/October 2013.

- [6] K. McCoy, "JANUS: From Primitive Signal to Orthodox Networks," in *Proceedings of the 3rd International Conference and Exhibition on Underwater Acoustic Measurements: Technologies & Results*, ser. UAM2009, Nafplion, Greece, June 21–26 2009.
- [7] K. McCoy, B. Tomasi, and G. Zappa, "JANUS: The genesis, propagation and use of an underwater standard," in *Proceedings of the European Conference on Underwater Acoustics*, ser. ECUA2010, Istanbul, Turkey, July, 5–9 2010.
- [8] J. R. Potter, J. Alves, D. Green, G. Zappa, I. Nissen, and K. McCoy, "The JANUS Underwater Communications Standard," in *Proceedings of IEEE UComms 2014*, Sestri Levante, Italy, September, 3–5 2014.
- [9] "JANUS wiki," Last time accessed: July 2016. [Online]. Available: <http://www.januswiki.org/tiki-index.php>
- [10] J. Pinto, P. Dias, R. Martins, J. Fortuna, E. Marques, and J. Sousa, "The LSTS toolchain for networked vehicle systems," in *Proceedings of MTS/IEEE OCEANS 2013*, Bergen, Norway, June, 10–13 2013.
- [11] "MOOS: Mission Oriented Operating Suite," Last time accessed: July 2016. [Online]. Available: <http://www.moos-ivp.org>
- [12] "ROS: Robot Operating System," Last time accessed: July 2016. [Online]. Available: <http://www.ros.org/>
- [13] C. Petrioli, R. Petrocchia, J. R. Potter, and D. Spaccini, "The SUNSET framework for simulation, emulation and at-sea testing of underwater wireless sensor networks," *Ad Hoc Networks*, vol. 34, pp. 224–238, 2015.
- [14] J. Das, T. Maughan, M. McCann, M. Godin, T. O'Reilly, M. Messie, F. Bahr, K. Gomes, F. Py, J. G. Bellingham, G. S. Sukhatme, and K. Rajan, "Towards mixed-initiative, multi-robot field experiments: Design, deployment, and lessons learned," in *International Conference on Intelligent Robots and Systems, IROS2011*, San Francisco, CA, USA, September, 25–30 2011, pp. 3132–3139.
- [15] J. Pinto, J. Sousa, F. Py, and K. Rajan, "Experiments with Deliberative Planning on Autonomous Underwater Vehicles," in *In Workshop on Robotics for Environmental Monitoring, IROS2012*, Algarve, Portugal, October, 7 2012.
- [16] N. A. Cruz, B. M. Ferreira, A. C. Matos, C. Petrioli, R. Petrocchia, and D. Spaccini, "Implementation of an underwater acoustic network using multiple heterogeneous vehicles," in *Proceedings of MTS/IEEE OCEANS 2012*, Hampton Roads, Virginia, October, 14–19 2012.
- [17] A. Caiti, V. Calabro, T. Fabbri, D. Fenucci, and A. Munafò, "Underwater communication and distributed localization of auv teams," in *OCEANS - Bergen, 2013 MTS/IEEE*, Bergen, Norway, June, 10–13 2013, pp. 1–8.
- [18] N. A. Cruz, B. M. Ferreira, O. Kebkal, A. C. Matos, C. Petrioli, R. Petrocchia, and D. Spaccini, "Investigation of Underwater Acoustic Networking Enabling the Cooperative Operation of Multiple Heterogeneous Vehicles," *Marine Technology Society Journal*, vol. 47, pp. 43–58, March/April 2013.
- [19] R. Masiero, S. Azad, F. Favaro, M. Petrani, G. Toso, F. Guerra, P. Casari, and M. Zorzi, "DESERT underwater: an NSmiracle-based framework to DEsign, simulate, emulate and realize testbeds for underwater network protocols," in *Proc. of IEEE OCEANS 2012*, Yeosu, Korea, May, 21–24 2012, pp. 1–10.
- [20] 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, pp. 1–7.
- [21] S. N. Le, Z. Peng, J.-H. Cui, H. Zhou, and J. Liao, "Sealinx: A multi-instance protocol stack architecture for underwater networking," in *Proceedings of ACM WUWNet 2013*, Kaohsiung, Taiwan, November, 11–13 2013, pp. 1–5.
- [22] V. Di Valerio, F. Lo Presti, C. Petrioli, L. Picari, and D. Spaccini, "A Self-Adaptive Protocol Stack for Underwater Wireless Sensor Networks," in *Proceedings of MTS/IEEE OCEANS 2016*, Shanghai, China, April, 10–13 2016, pp. 1–8.
- [23] R. Martins, P. S. Dias, E. R. Marques, J. Pinto, J. B. Sousa, and F. L. Pereira, "IMC: A communication protocol for networked vehicles and sensors," in *Proceedings of IEEE OCEANS 2009*, Bremen, Germany, May, 11–14 2009, pp. 1–6.
- [24] G. Ateniese, A. Caposelle, P. Gjanci, C. Petrioli, and D. Spaccini, "SecFUN: Security framework for underwater acoustic sensor networks," in *Proceedings of MTS/IEEE OCEANS 2015*, Genova, Italy, May, 18–21 2015, pp. 1–9.
- [25] C. Petrioli, R. Petrocchia, D. Spaccini, A. Vitaletti, T. Arzilli, D. Lamanna, A. Galizia, and E. Renzi, "The SUNRISE GATE: Accessing the SUNRISE federation of facilities to test solutions for the Internet of Underwater Things," in *Proceedings of UCOMMS 2014*, Sestri Levante, Italy, September, 3–5 2014.
- [26] L. Madureira, A. Sousa, J. Braga, P. Calado, P. Dias, R. Martins, J. Pinto, and J. Sousa, "The light autonomous underwater vehicle: Evolutions and networking," in *Proceedings of MTS/IEEE OCEANS 2013*, Bergen, Norway, June, 10–13 2013, pp. 1–6.
- [27] A. Caffaz, A. Caiti, V. Calabrò, G. Casalino, P. Guerrini, A. Mager, A. Munafò, J. R. Potter, H. Tay, and A. Turetta, "The Enhanced Folaga: A hybrid AUV with modular payloads," in *G. Roberts, R. Sutton (Eds.), Further Advances in Unmanned Marine Vehicles*, IET, London, 2012.
- [28] G. Cario, A. Casavola, V. Djapic, P. Gjanci, M. Lupia, C. Petrioli, and D. Spaccini, "Clock Synchronization and Ranging Estimation for Control and Cooperation of Multiple UUVs," in *Proceedings of MTS/IEEE OCEANS 2016*, Shanghai, China, April, 10–13 2016, pp. 1–9.
- [29] B. Allotta, S. Baines, F. Bartolini, F. Bellavia, C. Colombo, R. Conti, R. Costanzi, C. Dede, M. Fanfani, J. Gelli, H. T. Gndogdu, N. Monni, D. Moroni, M. Natalini, M. A. Pascali, F. Pazzaglia, L. Pugi, A. Ridolfi, M. Reggiannini, D. Roig, O. Salvetti, and E. I. Tekdemir, "Design of a modular Autonomous Underwater Vehicle for archaeological investigations," in *Proceedings of MTS/IEEE OCEANS 2015*, Genova, Italy, May, 18–21 2015, pp. 1–5.