Harnessing HyDRO: Harvesting-aware Data ROuting for Underwater Wireless Sensor Networks

Stefano Basagni
ECE Dept., Northeastern University
Boston, MA
basagni@ece.neu.edu

Valerio Di Valerio
Dip. di Informatica, Università di Roma "La Sapienza"
Rome, Italy
divalerio@di.uniroma1.it

Petrika Gjanci
Dip. di Informatica, Università di Roma "La Sapienza"
Rome, Italy
gjanci@di.uniroma1.it

Chiara Petrioli
Dip. di Informatica, Università di Roma "La Sapienza"
Rome, Italy
petrioli@di.uniroma1.it

ABSTRACT

We demonstrate the feasibility of long lasting underwater networking by proposing the smart exploitation of the energy harvesting capabilities of underwater sensor nodes. We define a data routing framework that allows senders to select the best forwarding relay taking into account both residual energy and foreseeable harvestable energy. Our forwarding method, named *HyDRO*, for *Harvesting-aware Data ROuting*, is also configured to consider channel conditions and route-wide residual energy, performing network wide optimization via local information sharing. The performance of our protocol is evaluated via simulations in scenarios modeled to include realistic underwater settings as well as energy harvesting based on recorded traces. HyDRO is compared to state-of-the-art forwarding protocols for underwater networks. Our results show that jointly considering residual and predicted energy availability is key to achieve lower energy consumption and latency, while obtaining much higher packet delivery ratio.

CCS CONCEPTS

• **Networks** → **Network protocol design**; **Routing protocols**;
• **Hardware** → **Power and energy**; **Renewable energy**;

KEYWORDS

Underwater Wireless Sensor Networks, underwater energy harvesting, reinforcement learning-based routing.

ACM Reference Format:
Stefano Basagni, Valerio Di Valerio, Petrika Gjanci, and Chiara Petrioli. 2018. Harnessing HyDRO: Harvesting-aware Data ROuting for Underwater Wireless Sensor Networks. In *Mobihoc '18: The Eighteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing, June 26–29, 2018, Los Angeles, CA, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3209582.3209610

Mobihoc '18, June 26–29, 2018, Los Angeles, CA, USA

ACM ISBN 978-1-4503-5770-8/18/06…$15.00
https://doi.org/10.1145/3209582.3209610

1 INTRODUCTION

Freeing communications from costly and range-limited cabling, underwater wireless technology and networking has the potential to propel scientific and industrial underwater applications into a new era of purpose and innovation [11]. Overall, it is widely believed that *Underwater Wireless Sensor Networks* (UWSNs) will enable innovative research, ushering in a new economy for the sustainable exploitation of our oceans and waterways. Widespread use of UWSNs, however, has been considerably slowed down by the very specific challenges of the environment where they operate. These challenges include long propagation delays, extremely low data rates (few hundreds of bits), unpredictable and highly variable channel conditions, high occurrence of asymmetric links, long interference ranges and underwater specific interference from external noise [15]. As a consequence, networking solutions developed for terrestrial wireless networks are rendered practically unusable for underwater networking. This is because these solutions are highly affected by the quality of the underwater link, whose variations are less predictable and manageable than that of wireless radio links, and by the difficult accessibility of the network nodes, often deployed at considerable depth.

Perhaps, the greatest challenge to the deployment of UWSNs is providing wireless sensor nodes with the energy needed to perform their operations. This is particularly more challenging than for wireless terrestrial networks, as transmitting data in water requires order of magnitude more power than in air, namely, from the milliwatts (mW) of prevailing mote technologies [22], to the tens of watts needed to deliver data to underwater recipients [10]. Underwater wireless nodes are commonly powered through batteries. For instance, the Teledyne Benthos acoustic Smart Modem comes with its own battery, which can be rechargeable (SM 976) or not (SM 975) [30]. Regular batteries last only a few weeks, even if the node activity is very low. Rechargeable batteries also have limited lifetime, in that they need to be replaced after a few recharging cycles [27]. Whether rechargeable or not, substituting or recharging batteries is a costly operation, which requires manned intervention: Devices need to be retrieved from their underwater locations and then put back. This constitutes a severe limitation to the realization of UWSNs, especially if one considers that for many critical applications (e.g., offshore monitoring of oil plants) node accessibility can be prevented by adverse environmental conditions. Attempts to

alleviate this problem have focused on designing devices and solutions aiming at "greenifying" applications or the protocol stack [8]. However, improvements to battery life appear to be unremarkable, and certainly not enough for the requirements of most applications.

Recent efforts to provide power to untethered underwater nodes have focused on endowing them with systems for *harvesting ambient energy*. Best source candidates are solar energy (for nodes deployed close to sea surface), the kinetic energy of underwater currents, and the energy that can be drawn from vibrations. This is because for these sources harvesters can be designed with the necessary small form factor that allows them to be deployed effectively with the node. For instance, piezoelectric methods for ocean energy harvesting achieve powers that ranges from milliwatts to several watts depending on the size of the harvester [18]; miniaturized sea-surface turbines have been designed that are able to provide up to 700mW, which are suitable to extend terrestrial networks over the water, and thermally powered Autonomous Underwater Vehicle are built that use phase-change materials to produce energy from temperature changes (around 1.5mW per 10 Kelvins of temperature difference) [5]. Recently, in Cario et al. [7] we presented platforms for powering underwater acoustic modems via solar panels floating on buoys and via a sea current turbine to be deployed with nodes at greater depths. In particular, the sea turbine is capable to generate power up to 4W at about 42 rpm, achievable with current speeds of about 1 knot. These works clearly show that it is possible to recharge the batteries of underwater devices without having to retrieve them from their location.

This paper constitutes the first attempt at taking advantage of these new upcoming technologies to demonstrate the feasibility of *energy harvesting-enabled UWSNs* (*EH UWSNs*). We show how by smartly combining knowledge on available energy *and* on the energy harvestable in the near future nodes can dynamically route data to their final destination (the network *sink*). This obtains significant performance gains over what is possible in networks only powered by batteries. Our contributions are summarized as follows.

(1) We define the first energy harvesting-aware data forwarding method for UWSNs, named *HyDRO* for *Harvesting-aware Data ROuting*. Using reinforcement learning-based knowledge, senders are able to select the best forwarding relay for their data packets considering a judicious combination of key node and network parameters. Particularly, every time a node has a data packet to forward, a relay is selected based on available residual energy, on foreseeable harvestable energy, on channel quality and on a measure of energy availability through the whole route to the sink provided by neighboring nodes, thus addressing network wide performance via local information exchange. We notice that while machine learning-based techniques have been extensively used for terrestrial wireless sensor networks solutions [1, 12, 32], this is the first application of these techniques to the definition and optimization of energy harvesting usage in UWSNs.
(2) We model multi-modal harvesting in UWSNs. Particularly, we model the solar panels and the turbine designed for our experiments for fish farms [7], and provide them as input energy sources based on real traces from ocean coastal settings [13, 21]. We implemented our models in SUNSET, a simulator that represents underwater environments realistically, supporting different channel models and detailed representation of communication component and node energy expenditure [23]. The flexibility of SUNSET permits us to seamlessly integrate the models we provide for the solar panels and sea current turbine, thus enabling experiments in heterogeneous scenarios where nodes that are deployed closer to the surface recharge their batteries through solar panels and nodes that are positioned at greater depth draw their energy from sea currents (Section 3).
(3) We test the performance of HyDRO through simulations with SUNSET. Comparison among HyDRO and state-of-the-art protocols for underwater data forwarding, namely CARP [4] and QELAR [16], show that HyDRO outperforms them in all settings: Energy consumption is inferior, nodes are kept operational considerably longer than by CARP and QELAR, data latency is kept to a few seconds (in the single digits!) and even in the most challenging scenarios—networks with 40 nodes and high traffic—HyDRO obtains a PDR that improves over that of CARP and QELAR by several tens of percentage points. Our protocol also achieves remarkable fairness, delivering packets from all nodes in the networks, even those further from the sink, which are usually penalized by longer routes and higher interference. The results that better advocate for EH UWSNs concern the comparison of running HyDRO also on UWSNs, namely, on networks without harvesters. In these networks, when a node "dies" for battery depletion it is for good. We observe that in EH UWSNs HyDRO obtains a PDR that is 57% higher than in UWSNs.

Overall, our investigation shows that by delivering more packets and by keeping nodes operational for longer times, energy harvesting and energy harvesting-aware smart routing protocols are necessary components of UWSNs. Perhaps, they are the only components capable of enabling a vast variety of critical underwater applications.

The rest of the paper is organized as follows. In Section 2 we survey previous works on underwater energy harvesting and routing. Section 3 introduces the networking scenario considered in the paper. Section 4 defines HyDRO in details. Section 5 reports results from our comparative performance evaluation of HyDRO, CARP, and QELAR. Finally, Section 6 concludes the paper.

2 RELATED WORK

The increasing interest for underwater wireless networking has brought the problem of how to supply power to the network nodes to center stage. Due to the high costs of replacing or recharging batteries, especially for the scarce accessibility to the nodes, ambient energy harvesting is becoming the premier solution to make UWSNs viable. In case of network nodes deployed in shallow waters, solutions using solar panels deployed on floating buoys have been proposed [7, 28]. Particularly, in Cario et al. [7] we describe the usage of this technology along with the design and testing of a sea current-based turbine designed explicitly to power an underwater sensor node. The turbine is able to generate 4W of power from a current of 1 knot. (These are the harvesting systems we have

considered in this paper.) Jbaily and Yeung survey the use of piezoelectric devices for ocean energy [18]. Although the idea of using the piezoelectric effect as a power take-off mechanism for ocean energy has been investigated for decades, only recently harvesting devices have been developed that produce power at a scale that suitable for EH UWSN nodes (from mWatts to Watts). None of the works introducing these technologies concerns the definition of data delivery strategies designed to take advantage of harvested energy. *In fact, to the best of our knowledge, there is no work to date presenting energy harvesting-aware routing for UWSNs, as it has instead been done extensively for terrestrial wireless networks [3, 25, 34].* Research on routing for UWSNs has mostly been concerned with energy efficiency (to maximize the savings of the energy available to the nodes), or performance. In this realm, the list of data delivery solutions is hefty, including the works of [9, 20, 26, 31] and those surveyed by [19].

Two solutions are worth mentioning in more details, as they provide suitable benchmarks for HyDRO, namely, CARP and QELAR. These two solutions have been shown to outperform previous underwater routing and machine learning-based routing protocols, respectively. The Channel-aware Routing Protocol (CARP) by Basagni et al. exploits link quality information and node residual energy for data forwarding, providing robust and reliable networking as never seen before for UWSNs [4]. Nodes are selected as relays based on their available energy and on the quality of the links to their neighbors judiciously monitored over time. CARP utilizes a channel reservation mechanism for channel access and for selecting packet relays, enabling cross layer design for enhanced performance. However, for this reason, while achieving reliability and suffering from few packet collisions, it incurs remarkable latency, especially in large networks. Among those protocols that make smart routing selection by using reinforcement learning-like frameworks, the QELAR solution by Hu and Fei is particularly interesting [16]. Following a Q-learning-based approach, the QELAR reward function is defined to maximize the residual energy among nodes, accounting for the residual energy of each node as well as for the energy distribution among neighboring nodes. Eventually, relays are chosen depending on the energy they can save. This makes QELAR a solution that compares well with previous protocols, especially in terms of network lifetime.

We observe that other solutions for routing that are based on learning techniques have been presented for underwater networking [2, 17, 24]. However, the scenarios considered in these works are quite different from those considered in this paper. Particularly, the MARLIN protocol by Basagni et al. concerns networks with multiple physical layers (multi-modality), and presents learned ways for choosing not just the best relay, but also the best modem to reach that relay [2]. The solutions proposed by Plate and Yakayama [24] and by Hu and Fei [17] are instead concerned with disruption tolerant networks. We observe that none of these solutions consider underwater networks with energy harvesting.

3 ENERGY HARVESTING-ENABLED UWSNS

Our investigation concerns *energy harvesting-enabled underwater wireless sensor networks* (EH UWSNs) comprising nodes that are statically placed at different depths in the ocean. Nodes are equipped with sensors that produce information in need to be routed to the network *sink*, namely, to a node providing functionalities for data storage, data processing, and as a gateway to terrestrial networks. The route from a node to the sink may be multi-hop, i.e., data packets may need to transit through multiple other nodes. For communication purposes each node is equipped with a half-duplex and omnidirectional acoustic modem, such as the Teledyne Benthos SM-975 [30], or the Evologics S2CR 48/78 [10]. To support their operations nodes harvest energy from the environment and store it in a rechargeable battery, like those usually provided with the acoustic modems. Harvesting for nodes deployed at sea bottom or at relevant depths happens via turbines drawing energy from sea currents. Nodes deployed closer to sea surface harvest energy via solar panels deployed on floating devices (e.g., moored buoys) cabled to the nodes. We recently tested a similar working system using both solar and current energy harvesting, with circuitry to provide energy to rechargeable batteries [7]. There might be times when a node has not enough energy in its battery to support its operations (e.g., sensing, computation, communications, etc.). When this is the case, the node turns off all its circuitry, and it is called an *all-off node*. It will restart its functions as soon as enough new energy has been harvested. A typical scenario is depicted in Figure 1. The sink is depicted as the node with additional wireless radio capabilities (upper right side of the picture).

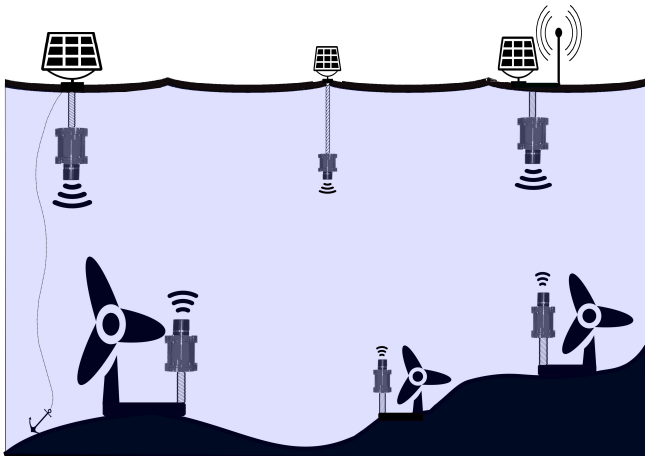

Figure 1: An energy harvesting-enabled UWSN (EH UWSN).

The management of all-off nodes is performed through timers, and through a simple mechanism used by nodes to proactively signal that their battery levels are crossing a critical threshold. Particularly, if a node i does not hear packets from node j for a given time t, it removes node j from the list of its neighbors until node j is heard again (through direct communication or promiscuously, by overhearing transmissions from it). Right before running out of energy, a node j notifies its neighbors about it by setting a field in its data packet headers. Upon receiving or overhearing packets with such a set field, node i temporarily removes node j from the list of its active neighbors.[1]

[1] We notice that the combination of timers and direct signaling could be also used to manage the presence of mobile nodes, which show dynamics similar to those of

4 THE HYDRO PROTOCOL

This section presents HyDRO in details. Specifically, we provide a description of the basic operation of choosing a relay for packet forwarding, and we define the framework through which HyDRO learns how to route.

The operations for transmitting a packet p are implemented by the following algorithm TransmitPacket(p).

TransmitPacket(p)

```
1:  if there are known neighbors then
2:      k = 0
3:      while k < K do
4:          j = SelectRelay(k)
5:          forward packet p to j
6:          if (j is heard to transmit p within τ time units) then
7:              break
8:          end if
9:          k = k + 1
10:     end while
11:     discard packet
12: else
13:     broadcast p
14: end if
```

In general, when a node i has a packet p to forward, it chooses the most suitable relay j among its neighbors and transmits p to that relay (lines 4 and 5). After transmission, node i awaits to overhear the forwarding of p by relay j (*implicit ACK*;[2] line 6). If that happens within a pre-defined time τ, the transmission operation terminates successfully (line 7). Otherwise, node i retransmits p till success, for at most $K-1$ additional times, $K \geq 1$ (**while** loop). If all retransmission attempts fail, the packet is dropped (line 11). (Parameters K and τ are global to the algorithm. They are set at the start of node operations.)

This description of the protocol assumes that node i has known neighbors, i.e., that it is always able to select a neighbor j as a suitable relay (line 4). In case node i knows of no neighboring nodes, packet p is broadcast in the failsafe attempt that some node may receive and forward it (lines 1 and 13). This latter situation arises at the very start of network operations, and every time a node comes back from being all-off or from some temporary malfunction.

The novel crux of our protocol resides in the algorithm SelectRelay, executed by node i to select the next hop relay j (line 4). This operation performs a learned choice of a relay, which may even change at every retransmission attempt k, keeping into account crucial parameters such as a potential relay residual energy and the foreseeable availability of some harvestable energy. The algorithm is driven by a reinforcement learning framework allowing node i to learn from its current environment, namely, the available and predicted energy at its neighbors, and their recent goodness in forwarding packets (a measure of link quality). In the remainder of this section we define the learning framework. (For details on reinforcement learning the reader is referred to the extensive literature on this subject [29].) We conclude the section by describing how, using this framework, procedure SelectRelay determines an optimal forwarding decision, namely, the most suitable relay j for node i.

A node i handling a packet p is in a *state* that indicates the number of times that the packet has been transmitted unsuccessfully. More formally, the state space $\mathcal{S}$ of a node i handling a packet p is the set $\{0, \ldots, K-1\} \cup \{rcv, drop\}$. The set $\mathcal{S}$ contains the number of times $k \leq K-1$ that node i has transmitted p unsuccessfully, successful packet transmissions (rcv), and the case when the maximum number K of transmissions has been exceeded and the packet is discarded ($drop$).

Node i makes forwarding decisions depending on the set of possible *actions* it can take from a state s. This set of actions is defined as follows:

$$A_i(s) = \{a = j \mid \ j \in N(i)\}, \tag{1}$$

where $a = j$ is the action of forwarding a packet to neighbor j, and $N(i)$ is the set of node i current neighbors. Since no action can happen when $s \in \{rcv, drop\}$, it is $A_i(rcv) = A_i(drop) = \emptyset$.[3]

Transitions between successive states s and s' depend on state s and on the action $a = j$ performed by node i. *Transition probabilities* $P^a_{i,s\to s'}$ depend on whether the transmission of packet p is successful or not. In the first case, if the success happens after k unsuccessful attempts, node i transitions from state $s = k$ to state $s' = rcv$ with the following probability:

$$P^a_{i,s\to rcv} = \begin{cases} P_{i,j}P_{j,i} & \text{if } 0 \leq k < K-1 \\ P_{i,j} & \text{if } k = K-1, \end{cases} \tag{2}$$

where $P_{i,j}$ denotes the probability of correct packet transmission on the link between nodes i and j. Particularly, successful transmission depends on the following probabilities: (a) The probability $P_{i,j}$ that the packet is received by node j. This probability is computed by node j and broadcast in the header of its packets. (b) The probability $P_{j,i}$ that the packet p, forwarded by node j, is successfully overheard by node i (implicit ACK). This probability is computed by node i based on overhearing node j transmissions. (The determination of these probabilities is described in details below; Equation (7)). When the packet p has been retransmitted the maximum number of times K, node i does not need to overhear an ACK, and the transition depends only on the goodness of the link from node i to node j. Note that transition probabilities depend on the action $a = j$, i.e., they are different for different neighbors j.

If the transmission of p fails, we have two possible transitions. If $k < K-1$ the next state is clearly $s' = k+1$. Otherwise, if $k = K-1$, the packet p is dropped and the next state is $s' = drop$. In both cases, the transition probability is given by $P^a_{i,s\to s'} = 1 - P^a_{i,s\to rcv}$.

HyDRO routes packets with the goal of maximizing the residual energy on the *whole route* towards the sink. To model route-long residual energy each state-action pair (s, a) is associated with a *reward function* r_i reflective of the residual energy of the sender

nodes going all-off and back over time. We intend to explore the extension of HyDRO to networks with mobile nodes in the future.

[2] Each packet is acknowledged *implicitly*. Particularly, after transmitting a packet, the sender starts listening to the channel. If it overhears the packet being retransmitted by the chosen relay within time τ, it considers the packet transmitted successfully. If it does not, the packet is considered lost. (Only the sink sends explicit ACKs back to its senders, as it does not forward the packet further.)

[3] Note that we are concerned with single packet forwarding. A state is an abstraction to model it. In this sense, if a node has no packet to transmit, it has no state.

node i and of the residual energy on the route from the selected relay to the sink. More formally:

$$r_i(s,a) = \begin{cases} e_i(s,a) + n_i(s,a) & \text{if } 0 \le k < K-1 \\ e_i(s,a) + n_i(s,a) - l_i(s,a) & \text{if } k = K-1. \end{cases} \quad (3)$$

In the following, we describe each single component of the reward $r_i(s,a)$. Component $e_i(s,a)$ is defined as follows:

$$e_i(s,a) = \begin{cases} b_i + h_i - e_{tx} & k = 0 \\ -e_{tx} & \text{otherwise,} \end{cases} \quad (4)$$

where b_i is node i residual energy, and h_i is the energy the node will harvest in the immediate future, which can be obtained by using some form of ambient energy prediction [6]. With e_{tx} we indicate the energy spent to transmit the packet. The first transmission of the packet considers both residual and harvestable energy. Successive transmissions lower the reward only by the amount of energy needed to transmit the packet. Component $n_i(s,a)$ indicates the residual energy on the path from the next hop relay j to the sink. Particularly:

$$n_i(s,a) = V_j P_{i,j}, \quad (5)$$

where V_j provides a measure of the residual energy on the path towards the sink starting from node j. (It is available to node i as it is broadcast by node j in the header of its packets.) The cost V_j is multiplied by the probability $P_{i,j}$ as node j will receive the packet only in case it is correctly transmitted from node i.

Finally, in case a packet has been unsuccessfully retransmitted for $K-1$ times, we associate to the action $a = j$ of the last retransmission the energy penalty $l_i(s,a) > 0$. This penalty aims at discouraging node i to drop the packet, i.e., transitions to the *drop* state. As such, $l_i(s,a)$ is defined as:

$$l_i(s,a) = L(1 - P_{i,j}), \quad (6)$$

where $(1 - P_{i,j})$ is the probability of dropping the packet, and L is set to an arbitrarily high value. In other words, as node i reaches the maximum number K of retransmissions, its actions favor the reliability of forwarding to the next hop.

We are finally able to describe how a node i that has a packet p to transmit learns how to route by using it, i.e., its optimal forwarding policy. Each node starts with no knowledge of its surrounding environment. Interacting with its neighbors, it iteratively acquires and updates its knowledge over time. A *value function* V_i is approximated and updated relying on current estimations of the transition probabilities $P^a_{i,s\to s'}$, and on the estimated value of the functions V_j from neighboring nodes j, needed to estimate the reward $r_i(s,a)$ (Equations (3) and (5)). (As such, all values of the transition probabilities, and of functions V and r are intended to be estimates changing over time.) Algorithm SelectRelay describes the learning process of node i and the corresponding determination of the best relay for packet p.

SelectRelay(k)

1: **for all** ($s \in \mathcal{S}$) **do**
2: **for all** $a \in A_i(s)$ **do**
3: $Q_i(s,a) = r_i(s,a) + \gamma \sum_{s' \in \mathcal{S}} P^a_{i,s\to s'} V_i(s')$
4: **end for**
5: $V_i(s) = \max_{a \in A_i(s)} Q_i(s,a)$ *#Update*
6: **end for**
7: $j = \arg\max_{j \in A_i(k)} Q_i(k,a)$ *#Forwarding decision*
8: **return** j

The algorithm takes as input the current state k. When packet p is ready for transmission, node i computes the new value function based on the most recent information received from its neighbors (lines 1 to 6). Once the value function has been updated, the best forwarding action, namely, a relay, can be selected (line 7).

The execution of algorithm SelectRelay relies on the knowledge of the following.

- *The transition probabilities* $P^a_{i,s\to s'}$. The estimation of the transition probabilities is based on the estimation of the link probabilities $P_{i,j}$ (Equation (2)). Nodes estimate link quality upon receiving a packet. In particular, a node j keeps count of the number of packets $n_{i,j}$ received from each neighbor i, even if node j is not the packet intended destination. The incoming link probability is then estimated as
$$P_{i,j} = \frac{n_{i,j}}{n_i}, \quad (7)$$
where n_i is the total number of packets sent by node i, an information that node i broadcasts in the header of its packets. These estimates are then broadcast by node j into its packet headers, to be overheard by its neighbors. To record varying link conditions the counts n_i and $n_{i,j}$ are computed over a sliding window. If node i fails to overhear transmissions from a neighbor j within a given time it "degrades" $P_{i,j}$ to $\frac{n_i}{(n_i+1)} P_{i,j}$.
- *The packet forwarding reward* $r_i(s,a)$. According to Equation (3) computing $r_i(s,a)$ requires access to information local to node i (including b_i, h_i, e_{tx}, L and $P_{i,j}$), and to the value function V_j of each if its neighbors j, which is broadcast by node j in the header of its packets.
- *The discount factor* γ, $0 \le \gamma \le 1$, which is used to provide a way of deciding the importance of future costs.

5 PERFORMANCE EVALUATION

We evaluate the performance of HyDRO via simulations in scenarios modeled to include realistic underwater channel conditions as well as energy harvesting based on recorded traces. The performance of our protocol is also compared to that of two solutions that represent the state-of-the-art in routing for UWSNs in scenarios like those considered here. The two protocols are: (i) CARP, a cross-layer solution using channel reservation via control packets (named PING and PONG) that carry routing information [4], and (ii) QELAR, a machine learning-based protocol designed for minimizing and balancing node energy consumption [16]. (Details can be found in Section 2 and in the original papers [4, 16].)

All routing protocols have been implemented in SUNSET [23] connected to the Bellhop ray tracing tool via the WOSS interface [14]. The input data to Bellhop are those from an area located in the Norwegian coast close to Trondheim, with the coordinate $(0, 0, 0)$ of the surface located at $63°, 29', 1.0752''N$ and $10°, 32', 46.6728''E$. We use well-known databases for sound speed profiles, bathymetry profiles and information on the type of bottom sediments of the selected area, namely, the World Ocean Database, the General Bathymetric Chart of the Oceans (GEBCO), and the National Geophysical Data Center's Deck41 data-base, respectively [33].

5.1 Energy harvesting traces and model

We consider nodes equipped with one harvesting device that is either a solar panel or a sea current turbine. We described both types of harvesting in Cario et al. for a UWSN deployment [7]. The solar panel is deployed on buoys at sea surface and cabled to the node. Its maximum output power is 20W. Nodes deployed at depths greater than 50m draw power from a sea current turbine. This device is capable to generate a power of about 4W from a current speed of about 1 knot [7]. Its orientation is fixed, i.e., it does not change with the direction of the current.

In our simulations we consider actual traces for both sun and sea currents. Traces concerning the sun are obtained from the National Renewable Laboratory at Oak Ridge [21]. Current traces are obtained from the Global Ocean Currents Database (GOCD) [13]. Particularly, we selected sea currents whose speed does not exceed 1 knot, which fits the design of the turbine we modeled [7]. Since the direction of the sea current changes over time, while each turbine is fixed, we approximate the speed of the current over each turbine taking into account the angle between the current itself and the turbine orientation.

5.2 Simulation scenarios and metrics

We investigate the selected protocols in networks of different size. Particularly, we consider networks with 20 nodes placed in a rectangular region with a surface of $2km^2$, and with 40 nodes deployed in a rectangular region with a surface of $4km^2$. Nodes are deployed in the areas randomly and uniformly at depths ranging from 10 to 240m. The sink is located at one of the corners of the deployment area, 10m below surface. We stipulate that the sink is always on, i.e., that it is continuously powered (e.g., by replaceable batteries). Nodes positioned at depth up to 50m harvest energy using solar cells. All remaining nodes use turbines. We consider only topologies where nodes are deployed so that there is always at least one route to the sink from any node.

Nodes communicate through an acoustic modem whose carrier frequency is set to 25.6kHz for a bandwidth of 4kHz, resulting in a bit rate of 4000b/s. For these selected values of bandwidth and carrier frequency the transmission and reception power of the modem are set to 8.5W and 0.5W, respectively. These values are consistent with those of the commercial modems by Teledyne Benthos [30] and Evologics [10]. Each node is equipped with a rechargeable battery. Initially, the batteries of all nodes are full, and contain 80kJ of energy.

Traffic is generated according to a Poisson process with a network-wide inter-arrival time averaging at $\{100, 50, 20\}$ seconds, corresponding to low, medium and high traffic, respectively. Once a packet is generated, it is associated with a source selected randomly and uniformly among all nodes except the sink, which is the destination of all packets. The data packet payload size is 1000B. The total size of a data packet is given by the payload plus the headers added by the different layers and protocols. The physical header overhead changes according to the data rate but is dominated by a 10ms synchronization preamble. At the MAC layer, the header size depends on the protocol. QELAR uses CSMA, whose header contains the sender and destination addresses and packet type, for which 3B are enough. This protocol also needs 6 extra bytes for information on the state space of nodes (e.g., the residual energy). CARP implements its own MAC. Because of its cross layer design the header of its MAC packets also carries routing information. Therefore, the size of its PING and PONG control packets is 10B and 6B, respectively. Its ACK and HELLO packets are 6B long. The CARP MAC data packet header is 4B long. CARP uses different transmission power levels for control and data packets. In our experiments the transmission power for control packet is set to 5.2W. Finally, HyDRO carries information on the value function, $P_{i,j}$ estimates from neighbors, and other control information in the packet header. As a consequence, its size varies with the network size. In our implementation HyDRO header sizes were 15B and 30B, for networks with 20 and 40 nodes, respectively. Through varying experiments we determined the value of the number of retransmissions K used by HyDRO, QELAR and CARP that produces best performance. Particularly, K is set to to 5 for low traffic, 4 for medium traffic, and to 3 for high traffic. We use an exponential moving average filter as energy harvesting predictor for the HyDRO reward function.

Neighbor discovery and maintenance is performed as follows. Each node i maintains a list of neighbors $N(i)$. A node j is in the list $N(i)$ if node i has received a packet from node j or has overhead node j transmitting a packet. If node i has not received from node j or has not overheard a packet transmitted by node j for $t = 900$s, it removes node j from $N(i)$ as j could be malfunctioning or all off. (See also Section 3.)

Simulation parameters are summarized in Table 1, which also shows the values chosen for other HyDRO-specific parameters.

The performance of the considered protocols is evaluated by investigating the following metrics.

(1) *Energy consumption*, defined as the energy consumed by the whole network.
(2) *All-off time*, namely, the fraction of simulation time (percentage) when a node is off for lack of energy.
(3) *End-to-end latency*, i.e., the time between packet generation and that of its correct delivery to the sink.
(4) *Packet delivery ratio* (PDR), defined as the ratio of packets correctly received by the sink and the number of all generated packets.

All results are obtained by averaging over data from a number of simulation runs that allow us to achieve a statistical confidence of 95% within a 5% precision.

Table 1: Simulation parameters.

Parameter	Value
Simulation duration	6 days
Number of nodes	[20, 40]
Size of the deployment area	[2km^2,4km^2]
Depths of deployment	From 10m to 240m
Bit rate	4000b/s
Modem Tx power	8.5W
Modem Rx power	0.5W
Modem Tx power (CARP, ctl pkt)	5.2W
Battery capacity	80kJ
Packet payload size	1000B
HyDRO header size	[15, 30]B
CARP header size	4B
QELAR header size	6B
Packet inter-arrival time	[100, 50, 20]s (low to high traffic)
Number of retransmissions K	[5, 4, 3] (low to high traffic)
Time t to remove node from neighbor list	900s
Time τ to wait for an ACK	Traffic and topology dependent
Discount factor γ	0.95

5.3 Performance results

We show results for network scenarios with 40 nodes. (Results for smaller networks show similar trends and provide no further insights.)

5.3.1 Energy consumption. Figure 2 shows the average energy consumed as traffic increases.

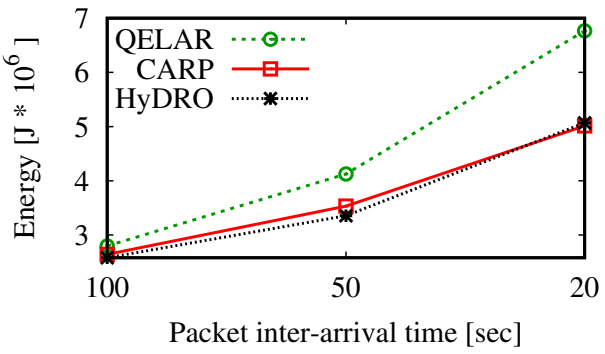

Figure 2: Energy consumption.

Clearly, all protocols consume more at higher traffic because of the higher number of data bits to carry. HyDRO and CARP show the best performance. The energy consumed by CARP is naturally higher than that of HyDRO because of CARP channel reservation handshake. However, as the PING and PONG packets are quite small, and as the power used to transmit them is smaller than that used for data packets, the increase in energy consumption is contained, being at most 5% higher than that of HyDRO. As transmission reliability is not explicitly factored into its reward, QELAR incurs a higher number of retransmissions, especially at higher traffic, and therefore its energy consumption is 33% higher than that of HyDRO. We observe that the lower energy consumption of HyDRO also depends on the inclusion in its reward function of a penalty for dropping packets. This higher reliability implies less retransmissions and therefore superior energy savings.

5.3.2 All-off time. Figure 3 shows the average all-off time for increasing traffic. HyDRO shows the best performance. Despite

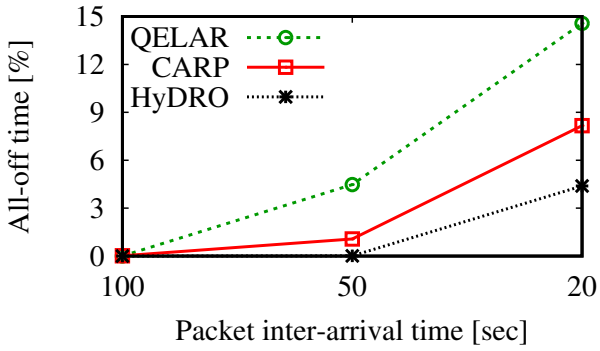

Figure 3: All-off time.

the energy consumption of HyDRO and CARP is about the same (see above), nodes running HyDRO are all-off 47% less than those running CARP. This is because HyDRO makes routing choices that are based also on predicting which nodes will have higher availability of energy in the near future. This allows HyDRO a judicious use of available energy. Conversely, the performance of QELAR is consistent with its higher energy consumption, which imposes higher number of nodes that are all-off for longer times.

To gain further insights on the protocol management of available energy we investigated the time till the first node goes all-off. We observe that, on average, when the network runs HyDRO the first node goes all-off 43% time later than CARP, and 108% later than QELAR. This provides further evidence that choosing relays based on the smart combination of residual energy and predicted harvested energy keeps nodes operational consistently longer than other solutions.

5.3.3 End-to-end latency. Figure 4 shows the average end-to-end latency for increasing traffic.

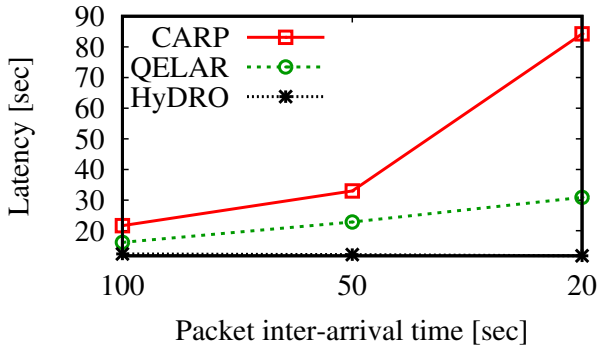

Figure 4: End-to-end latency.

Not surprisingly, the latency performance increases with traffic, because of retransmissions due to interference and longer routes

due to a higher number of all-off nodes. HyDRO is successful in providing swift routes to the sink. Compared to that of CARP, its latency is up to 85% lower, which is mostly because HyDRO does not have to pay the toll of the lengthy CARP handshakes. The same holds for QELAR, which also routes packets to the sink considerably faster than CARP. The latency performance of HyDRO is up to 66% lower than that of QELAR. This is due to HyDRO reward function, which considers energy prediction and encodes a hefty penalty for dropping. This allows nodes running HyDRO to choose more reliable relays and to route packets along shorter routes.

5.3.4 Packet delivery ratio. Figure 5 shows the average packet delivery ratio for increasing traffic.

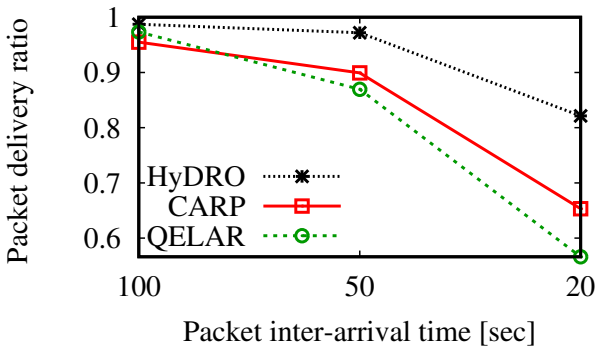

Figure 5: Packet delivery ratio.

The PDR of all protocols decreases with increasing traffic. This is because of the increased number of packet collisions and also because the higher number of all-off nodes. HyDRO shows consistently better performance than CARP and QELAR, with performance gaps increasing with traffic, which indicates higher scalability. Specifically, its PDR is 26% higher than that of CARP, and 45% than that of QELAR (high traffic). The reason is twofold: (i) in HyDRO nodes are all-off for less time (Figure 3), and (ii) by transmitting packets on shorter and more reliable routes, and without the need of a channel reservation handshake, HyDRO reduces the offered network load, inducing a noticeable lower number of packet collisions.

We provide further evidence of the superior effectiveness of HyDRO in managing residual and predicted energy by showing joint snapshots of packet delivery ratio and all-off time per node. Figure 6 concerns the topology of a network with 40 nodes and high traffic running HyDRO, CARP and QELAR (vertical section view).

The sink is the black square at the top left corner. Nodes that harvest energy using solar panels are depicted as triangles, while circles correspond to nodes that use sea current as their harvesting source. The size of each node is proportional to the all-off time (the smaller the better), while the color indicates the PDR of that node (the darker the color, the higher the PDR). Figure 6a shows that HyDRO is inherently fair to all node. The picture also clearly confirms that the reason some nodes have lower PDR depends on the fact that those nodes are all-off, on average, more than other

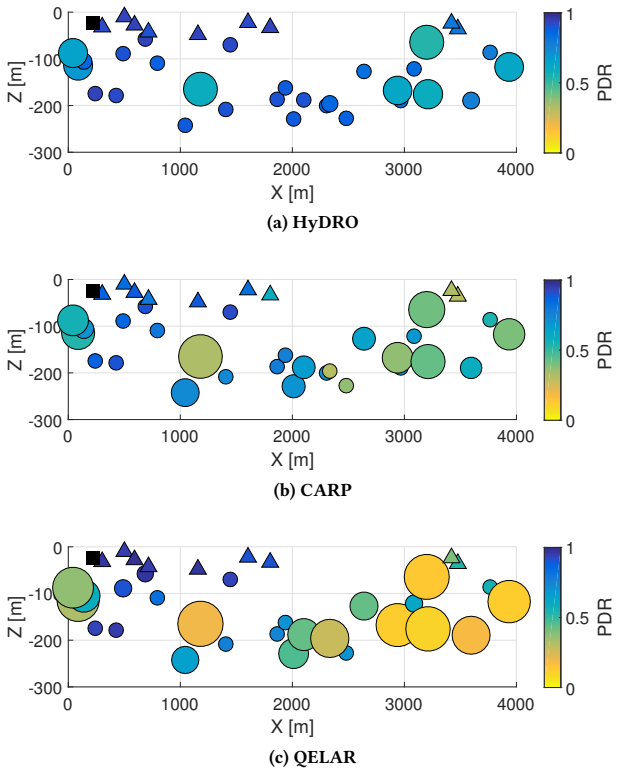

Figure 6: A vertical view of a network topology with 40 nodes and high traffic: Joint PDR and all-off time.

nodes. In fact, these nodes are all-off not because of protocol operations, but because of the actual traces we used: Those nodes are positioned at locations where the current is not particularly strong. Nonetheless, HyDRO makes the best out of this harvested energy, and even nodes with lower PDR are able to deliver considerably more packets than those delivered by CARP and QELAR, whose color is definitely lighter (Figure 6b and Figure 6c). The performance of CARP (Figure 6b) is reflective of its lack of considering incoming future energy. For this reason, nodes are all-off for longer times, which explains why the PDR is lower throughout the network. QELAR shows a marked unfairness to the nodes that are farthest from the sink (Figure 6c). This is because its reward function does not impose any penalty for dropped packets, which has a higher impact on nodes needing longer routes to the sink. We also observe that nodes that have the highest all-off times obtain abysmal PDR (as low as 10%), suggesting that, differently from HyDRO, QELAR cannot consider energy harvestable in the near future.

We conclude this section by providing results that make an even stronger case for EH UWSNs. We ran our simulation experiments in scenarios that are the same of those considered so far except that the nodes do not have energy harvesting capabilities. We observed that in EH UWSNs HyDRO obtains a PDR that is 57% higher than that obtained in UWSN. Furthermore, in scenarios with harvesting, the time when the first node goes all-off happens 48% later than when the first node dies (for energy depletion) in UWSNs.

6 CONCLUSIONS

This paper concerns EH UWSNs, namely UWSNs where nodes have energy harvesting capabilities. In this setting, we present HyDRO, the first routing protocol that explicitly takes into account harvested energy for selecting next hop relays. HyDRO rests on a reinforcement learning-based framework that instructs senders to select the best forwarding relay for their data depending on a well-balanced combination of residual energy, of energy that can be harvested in the foreseeable future, and on the forwarding quality of the channel to neighboring nodes. Through a SUNSET-based performance investigation we show that HyDRO always outperforms previous state-of-the-art forwarding protocols: It consumes less energy, it keeps nodes operational for noticeably longer times, and provides superior PDR while keeping end-to-end latency at bay. HyDRO is also fair throughout the network, allowing all nodes to deliver high PDR, even those that are farther from the sink. By running HyDRO on networks without harvesters we observe a remarkable decrease in performance than in EH UWSNs. This result confirms that the smart exploitation of energy harvesting via suitably designed protocols constitutes the key to unlock superior performance and to enable a wider variety of underwater applications.

ACKNOWLEDGMENTS

The work was performed under the sponsorship of the EC EASME ArcheoSub project "Autonomous underwater Robotic and sensing systems for Cultural Heritage discovery Conservation and in situ valorization." (Grant Agreement n. EASME/EMFF/2016/1.2.1.4/01/SI2.749264.) Stefano Basagni was supported in part by grants NSF CNS 1428567 ("MRI: Development of the Northeastern University Marine Observatory NETwork—NU MONET") and NSF CNS 1726512 ("MRI: SEANet: Development of a Software-Defined Networking Testbed for the Internet of Underwater Things"). Part of this work has been performed when Dr. Basagni was visiting the department of Computer Science of the University of Rome "La Sapienza," while on sabbatical leave from Northeastern University. During his stay he was also supported by a grant from the Sapienza Visiting Professor programme (2016).

REFERENCES

[1] M. A. Alsheikh, S. Lin, D. Niyato, and H.-P. Tan. 2014. Machine Learning in Wireless Sensor Networks: Algorithms, Strategies, and Applications. *IEEE Communications Surveys & Tutorials* 16, 4 (April 24 2014), 1996–2018.

[2] S. Basagni, V. Di Valerio, P. Gjanci, and C. Petrioli. 2017. Finding MARLIN: Exploiting Multi-Modal Communications for Reliable and Low-latency Underwater Networking. In *Proceedings of IEEE Infocom 2017*. Atlanta, GA, 1701–1709.

[3] S. Basagni, M. Y. Naderi, C. Petrioli, and D. Spenza. 2013. Wireless Sensor Networks with Energy Harvesting. In *Mobile Ad Hoc Networking: Cutting Edge Directions*, S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic (Eds.). John Wiley & Sons, Inc., Hoboken, NJ, Chapter 20, 703–736.

[4] S. Basagni, C. Petrioli, R. Petroccia, and D. Spaccini. 2015. CARP: A Channel-Aware Routing Protocol for Underwater Acoustic Wireless Networks. *Elsevier Ad Hoc Networks and Physical Communication, joint Special Issue on Advances in Underwater Communications and Networks* 34 (November 27 2015), 92–104.

[5] J. R. Buckle, A. Knox, J. Siviter, and A. Montecucco. 2013. Autonomous Underwater Vehicle Thermoelectric Power Generation. *Journal of Electronic Materials* 42, 7 (July 2013), 2214–2220.

[6] A. Cammarano, C. Petrioli, and D. Spenza. 2016. Online Energy Harvesting Prediction in Environmentally Powered Wireless Sensor Networks. *IEEE Sensors Journal* 16, 17 (2016), 6793–6804.

[7] G. Cario, A. Casavola, P. Gjanci, M. Lupia, C. Petrioli, and D. Spaccini. 2017. Long Lasting Underwater Wireless Sensors Network for Water Quality Monitoring in Fish Farms. In *Proceedings of MTS/IEEE OCEANS 2017*. Aberdeen, Scotland, 1–7.

[8] R. W. L. Coutinho, A. Boukerche, L. F. M. Vieira, and A. A. F. Loureiro. 2016. On the Design of Green Protocols for Underwater Sensor Networks. *IEEE Communications Magazine* 50 (2016), 67–73. Issue 10.

[9] R. Diamant, P. Casari, F. Campagnaro, and M. Zorzi. 2017. Routing in Multi-Modal Underwater Networks: a Throughput-optimal Approach. In *Proceedings of IEEE WCNEE 2017*. Atlanta, GA, 501–506.

[10] Evologics. 2017. Evologics S2CR 18/34: Product Information. (2017). https://evologics.de/files/DataSheets/EvoLogics_S2CR_1834_Product_Information.pdf

[11] E. Felemban, F. K. Shaikh, U. M. Qureshi, A. A. Sheikh, and S. Bin Qaisar. 2015. Underwater Sensor Network Applications: A Comprehensive Survey. *International Journal of Distributed Sensor Networks* 11 (November 1 2015), 72–78. Issue 11.

[12] R. GhasemAghaei, M. A. Rahman, W. Gueaieb, and A. El Saddik. 2007. Ant Colony-Based Reinforcement Learning Algorithm for Routing in Wireless Sensor Networks. In *Proceedings of IEEE IMTC 2007*. Warsaw, Poland, 1–6.

[13] GOCD. 2017. GLobal Ocean Currents Database. (2017). https://www.nodc.noaa.gov/gocd/

[14] F. Guerra, P. Casari, and M. Zorzi. 2009. World Ocean Simulation System (WOSS): A simulation tool for underwater networks with realistic propagation modeling. In *Proceedings of ACM WUWNet 2009*. Berkeley, CA, 1–8.

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

[16] T. Hu and Y. Fei. 2010. QELAR: A Machine-learning-based Adaptive Routing Protocol for Energy-efficient and Lifetime-extended Underwater Sensor Networks. *IEEE Transactions on Mobile Computing* 9, 6 (June 2010), 796–809.

[17] T. Hu and Y. Fei. 2013. An adaptive routing protocol based on connectivity prediction for underwater disruption tolerant networks. In *Proceedings of IEEE Globecom 2013*. Atlanta, GA, 65–71.

[18] A. Jbaily and R. W. Yeung. 2015. Piezoelectric devices for ocean energy: A brief survey. *Journal of Ocean Engineering and Marine Energy* 1, 1 (February 2015), 101–118.

[19] N. Li, J.-F. MartÃŋnez, J. M. Meneses Chaus, and M. Eckert. 2016. A Survey on Underwater Acoustic Sensor Network Routing Protocols. *Sensors* 16, 3 (March 22 2016), 1–28.

[20] Y. Noh, U. Lee, P. Wang, B. S. C. Choi, and M. Gerla. 2013. VAPR: Void-Aware Pressure Routing for Underwater Sensor Networks. *IEEE Transactions on Mobile Computing* 12, 5 (2013), 895–908.

[21] NREL. 2017. Measurement and Instrumentation Data Center. (2017). http://www.nrel.gov/midc

[22] M. Paoli, D. Spenza, C. Petrioli, M. Magno, and L. Benini. 2016. MagoNode++: A Wake-up Radio-enabled Wireless Sensor Mote for Energy-neutral Applications. In *Proceedings of ACM/IEEE IPSN 2016*. Vienna, Austria, 1–2.

[23] C. Petrioli, R. Petroccia, J. R. Potter, and D. Spaccini. 2015. The SUNSET framework for simulation, emulation and at-sea testing of underwater wireless sensor networks. *Elsevier Ad Hoc Networks* 34, C (November 2015), 224–238.

[24] R. Plate and C. Wakayama. 2015. Utilizing Kinematics and Selective Sweeping in Reinforcement Learning-based Routing Algorithms for Underwater Networks. *Ad Hoc Networks* 34 (2015), 105–120.

[25] C. Renner, S. Unterschütz, V. Turau, and K. Römer. 2014. Perpetual Data Collection with Energy-Harvesting Sensor Networks. *ACM Transactions on Sensors Networks* 11, 1 (September 2014), 12:1–12:45.

[26] D. Shin, D. Hwang, and D. Kim. 2012. DFR: An Efficient Directional Flooding-based Routing Protocol in Underwater Sensor Networks. *Wireless Communications and Mobile Computing* 12, 17 (December 2012), 1517–1527.

[27] F. I. Simjee and P. H. Chou. 2008. Efficient Charging of Supercapacitors for Extended Lifetime of Wireless Sensor Nodes. *IEEE Transactions on Power Electronics* 23, 3 (May 2 2008), 1526–1536.

[28] SUNRISE. 2013-2017. Sensing, monitoring and actuating on the UNderwater world through a federated Research InfraStructure Extending the Future Internet. (2013-2017). http://fp7-sunrise.eu

[29] R. S. Sutton and A. G. Barto. 2017. *Reinforcement Learning: An Introduction* (second ed.). MIT Press, Cambridge, MA.

[30] Teledyne Benthos. 2017. Teledyne Benthos SMART product SM-975: Product Information. (2017). http://teledynebenthos.com/product/smart_products/sm-975

[31] G. Toso, R. Masiero, P. Casari, O. Kebkal, M. Komar, and M. Zorzi. 2012. Field experiments for Dynamic Source Routing: S2C EvoLogics modems run the SUN protocol using the DESERT Underwater Libraries. In *Proceedings of MTS/IEEE OCEANS 2012*. Hampton Roads, VA, 1–10.

[32] P. Wang and T. Wang. 2006. Adaptive Routing for Sensor Networks using Reinforcement Learning. In *Proceedings of IEEE CIT 2006*. Seoul, Korea, 219–219.

[33] WOD, GEBCO, and Deck41. 2017. WOD, GEBCO, and Deck41: Database Description. (2017). http://www.nodc.noaa.gov/OC5/WOA05/pr_woa05.html,http://www.gebco.net,http://www.ngdc.noaa.gov/mgg/geology/deck41.html.

[34] S. Yang, X. Yang, J. A. McCann, T. Zhang, G. Liu, and Z. Liu. 2013. Distributed Networking in Autonomic Solar Powered Wireless Sensor Networks. *IEEE Journal on Selected Areas in Communications* 31, 12 (December 2013), 750–761.