

A Study on Channel Dynamics Representation and its Effects on the Performance of Routing in Underwater Networks

Paolo Casari^{*†}, Daniele Spaccini[§], Giovanni Toso^{*†}, Beatrice Tomasi^{*},
Roberto Petrocchia[§], Chiara Petrioli^{§†}, Michele Zorzi^{*†}

^{*}Department of Information Engineering, University of Padova, via Gradenigo 6/B, I-35131 Padova, Italy

[§]Department of Computer Science, University of Rome “La Sapienza”, via Salaria 113, I-00198 Rome, Italy

[†]Consorzio Ferrara Ricerche, via Saragat 1, I-44122 Ferrara, Italy

[‡]Consorzio Interuniversitario Nazionale per l’Informatica (CINI), Roma, Italy

^{*}{casari,p,tosogiov,tomasibe,zorzi}@dei.unipd.it [§]{spaccini,petrioli,petrocchia}@di.uniroma1.it

Abstract—We consider an underwater networking scenario, and test the performance of two multihop routing paradigms, source routing and hop-by-hop relay selection, in the presence of different representations of the channel dynamics. We focus on a static channel case (obtained via empirical equations for path-loss), and on a sequence of channel realizations obtained using ray tracing, that vary both slowly and rapidly over time with respect to the expected reaction time of routing protocols; the two latter cases are also explored in the presence both of a flat bottom and of a rough bottom with several seamounts, to yield a total of five different channel models.

Our results show that channel variations induced by environmental changes over time have an impact on routing performance metrics in connected topologies. A sea bottom with a rough shape adds a further impact to the routing performance, which is shown to be larger for source routing. We conclude that while empirical channel models yield a good first-order approximation, the time-variability of the channel and the shape of the network area boundaries are to be taken into account in order to achieve more realistic network performance estimates.

I. INTRODUCTION AND RELATED WORK

The recent improvements of underwater acoustic transmission techniques [1] and the development of high-performance underwater modems have fostered interest on the use of networked underwater devices to support such applications as monitoring, movement detection and tracking, site survey, etc. However, prototyping such systems and deploying them for proof-of-concept tests is still an expensive task involving costly hardware and specialized personnel for supporting sea trials. On one hand, this makes simulations a very valuable tool for checking the feasibility of the designed solutions; on the other hand, it keeps the problem of providing realistic simulation results open and timely.

It has been argued that, compared to more realistic channel propagation snapshots provided, e.g., via ray tracing, the simple empirical equations for the path-loss experienced by acoustic waves under water may not yield sufficient precision, not only quantitatively, but also qualitatively [2]. In addition, it has been shown that a precise representation of the channel behavior requires to account for such propagation phenomena as the Doppler spread, for which a widely agreed statistical model does not yet exist; therefore, these phenomena can only be measured at sea, and stochastically replicated later in simulations [3].

With regard to network performance studies, in [4] the authors resorted to ray tracing, instead of empirical models, for predicting the capacity of an underwater seabed network. Further work suggested that it is important not only to resort to

the better accuracy provided by ray tracing, but also to account for channel variations induced by environmental changes in the networking area [5]; the authors also show that some degree of knowledge of these changes would improve the performance of multihop routing protocols. The timing of underwater modems and the additional delays incurred in such operations as receiver-side processing and interfacing with modem electronics also play a role in the accuracy of simulations, as explained in [6].

Prompted by the above results, in this paper we consider the interplay between channel variability and routing protocols for underwater acoustic networks. Rather than just considering the variation of the acoustic power attenuation over time as in [5], we study what variations have a significant impact on routing performance, and should therefore be taken into account when modeling the acoustic channel for simulations. In more detail, we test the performance of two multihop routing paradigms, source routing and hop-by-hop relay selection, for five different channel models, each involving a different representation of the channel dynamics. We focus on a static channel case (obtained via empirical equations for path-loss), and on a sequence of channel realizations obtained using ray tracing based on KAM11 data [7]. These realizations vary both slowly and rapidly over time with respect to the reaction time of routing protocols, and are computed in the presence of both a flat and a rough bathymetry profile. We employ this dataset to simulate the performance of routing protocols in a multihop network deployment, where the average distance between subsequent relays is progressively increased.

II. PROTOCOL DESCRIPTIONS

A. Source routing for Underwater Networks (SUN)

SUN is a reactive source routing approach inspired to DSR [8]; however, with respect to DSR, SUN provides several improvements that have been found to be useful in underwater networks [9].

The SUN protocol separately addresses the behavior of sinks and nodes. The sinks perform only two tasks: they collect the data generated in the network and advertise their presence to the nodes within their communications range by periodically sending probe messages. This is the only proactive feature of the SUN protocol, and allows the sink to offload the task of answering path requests to its own neighbors. The nodes receiving sink probes are called “end nodes.” The sink sends probes periodically, hence the set of end nodes is periodically refreshed: this also provides support for sink mobility.

All other nodes can send data, ask for paths, answer path requests, act as relays and report broken routes. In SUN, every node maintains a buffer with packets to be forwarded, part generated by the node itself, and part to be advanced towards the sink on behalf of other neighbors. An agent checks the buffer periodically, and if any packets are present, they are served according to a First-In-First-Out policy. The behavior of the agent is different according to the hop count of the node: end nodes (hop count 1) send the packets directly to the sink; the nodes with hop count 2 or more are not neighbors of the sink, but are aware of a valid path to the sink, which can hence be reached via multihop relaying. If the agent runs on a node that knows no valid path to the sink, a path discovery process is triggered. In this case, the complete route to be followed is specified within the packet header, according to the source routing paradigm.

The path discovery process works similarly to wireless radio source routing, but stops at end nodes instead of stopping at the sink. In particular, a node starts the process by sending a path request, which is flooded until any end nodes are reached. Every node that replicates the path request writes its own address in the header of the packet. This process eventually provides a complete route description as the request reaches the sink. Loops are avoided through a stateful inspection of path requests, whereby a relay never forwards the same request twice. At this point, end nodes create a path reply packet, which is then transmitted back to the sink following the reverse of the path request route, as read from its header.

A node may receive multiple replies to the same path request. This makes it possible for the node to choose the best route according to a given metric. In this work, we pick the route which maximizes the minimum SNR throughout all links of the route. The nodes can store valid routes and use them to replace broken routes upon reception of path requests. However, all routes have an expiration time (set here to 12 min based on empirical tuning): if a path request is received and all known routes have expired, the path request is furthered towards the end nodes.

For efficiency reasons, SUN administers data packet re-transmissions on behalf of the link-layer. This is because it cannot be taken for granted that every lower-layer protocol implements error control, and that only data packets are actually retransmitted. Correct packet receptions are confirmed via ACK packets. Failed attempts are hence detected because of ACK timeout expiration, in which case retransmissions occur through a Stop-and-Wait scheme up to a maximum number of attempts. If all retransmissions fail, the node will infer that the path is broken, and will report the issue to the nodes upstream in the route by sending a path error packet. This makes it possible to perform a fresh route discovery and come up with a new route.

B. Channel Aware Routing Protocol (CARP)

The Channel-Aware Routing Protocol (CARP) [10] is a distributed cross-layer solution for multi-hop delivery of data to a sink in underwater networks. Next hop selection takes explicitly into account the history of data packet delivery, the link quality and how successful a neighbor has been in forwarding

data towards the sink. Moreover, power management is used when transmitting control and data packets in order to select only robust links when forwarding data towards the sink. When the network is deployed, CARP starts flooding HELLO packets from the sink through the network, thus allowing each node to estimate the number of hops needed to reach the sink (hop count). After the setup phase, the protocol and the information gathered by the nodes start evolving as packets flow in the network. When a node x has one or more data packets to transmit, a PING packet is broadcast to its neighbors, carrying information about node x 's ID, its estimated hop count and the number of data packets it would like to transmit. If more than one packet has to be transmitted, a *train* of data packets is transmitted in sequence. After sending a PING packet, node x awaits to receive a reply (PONG packet) from its neighbors for a time δ . The waiting time δ is initially set depending on the modem nominal transmission range and on the acoustic signal propagation speed in water. It is then continuously updated by using the actual round trip time of PING/PONG handshakes.

Upon receiving a PING packet, a node y replies with a PONG packet to node x (unicast communication). Each PONG packet contains several pieces of information about node y : i) its residual energy; ii) the number of data packets that it can store in its buffer; iii) its hop count $HC(y)$; iv) its *goodness* (lq_y), which is a measure of how good the node is as a relay, based on the success of past transmissions to its neighbors. The *goodness* is defined as an exponential moving average, whereby the transmissions performed in the past are less influential than more recent ones. This enables CARP to take into account the time-varying nature of the channel, giving more importance to what has happened recently.

The selection of the next hop relay is performed at node x after the time δ according to the information carried by the received PONG packets. Node x collects the link quality information lq_y for every neighbor node y that replied to x 's request, and combines it with the quality of the link from x to y , $lq_{x,y}$. In particular, for each responding y , node x computes: $goodness_y = lq_y lq_{x,y}$.

This value represents an estimate of the quality of the channel from x to y and from y to its best reachable neighbor in a route to the sink. The node y with the highest ratio $goodness_y + \frac{1}{HC(y)}$ is chosen as the relay, and a (train of) data packet(s) sized so as not to exceed the neighbor buffer is sent directly to it. In doing so, nodes that are good candidates to forward data to the sink and with a lower hop distance from the sink are preferred. Nodes with a higher hop count are chosen only if their link quality is significantly better than those closer to the sink. If there are ties, priority is given to the node with the highest energy, and then to the node with the higher available buffer space. Further ties are broken by using the nodes' unique identifiers. Upon receiving a train of data packets, a node y replies with a cumulative ACK, acknowledging each packet in the train (bit mask). Each node continuously updates its hop count information according to the PING, PONG and ACK packets it receives. In this way, the hop count information is dynamically updated according to possible changes of the network topology. When node y has received a train of one or more data packets, it checks whether

it has received them previously, so as to re-transmit only those that it has not forwarded already.

One of the key features of the CARP protocol is that power management is used to allow the selection of robust links when forwarding data towards the sink. Data packets are usually longer than control ones. If a relay node is selected based on the correct transmission of short control packets, it is then possible that the same link will be really unreliable when long data packets want to be transmitted. Therefore, in CARP, the transmission of control packets takes place at a lower power than that used for sending data packets. This is done in order to achieve similar packet error rates for both data and control packets. Therefore, CARP makes it possible for data packets from x to y to have the same probability of success than that of PING packets from x to y by gauging its transmission power appropriately.

III. SIMULATION SCENARIOS

The simulation scenario considered in this study involves a network of 12 nodes which relays data packets from a source to a sink, located at opposite sides of the network area. This area has size $3L \times 4L$, and is divided into 12 $L \times L$ cells. One relay node is deployed uniformly at random within each cell, as in Fig. 1. All nodes are assumed to be at the same depth of 45 m.

In order to study how different channel representation techniques affect the performance of routing protocols, we consider the following five models for computing the attenuation incurred by the acoustic signals: *i*) the empirical path-loss equations summarized in [11], and shortly denoted as the ‘‘Urlick’’ model in this paper; *ii*) and *iii*) the ‘‘coherent’’ transmission loss predicted by the Bellhop ray tracing software [12] in a flat bottom environment, where the channel realizations are updated respectively once every 5 s and once every 3600 s; *iv*) and *v*) the ‘‘coherent’’ transmission loss predicted by Bellhop in the presence of a very irregular bottom; again the channel realizations are updated once every 5 s and once every 3600 s, respectively.

The Bellhop runs are supplied with realistic measurements of the sound speed profile (SSP) taken during the KAM11 sea trials [7]. These SSPs were measured once every 5 s, which is in line with the configuration of *ii*) and *iv*) above. For *iii*) and *v*) we subsampled the SSP data set to yield one sample per hour.

The scenario employed for computing the channel response in *ii*) and *iii*) involves a flat bottom with muddy/rocky sediments where the bottom depth is 100 m (this is also in line with the KAM11 environment). For *iv*) and *v*) we created a random bottom profile with many seamounts raising up to a depth of around 50 m. The main macroscopic difference between these two configurations is that in the latter a certain amount of energy is reflected backwards by the seamounts, which yields an attenuation about 10 dB higher (on average) than in the former dataset.

We perform a Bellhop run for each SSP in the KAM11 dataset, and for several values of the transmitter depth (from 5 to 95 m in steps of 5 m), resulting each time in a different channel realization. This realization is sampled every 50 m

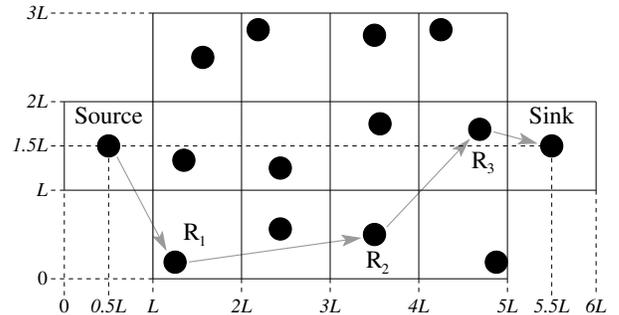


Figure 1. The scenario considered in this paper: 12 relay nodes are randomly placed within a $3L \times 4L$ area, with the source and the sink located at opposite sides. L is varied in order to change the average hop distance (from [13]).

up to 10 km in the range dimension, and every 5 m up to 100 m in the depth dimension. The resulting channel power gain data set is therefore a 4-D matrix indexed by the source depth, the receiver depth, the receiver range and the time. In this paper, we consider all channel realizations to be invariant with respect to rotation and translation. While this is not a realistic assumption in general, we remark that our objective in this paper is to check how different models for channel variations in time and space affect the performance of routing protocols: our assumption is adequate for this task.

The communications take place in a band of 4 kHz centered around a carrier frequency of 25 kHz, using a 2048-bps uncoded Frequency-Shift Keying (FSK) modulation format. The size of a data payload is 8192 bits. The transmitter-source power level (SPL) is set to 150 dB re μPa . CARP reduces this to 148 dB re μPa for control packets. For the computation of communications performance figures such as Signal-to-Interference and Noise Ratios (SINRs), we employ the transmission loss, estimated via one of the models *i*)–*v*), and the empirical equations for the noise power spectral density in [11]. A threshold of 1 dB of SINR is set to discriminate the packets that are actually detected by the receiver from the packets that are unheard because of noise or interference. For detected packets, an independent bit error model involving the incoherent FSK bit error equations is employed to derive the packet error rate (PER). Erroneous packets are retransmitted up to 4 times. The source generates 1 packet every 2 minutes, so that we can neglect any issues related to excessive traffic and focus on the interplay between the representation of channel dynamics and the performance of routing protocols.

IV. SIMULATION RESULTS

We start our analysis from Fig. 2, which shows the throughput (defined as the number of correct information bits that are correctly delivered to the sink per unit time) experienced by the SUN protocol. The throughput is reported as a function of the cell side length L : we recall that the number of relay nodes in our scenario is fixed to 12, hence increasing L increases the average distance to be covered by one hop. The lowest value of L is 500 m, which corresponds to a very likely direct connection between the source and the sink. While the simulations have been run for values of L up to 3500 m,

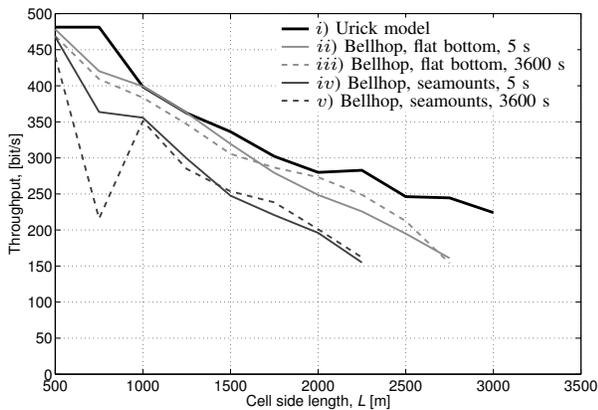


Figure 2. Throughput achieved by the SUN protocol vs. L .

the probability of delivering packets to the sink decreases progressively for increasing L . This means that the link quality oscillates too fast for SUN to take advantage of good links as they occur. Such regimes have been trimmed in the figures, as they represent an undesirable protocol behavior.

Fig. 2 contains 5 curves, one for each of the datasets described in Section III. The Urick model curve, or case *i*), corresponding to a static channel model, is represented by a solid black curve; the model employing the Bellhop ray tracer in a scenario with a flat bottom, or cases *ii*) and *iii*), is represented using light grey; the model employing the Bellhop ray tracer in a scenario with seamounts, or cases *iv*) and *v*), is represented using dark grey; in particular, the solid curve refers to a channel that varies once every 5 s, whereas the dashed curves to a channel that varies once every 3600 s.

For all values of L considered here, the SUN protocol behaves best when the channel is stable throughout the simulation, as with the Urick model: in this case any discovered routes are also stable, and the amount of control traffic required for rediscovering the routes is lower. With Bellhop, the flat bottom case provides better performance than the seamount case: as explained in Section III, the latter leads to higher attenuation which, in general, tends to make packet delivery more difficult (e.g., the throughput is about 17% lower at $L = 1500$ m with respect to the flat bottom case). This intuition is also confirmed by the fact that the seamount curves behave basically like the flat bottom curves, but for values of L about 1000 m shorter. There are two notable behaviors to highlight in Fig. 2: the first is that dashed curves (representing slow channel variations) tend to overcome solid curves as L increases; the second is the locally low throughput experienced in case *v*) at $L = 750$ m. The reason behind these behaviors is that SUN periodically rebuilds routes, and every time this requires a flooding process, which in dense topologies gives rise to interference between the replicas transmitted by different relays. As L increases, the higher attenuation experienced in cases *iii*) and *v*) makes such interference less likely, decreasing the time required to find a reliable route, and increasing throughput as a consequence. The sequence of events leading to lower throughput for case *v*) at $L = 750$ m is similar: at $L < 750$ m, the sink is often within coverage of the source, and there is rarely any actual route discovery; for $L = 750$ m, the source is not directly

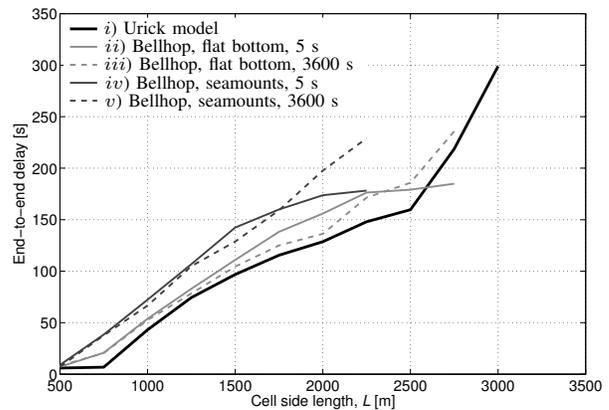


Figure 3. End-to-end delay for the SUN protocol vs. L .

connected to the sink, however most of the relays are end nodes: when they receive a route request, the interference between route replies can be very significant and lead to the locally lower throughput observed in Fig. 2. We also observe that the channel varies only once every 3600 s, which makes good channel realizations more likely and more stable at this distance, which also contributes to interference. For $L \geq 1000$ m, this effect is mitigated by the larger average distance between the relays, which exposes transmissions to both higher attenuation and longer propagation delays, reducing the possibility that interference affects the route discovery process.

Similar results can be observed for the end-to-end delivery delay experienced by the SUN protocol: the latency of cases *ii*) and *iii*) is slightly worse than with the Urick model, and further decreased in cases *iv*) and *v*). Cases *iii*) and *v*) tend to overcome cases *ii*) and *iv*), respectively, for increasing L . We note that the locally lower throughput experienced in case *v*) at $L \geq 750$ m does not lead to a locally higher delay: this confirms the intuition that it is the route discovery process that is impaired by interference, not the delivery of data packets.

Let us now proceed with the CARP protocol, and consider its throughput, shown in Fig. 4 as a function of L . We recall that, unlike SUN, CARP is based on a hop-by-hop relay selection process. Therefore, CARP is affected by channel variations whenever they occur during the selection of the next hop. More precisely, a node can successfully forward its data packet(s) to the next-hop relay if, while it is completing the handshake procedure and delivering the message the channel variation does not affect the quality of the selected link. When the channel conditions change faster than the time needed to forward the data to the selected relay, the probability that these changes impair the reliability of the selected link is higher. In case *i*) (Urlick), the channel does not vary over time, hence we always obtain a higher throughput, with respect to the case of a time-varying conditions. The end-to-end delay experienced by CARP (see Fig. 5) is also shorter in case *i*) than in all other cases. In the same way, cases *iii*) and *v*) (channel updated once every hour) result in better performance than achieved in cases *ii*) and *iv*) (channel updated once every 5 s), experiencing a throughput 10% higher and an average end-to-end delay up to 150% shorter. In the case of fast variations the number

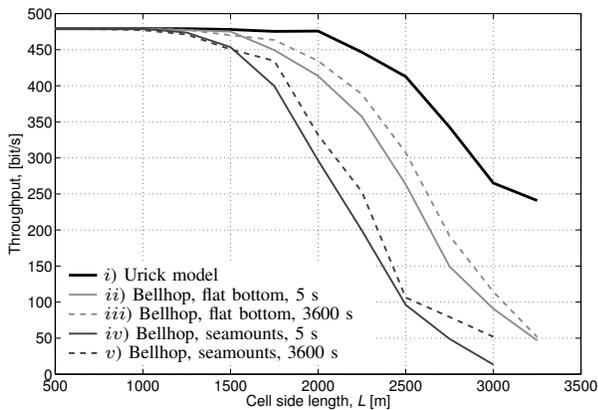


Figure 4. Throughput for the CARP protocol vs. L .

of attempts needed to find a relay and deliver the data over the selected link is up to $\sim 30\%$ ($\sim 50\%$) higher than in the case of slow variation when a flat (irregular) bottom is considered. Increasing the number of PING and data packet re-transmissions reduces the throughput and increases both the packet delivery delay and the protocol overhead. Moreover, when the average packet latency becomes longer than the data generation rate, more packets are in the channel at the same time, leading to heavier interference. Another important aspect affecting the protocol performance is that increasing L increases the average distance among the nodes. Since the transmission power is fixed and does not change with the cell side, increasing the distance among the nodes increases the attenuation affecting the transmitted signal, which results in a lower SNR at the different receivers and in longer routes to the sink (from 1 hop for $L = 500$ m up to 6-7 hops for $L > 2500$ m). When the link quality becomes poor, the probability to find unreliable links or links more strongly affected by channel variation becomes higher. Figs. 4 and 5 clearly show this trend, as the protocol performance decreases with increasing L in all cases i)– v). Moreover, when longer links are considered, the time to complete the handshake procedure and deliver the data becomes larger, thus increasing the probability that the quality of the selected link changes over time. In case of fast variation of the channel, when the cell side increases the number of attempts to find a relay varies from $\sim 1\%$ to $\sim 30\%$ when a flat bottom is considered and from $\sim 1\%$ to $\sim 80\%$ with an irregular bottom. Finally, for $L \geq 3250$, the network becomes disconnected and no data can be delivered to the sink.

V. CONCLUSIONS

In this work, we tested the performance of source routing and hop-by-hop relay selection in underwater networks, in the presence of five channel models, each involving a different representation of the channel dynamics. We considered a network deployment where the number of relays is fixed and the network area size is progressively increased, which in turn increases the impact of channel variations.

Our results show that the performance of both routing paradigms changes significantly in the presence of time-varying channels and irregular boundaries of the network

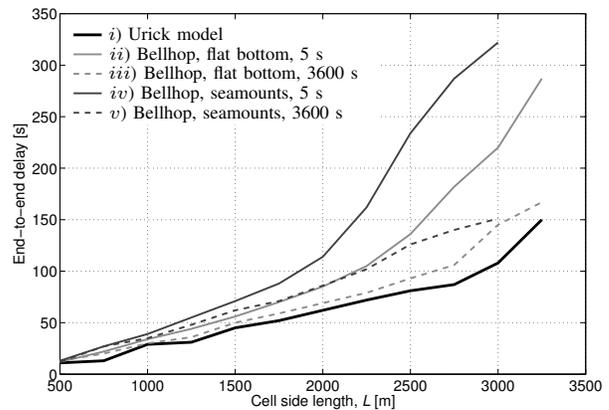


Figure 5. End-to-end delay for the CARP protocol vs. L .

area. We conclude that both effects are to be taken into account when designing routing protocols and simulating their performance.

ACKNOWLEDGMENTS

This work has been supported in part by the European Commission under the 7th Framework Programme (G.A. 258359–CLAM), by ONR under grants no. N00014-05-10085 and N00014-10-10422, and by the Italian Institute of Technology within the Project SEED framework (NAUTILUS project).

REFERENCES

- [1] M. Chitre, S. Shahabudeen, and M. Stojanovic, "Underwater acoustic communications and networking: Recent advances and future challenges," *Marine Tech. Soc. Journal*, vol. 42, no. 1, pp. 103–116, 2008.
- [2] P. Casari and M. Zorzi, "Protocol design issues in underwater acoustic networks," *Elsevier Comp. Commun.*, vol. 34, no. 17, pp. 2013–2025, 2011.
- [3] R. Otnes, P. van Walree, and T. Jensen, "Validation of direct and stochastic replay using the Mime acoustic communication channel simulator," in *Proc. of UCOMMS*, Sestri Levante, Italy, Sep. 2012.
- [4] P. King, R. Venkatesan, and C. Li, "A study of channel capacity for a seabed underwater acoustic sensor network," in *Proc. of MTS/IEEE OCEANS*, Québec City, Canada, Sep. 2008.
- [5] S. Azad, P. Casari, R. Petroccia, C. Petrioli, and M. Zorzi, "On the impact of the environment on MAC and routing in shallow water scenarios," in *Proc. of IEEE/OES Oceans*, Santander, Spain, May 2011.
- [6] C. Petrioli, R. Petroccia, and J. Potter, "Performance evaluation of underwater MAC protocols: From simulation to at-sea testing," in *Proc. of IEEE/OES OCEANS*, Santander, Spain, Jun. 2011.
- [7] W. S. Hodgkiss and J. C. Preisig, "Kauai Acomms MURI 2011 (KAM11) experiment," in *Proc. of ECUA*, Edinburgh, Scotland, Jul. 2012.
- [8] D. B. Johnson and D. A. Maltz, *Mobile Computing*. Kluwer Academic Publishers, February 1996, ch. Dynamic source routing in ad hoc wireless networks, pp. 153–181.
- [9] 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.
- [10] S. Basagni, C. Petrioli, R. Petroccia, and D. Spaccini, "Channel-aware routing for underwater wireless networks," in *Proc. of IEEE/OES OCEANS*, Yeosu, Korea, May 2012.
- [11] M. Stojanovic, "On the relationship between capacity and distance in an underwater acoustic communication channel," *ACM Mobile Comput. and Commun. Review*, vol. 11, no. 4, pp. 34–43, Oct. 2007.
- [12] M. Porter et al., "Bellhop code." [Online]. Available: <http://oalib.hlsresearch.com/Rays/index.html>
- [13] B. Tomasi, G. Toso, P. Casari, and M. Zorzi, "On the impact of time-varying acoustic channels on routing protocols for underwater networks," in *Proc. of UCOMMS*, Sestri Levante, Italy, Sep. 2012.