Channel Replay-based Performance Evaluation of Protocols for Underwater Routing

Stefano Basagni,* Chiara Petrioli,†‡ Roberto Petroccia†‡ and Daniele Spaccini†
* Dept. of Electrical and Computer Engineering
Northeastern University, Boston, MA, U.S.A.
basagni@ece.neu.edu
† Dipartimento di Informatica
Università di Roma "La Sapienza," Rome, Italy
{petrioli, petroccia, spaccini}@di.uniroma1.it
‡ WSENSE s.r.l., Rome, Italy

***Abstract*—The paper presents a comparative performance evaluation of two routing protocols for underwater wireless sensor networks (UWSNs). The two protocols are the Channel-Aware Routing Protocol (CARP), exemplary of a cross layer approach to underwater routing, and a simple variation of common flooding, called EFlood, where performance is enhanced by introducing random re-transmission times. The scenarios we consider are obtained via simulations and from trials at sea performed under a collaboration agreement between the University of Roma "La Sapienza" and the NATO Science and Technology Organization Centre for Maritime Research and Experimentation (STO CMRE). Two sets of simulations results are shown where the physical layer is modeled by a ray-traced channel as well as by replaying real channel traces, under the same network configuration. Results are also reported from campaigns of experiments at sea. Comparing all results shows how channel replay mimics faithfully actual channel dynamics with respect to what is achievable through a simulated channel model, thus demonstrating the effectiveness of this technique for a fair and repeatable performance comparison of solutions for UWSNs.**

***Index Terms*—Underwater wireless sensor networks, channel replay, performance evaluation.**

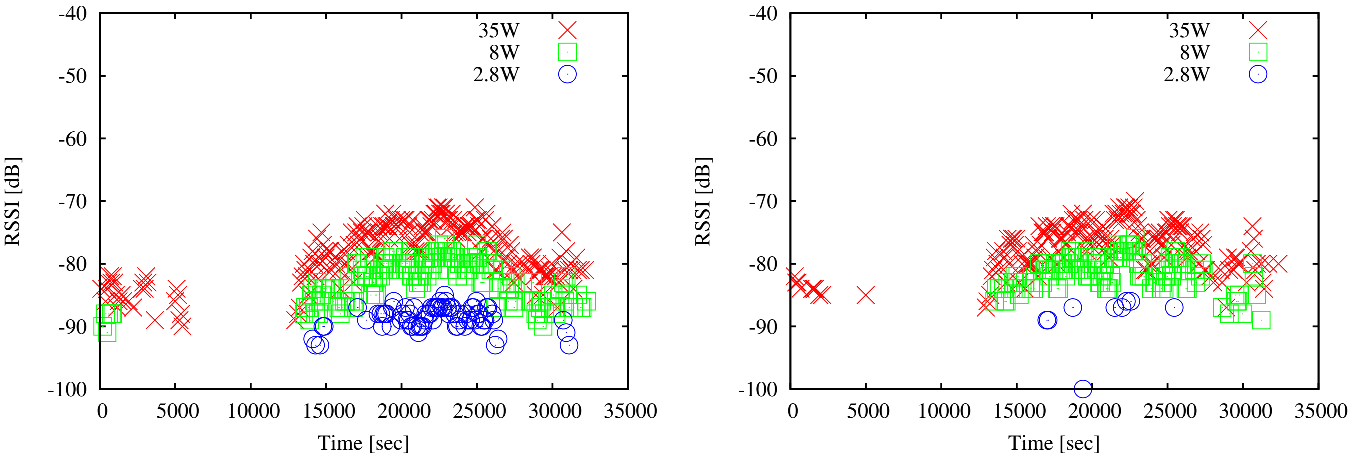
I. Introduction

Commercial and academic interest on Underwater Wireless Sensor Networks (UWSNs) has gained incredible momentum in recent years as they are seen as enablers of a host of applications including environmental monitoring, surveillance, port security, marine biology and discovery and protection of marine archaeology [1]. Several communication solutions have been proposed in literature for UWSNs where heterogeneous platforms, static or mobile, collaborate to accomplish the challenging tasks demanded by these diverse applications [2]. To this day, the functioning and performance of underwater protocols and systems have been mostly investigated through simulations. However, simulations can only capture little of the variability that is typical of the underwater environment, resulting in an oversimplified model of the acoustic channel and its dynamics. Furthermore, simulation models do not generally capture constraints introduced by actual hardware, which usually have a significant impact on protocol performance. Simulation platforms currently available for UWSNs do not accurately model the underwater acoustic channel, transmission loss, signal propagation and environmental noise. As a consequence, performance results can be significantly different from those obtained in field. Trials at sea are therefore the best form of experiments for accurate protocol performance evaluation. These kind of trials, however, incur high costs and logistic complexity, involving specialized personnel, ships and costly resources. Additionally, evaluating different solutions at sea at different times might introduce unfairness, in that the underwater acoustic channel is highly dynamic and changes quickly over time. It can therefore be possible that link quality and the actual network topology change drastically even for two consecutive tests, thus making a performance comparison difficult or impossible. As an example, in Figure 1 we depict the variation in time of the acoustic signal strength (via Received Signal Strength Indicator, RSSI) of the link between node 1 and node 4 of an actual network. The two nodes exchange packets of 50B to each other. We considered three different transmission power levels available on the Evologics S2C R 18/34 acoustic modem, namely, 2.8W, 8W and 35W. (The nodes are part of a UWSN deployed for the CommsNet'13 campaign of experiments described below. The distance between these two nodes is around 1.6km.) We can clearly see that at different times the acoustic links can become asymmetric (e.g., at 15000s, at the lowest power) or can completely disappear (e.g., at 10000s, at any power).

Fair comparison of the performance of underwater protocols can be obtained by a hybrid methodology mixing measurements or environmental information from real deployments with simulations. This technique goes under the name of *channel replay*, indicating that a model of the channel derived from real scenarios is used in the controlled and repeatable simulation environment. In this realm, channel simulators are categorized according to the specific method used to model the channel. In particular, van Walree et al. [3] distinguish between the three following possible approaches:

1) Reproduce in simulations the measured channel conditions (*direct replay*).
2) Simulate channel conditions with statistical properties

978-1-4799-4918-2/14/$31.00 ©2014 IEEE



(a) Link from node 4 to node 1.

(b) Link from node 1 to node 4.

Figure 1: Variations in time of the quality of the link between two underwater nodes at three different transmission powers.

similar to those from measurements (*stochastic replay*).

3) Derive simulated channel conditions from environmental (physical) information (*model-based simulations*).

Papers following the listed approaches include [4], [5], [6], where further references can be found.

In this paper we have followed the first approach to compare the performance of two routing protocols for data collection in UWSNs. Channel information has been collected from a real UWSN of 7 nodes deployed off the cost of the Palmaria island, in the vicinity of La Spezia, in the northeastern coast of Italy (Figure 2). Measurements refer to the campaign of experiments “CommsNet’13” performed in September 2013.[1] In particular, we used the simulation/emulation tool SUNSET 2.0 [7] to measure channel quality (i.e., correct packet reception) by having node transmitting data packets in round robin fashion following a TDMA schedule, where each node was transmitting in its own slot. We have explored different packet sizes and modem power levels to measure how these parameters where affecting the channel. The protocols whose performance we compare are the Channel-Aware Routing Protocol (CARP [8]), as exemplary of a cross layer approach to multi-hop routing where control packets are used to access the channel and to choose the best next-hop relay toward a data collector node (sink), and EFlood, an enhanced version of the common flooding protocol where a node wait for a random time before forwarding the packet. CARP and EFlood have been compared using a ray-traced channel model or by replaying real channel traces, under the same network configuration. Results from trials at sea are also reported. Comparing all results shows how channel replay mimics faithfully actual channel dynamics with respect to what is achievable through a simulated channel model, thus demonstrating the effectiveness of this technique for a fair and repeatable performance comparison of solutions for UWSNs.

The rest of the paper is organized as follows. In Section II we describe how we collected measurements about the channel and we prepared it for use in our simulations. Section III describes the results of our performance evaluation of CARP and EFlood. Finally, Section IV concludes the paper.

II. Performing Channel Measurement/Replay

We model the underwater acoustic channel by replaying the channel conditions measured at sea.

To accomplish this task, the first step we took was collecting information about the underwater acoustic channel and the quality of the network links over time. We used the most recent version of the SUNSET framework [7] to perform a probing of the underwater acoustic channel, to store the collected information, and to process it in order to create and use the underwater acoustic channel replay model. During the channel probing phase, the network nodes transmitted data packets in a round robin fashion according to a TDMA schedule where each node had its own slot. The slot duration was set to 4 seconds, inclusive of the maximal transmission and propagation delays plus some guard time. The maximal distance between the nodes was about 2km.

Information about the correct reception at the nodes was collected together with link quality information provided by the acoustic modem, namely, the Evologics S2C R 18/34 [9]. We have explored different packet sizes and modem power levels to measure how these parameters affected the channel. Several channel recording experiments have been performed during CommsNet’13 interleaved with other tests. The longest recording lasted for about 7 hours in the night of September 22nd. These are the measurements used in the performance evaluation (Section III), where we consider a fixed transmission power level.[2] Data packet length was chosen to be of

[1] The experiments were performed through a collaboration agreement between the University of Roma “La Sapienza” and the NATO Science and Technology Organization Centre for Maritime Research and Experimentation (STO CMRE).

[2] Transmission power level 2 of the Evologics modem, namely, 8W.

50 bytes. Control packets were 6 bytes long.[3] TDMA frames (each of 7 slots, one per node) were repeated one after the other. During the odd-numbered frames each node transmitted a data packet, in its assigned slot, while it listened to the channel during the other six slots. The even-numbered frames were used to transmit the shorter packets.

The collected measurements have been used to create a binary channel matrix where for each combination of link and time slot we have 1 if a correct reception occurred, 0 otherwise. To make the channel model as accurate as possible, a second matrix has also been created that captures the actual propagation delays experienced at sea for each link and slot time during the channel probing. Even if the nodes were all static (moored on the sea floor or at the surface), differences in the propagation delay can be due to node drifting around the mooring system or to multi path effects. It is possible that a node missed the first signal arrival (on the direct path) and received another one bouncing off the seafloor or off the sea surface, with some delay.

In order to process the collected information and create a channel representation that can be used for simulations, the two generated matrices have been processed to create a more general model of the underwater acoustic channel. We make the following assumptions:

- The channel is stationary for the duration of two consecutive frames (those for data packet and control packet transmissions). It is as if a snapshot of 14 slots (56 seconds) is taken, and no changes in the channel occur during that time. We call those two frames a *macro-frame*.
- The data about short and long transmissions are merged together in order to have for each instant of time the information on correct packet reception of both short and long messages. We assume that if a correct data reception occurs on a link during the first half of the macro-frame (long packet transmissions) there is also a correct reception of a shorter control packet. In the same way, we assume that if an error occurs during the reception of a short control packet (second half of the macro-frame) there is also an error in the reception a longer data packet. For the other cases, we assume that the same link conditions recorded for long packets during the first half of the macro-frame hold also on the second half of the macro-frame and vice versa for short control packets.
- During the channel probing phase only one single transmission is pushed into the channel during each slot and no overlapping receptions can occur. Overlapping packet receptions have to be therefore handled to model the underwater acoustic channel and run simulations. To address this problem we have considered two different approaches, namely, a *conservative* one and a *non conservative* one. The conservative approach assumes that each time two or more packets overlap at some receiver, a collision occurs and the packets are discarded. In this case we assume that each of the occurring transmissions, even if not correctly decoded at the receiver (i.e., with a 0 in the packet reception channel matrix), produces high interference, thus disturbing other packet receptions. The non conservative approach, instead, assumes that when two or more packets overlap at some receiver only those correctly decoded by the receiver (i.e., with 1 in the packet reception channel matrix) are considered. A collisions therefore occurs only when the reception of two or more correctly decoded packets occurs at the receiver. It is as if packets not correctly decoded are not even heard by the receiver and therefore do not produce interference. The results presented in this paper have been obtained according to the conservative approach. This results in a more accurate channel model for the network scenario considered here, where nodes are not that spread apart from each other, and where each packet transmission significantly impairs the reception of packets transmitted concurrently.

This underwater acoustic channel model has been used to compare the performance of two different routing protocols described in Section III. The design of a more accurate model, which considers not only correct packet reception but also the measurements of the quality of the link provided by the acoustic modem, if any, is currently on-going. These additional information will allow us to improve the current model, thus refining the way overlapping packet reception is treated.

III. Performance Evaluation

This section describes the performance evaluation of two different underwater protocols, namely CARP and EFlood, by means of simulations. CARP, for Channel-Aware Routing Protocol [8], is a cross layer routing solution where nodes exchange control packets to access the channel and to choose the best relay among their neighbors. Once the channel is reserved, optimization is performed by sending a train of data packets and cumulative acknowledgments. CARP takes also advantage of power control, if available at the node, for selecting levels of transmission power so that only reliable links are used when transmitting both short and long packets. EFlood is an enhanced version of the common flooding protocol. The performance of common flooding, i.e., of a broadcast protocol where a node immediately retransmits a received packet (unless it is a duplicate), is enhanced in EFlood by letting a node wait for a random time before forwarding the packet. This random time depends on the network topology and on the packet transmission delay. A CSMA protocol has been used as the MAC layer.

Both protocols have been compared using a ray-traced channel model or by replaying real channel traces, collected as described in Section II. These results are then compared with those obtained from trials at sea to show how well channel replay approximates actual channel dynamics, especially with respect to what achievable through a simulated channel model.

[3] The maximum packet size when using the Evologics modem with the "Instant Messages" feature is 64 bytes. The use of Instant Messages allows us to bypass the in-house MAC protocol that comes with the modems and to transmit at arbitrary times.

A. Simulation Settings

CARP and EFlood have been implemented in SUNSET 2.0 [7], our extension of ns-2 [10] for underwater networking. The two protocols have been compared through different sets of experiments considering the Bellhop ray-tracing software [11] and the real traces we collected at CommsNet'13. Sound speed profiles (SSP), bathymetry profiles and information on the type of bottom sediments of the network deployment area required by Bellhop to compute the acoustic path loss are obtained from the World Ocean Database [12], from the General Bathymetric Chart of the Oceans (GEBCO) [13], and from the National Geophysical Data Center Deck41 data-base [14], respectively.

Figure 2 shows the topology of the CommsNet'13 campaign used to record the channel traces and also as the simulation scenario. The network has 7 nodes: Nodes 1 to 4 have been placed at a depth of about 30m. Nodes 5, 6 and 7 were placed at a depth ranging from 5m to 15m. All nodes were equipped with Evologics S2CR $18/34$ acoustic modems [9]. Each packet successfully delivered to the sink traverses an average of 1.5 hops (the maximum number of hops of a route is 3). Node 1 is designated as the sink, while nodes 3, 4 and 5 are used as data sources. The remaining nodes act as relays between the sources and the sink. Data packets are generated according to a Constant Bit Rate (CBR) traffic. The settings related to the transmission power, bandwidth, frequency, payload size and bit rate have been set according to the specifics of the Evologics S2CR $18/34$ acoustic modems [9]. Data packet are 50B long, consistent with the experiments we performed to collect channel quality measurements. Similarly, we chose control packets of 6B for our implementation of CARP. The modem bit rate is 480bps, which is the available rate for Evologics devices if, as in our case, one wants to use their Instant Messages. The transmission and reception powers have been set to 2.8W and 0.5W, respectively. Even if CARP supports the use of power control, we have used the same transmission power for both control and data packets when running in field experiments. This is because the Evologics S2CR $18/34$ acoustic modems do not feature power control with the fine granularity required by CARP. For fairness of comparison, we did not implement the use power control in both simulation experiments, namely, those involving the ray-traced model and those using channel replay.

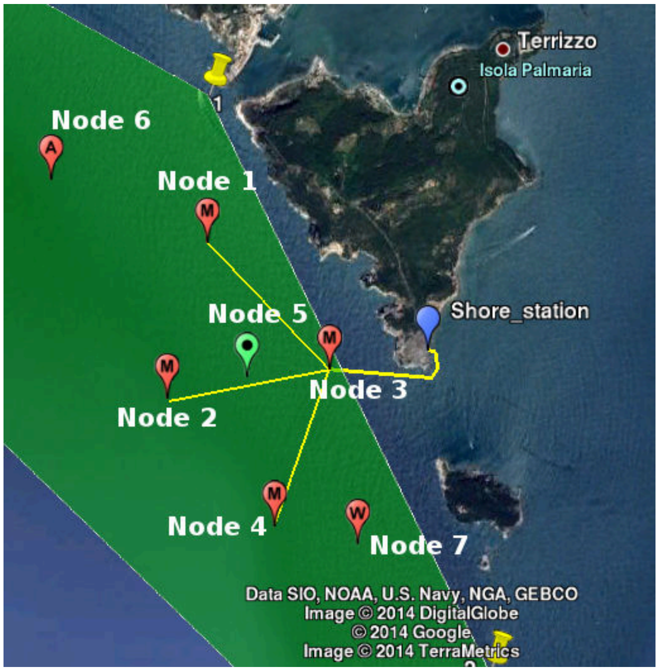


Figure 2: UWSN topology.

B. Simulation metrics

Effectiveness and costs of delivering data to the sink are assessed through the investigation of the following metrics:

- *Packet delivery ratio* (PDR) at the sink, defined as the ratio between the packets correctly received by the sink and the packets generated by the nodes.
- *End-to-end latency*, defined as the time between the packet generation and the time of its correct delivery at the sink.
- *Energy per bit*, i.e., the energy consumed by the network to correctly deliver a bit of data to the sink.

C. Simulation results

Table I shows simulation results for CARP and EFlood in scenarios using the ray-traced ("Bellhop") or the replayed ("Replay") channel models. The traffic load is constant bit rate, with a packet being generated and assigned (randomly and uniformly) to one of the source nodes every 30s ("CBR 30") and 10s ("CBR 10"). The two protocols achieve quite different performance depending on how we modeled the underwater channel. Performance is always worst in the case of results based on channel replay, because of the time-varying quality links measured at sea. The channel variability in time is shown in Figure 3, depicting the number of *active links* in the network for each considered macro-frame (we recall that each macro-frame is 56s long). The definition of "active link" depends on whether the channel model is ray-traced ("Bellhop") or replayed ("Replay"). In the first case, when the Packet Error Rate (PER) on a link between two nodes is low (i.e., less then 0.1) then the link is considered active. (We also report the number of active links when the PER is less than 0.5.) In the second case, a link is active depending on the value of the channel matrix built from the measurements (Section II). In the Bellhop case, we notice that link quality does not vary in time, and that the number of active links is very high. For instance, when the PER is fairly low (e.g., 0.1), the number of active links is always 27, out of the possible 42 links among the 7 nodes in the network. In the case of channel replay, the link quality is highly dynamic and the number of active links is always lower than that of Bellhop. More specifically, the number of active links varies from 5 to 24.

The dynamics of the channel and the different number of active links per frame affect the two protocols in different

Table I: Simulation results.

Metric	CBR 30				CBR 10			
	Bellhop		Replay		Bellhop		Replay	
	CARP	EFlood	CARP	EFlood	CARP	EFlood	CARP	EFlood
Packet Delivery Ratio [%]	100	97.8	97.2	84.7	98.3	80.5	94.6	76
End-to-end latency [s]	5.87	5.4	33	5.8	21.7	7.89	73	6.19
Energy per bit [J/b]	0.024	0.062	0.034	0.035	0.021	0.059	0.027	0.034

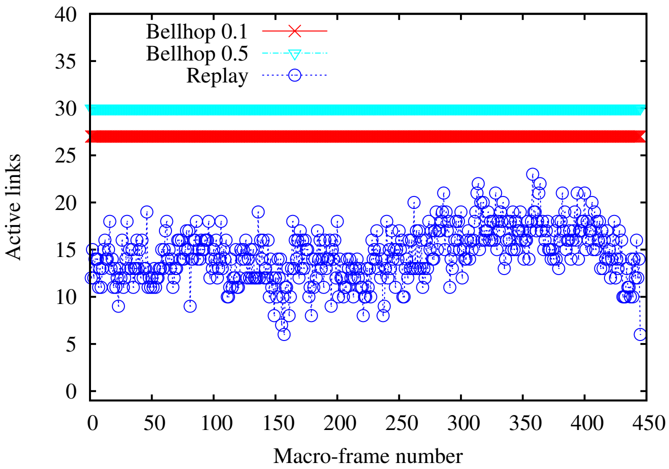


Figure 3: Number of active links per macro-frame.

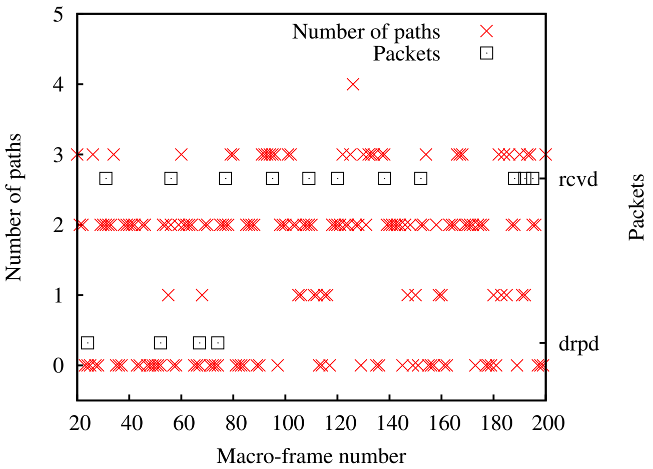


Figure 4: EFlood: Packets from node 4 to the sink.

ways. Table I shows that the PDR of EFlood decreases for increasing traffic and passing from Bellhop to channel replay. In the case of the Bellhop ray-tracer, since the the number of active links is very high (Figure 3), the PDR decreases when passing from a low traffic (CBR 30) to a higher traffic (CBR 10) because of the higher number of packet collisions and the higher probability to find the channel busy, which results in a higher average latency to deliver data towards the sink. When we use channel replay the number of active links is lower, which corresponds to a lower number of neighbors per node and therefore to a lower amount of packets collisions. In this case, the lower PDR depends mostly on the absence of routes to the sink, due to the high time-varying quality of the links. As an example, in Figure 4 we depict the packets dropped (drpd) and those successfully received (rcvd) by the sink among the packets generated by node 4. With the crosses we indicate the number of different routes between node 4 and the sink in each macro-frame. For instance, we observe that in the macro-frames from number 47 to number 52 there is no route, while there are two routes during the macro-frames from number 38 to number 42. With the squares we instead indicate whether a packet generated in a macro-frame has been dropped or successfully delivered. As expected, all packets that have been dropped have been generated in frames where no route exists from node 4 to the sink. If such routes do exist, then the data packet will be delivered with very low latencies, as can be noticed from Table I.

The PDR of CARP is high, irrespective of channel modeling. Independently of traffic, packets are not lost because of collisions as in the case of EFlood. This is due to the channel reservation mechanism of CARP and the use of acknowledgment packets, which realizes collision avoidance quite effectively, and to the fact that nodes can transmit trains of packets (if available), thus fully exploiting the channel once it is acquired by a node. The latencies experienced by packets transported according to CARP are generally higher than those of EFlood. When the channel is ray-traced latency is sensitive to traffic. At low traffic, the latency experienced by CARP packets is just a little bit higher of that of EFlood packets. This is because of the handshaking mechanism needed by CARP nodes for acquiring the channel prior to data packet transmission. When traffic increases, latency also sensibly increases: Because of the higher number of packets in the network, collisions are higher, requiring further time for packet retransmission. When the channel is replayed, latency increases much faster then when the channel is ray-traced. This is clearly a consequence of the high link quality variability obtained through channel replay. Each time a control or a data packet is lost due to channel quality variation, the CARP handshake is repeated after a backoff time that increases the latency. In particular, we observed that the ratio between the number of channel access attempts and the successful transmission of a (train of) data packet(s) is 68% higher in the case of the replayed channel than when we use Bellhop. In other words, with channel replay an average of 3.54 PING packet transmissions are needed to successfully reserve the channel, while only 2.18 transmissions of such packets are sufficient when the channel is ray-traced.

Figure 5 shows that the data packets successfully delivered by CARP with the highest latency are those generated in the macro-frames where no routes are available from a source node to the sink, i.e., macro-frames from number 65 to number 67 and from number 71 to number 74. In particular, we consider the data packets generated by node 4.

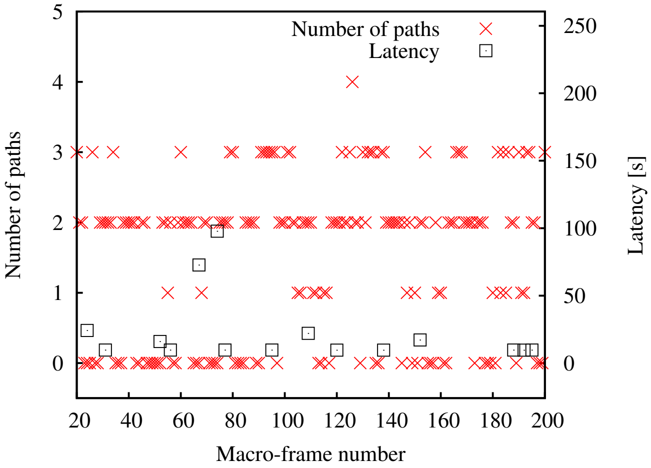


Figure 5: CARP: Latency of packets from node 4 to the sink.

Table I shows also that the energy per bit of both protocols is higher when using the ray-traced model than when using channel replay. This is due to the higher number of active links, which in turn results in a higher number of neighbors per node. In the case of using EFlood, a higher number of active links induces a higher number of re-transmissions, i.e., higher energy consumption. Similarly, when using CARP, higher energy is required by a higher number of nodes contending for channel access. When the traffic increases, the energy per bit decreases for both protocols. This is because the number of packets transmitted and received in the network is lower, due to the high number of packet collisions and to a high probability to find the channel busy.

D. Trials at sea

In this section we describe results obtained during the campaign of trials at sea called *CommsNet'13*. The experiments were performed on a day different from the one when we recorded the channel traces used in our simulations. Moreover, the network topology was also slightly different, as it included an additional node, namely, node 8, as shown in Figure 6. The performance of CARP and EFlood has been compared considering the same bit rate, packet size, traffic and power consumption values as described above. Node 1 was designated as the sink; nodes 3, 4 and 8 were used as data sources. Results are shown in Table II.

In these experiments we observed a variation of the link quality which was more limited with respect to that recorded in the day when we collected measurements for channel replay. As a consequence, EFlood obtains better results than those shown through simulations with the replayed channel. In the trials at sea each node was able to communicate with several of its neighbors, resulting in a higher number of routes from each source node to the sink and in a higher energy consumption. However, as observed via simulations with channel replay, several packets are lost due to the variations of the channel quality, which impair routes to the sink. Moreover, due to the higher number of nodes involved in the communications, the latency increases because of the extra random time a node waits before broadcasting a packet. Both in channel replay-based simulations and trials at sea, the PDR of EFlood depends more on channel quality variations (which impairs routes to the sink) than on higher traffic and therefore on collisions, as it is instead the case with ray-tracing-based simulations.

The results for the performance of CARP are similar to those obtained by simulations with channel replay for all the considered metrics. We can see a quite accurate matching in terms of PDR and energy consumption and some difference in the packet latency due to the presence of more active links. With respect to the Bellhop-based simulation, instead, we can notice how the average packet latency experienced during the in filed tests is much higher than the one from simulations.

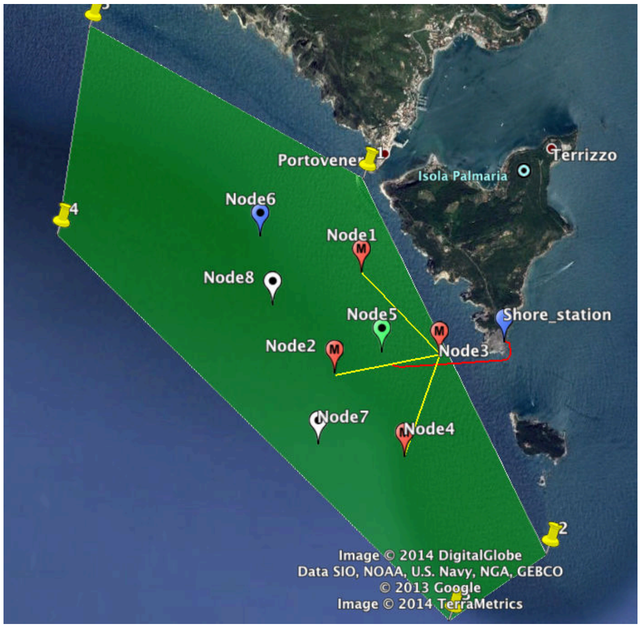


Figure 6: Network topology of the CommsNet'13 experiments.

Table II: Results from trials at sea.

Metric	CBR 30		CBR 10	
	CARP	EFlood	CARP	EFlood
Packet Delivery Ratio [%]	96	96	95	87
End-to-end latency [s]	37	9	52	10
Energy per bit [J/b]	0.03	0.049	0.027	0.04

This is due to the presence of variations on the link quality which imposes a high number of retransmissions for the CARP handshaking, needed to access the channel. For the same reasons, a lower PDR and a higher energy consumption are experienced in field.

IV. Conclusions

We have presented a comparative performance evaluation of two routing solutions for underwater wireless sensor networks (UWSNs), namely, CARP (for a cross layer approach to data gathering) and EFlood, a simple enhancement to common flooding. Simulations and testbed-based results from a UWSN network deployment off the coast of Italy were considered. Our simulation results are based on a channel model obtained "synthetically" via a ray tracer and by replaying real channel traces obtained by measurements at sea. Results are also reported from campaigns of experiments at sea. All results clearly show that channel replay approximates actual channel dynamics quite better than what can be obtained through ray tracing-based simulations. We therefore conclude that channel replay allows a fair and repeatable performance comparison of solutions for UWSNs.

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." We gratefully acknowledge the NATO STO CMRE for invaluable feedback and for providing expertise and logistics during experimentation.

References

[1] 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.

[2] T. Melodia, H. Kulhandjian, L.-C. Kuo, and E. Demirors, "Advances in underwater acoustic networking," in *Mobile Ad Hoc Networking: Cutting Edge Directions*, S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic, Eds. Hoboken, NJ: John Wiley & Sons, Inc., March 5 2013, ch. 23, pp. 804–852.

[3] P. A. van Walree, T. Jenserud, and M. Smedsrud, "A discrete-time channel simulator driven by measured scattering functions," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 9, pp. 1628–1637, December 2008.

[4] R. Otnes, P. van Walree, and T. Jenserud, "Validation of replay-based underwater acoustic communication channel simulation." *IEEE Journal on Oceanic Engineering*, vol. 38, no. 4, pp. 689–700, October 2013.

[5] F.-X. Socheleau, C. Laot, and J.-M. Passerieux, "Stochastic replay of non-WSSUS underwater acoustic communication channels recorded at sea," *IEEE Transactions on Signal Processing*, vol. 59, no. 10, pp. 4838–4849, October 2011.

[6] P. Casari, D. Spaccini, G. Toso, B. Tomasi, R. Petroccia, C. Petrioli, and M. Zorzi, "A study on channel dynamics representation and its effects on the performance of routing in underwater networks," in *Proceedings of the IEEE Asilomar Conference on Signals, Systems and Computers 2012*, Pacific Grove, CA, USA, November 4–7 2012, pp. 1536–1540.

[7] C. Petrioli, R. Petroccia, and 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, November 11–13 2013, pp. 1–8.

[8] 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, pp. 1–9.

[9] Evologics, "Evologics S2C acoustic modems." [Online]. Available: http://www.evologics.de/

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

[11] M. Porter *et al.*, "Bellhop code." [Online]. Available: http://oalib.hlsresearch.com/Rays/index.html

[12] "World ocean atlas," www.nodc.noaa.gov/OC5/WOA05/pr_woa05.html.

[13] "General bathymetric chart of the oceans," www.gebco.net.

[14] "National geophysical data center, seafloor surficial sediment descriptions," http://www.ngdc.noaa.gov/mgg/geology/deck41.html.