Channel-aware Routing for Underwater Wireless Networks

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," Roma, Italy
{petrioli,petroccia,spaccini}@di.uniroma1.it
[‡] Consorzio Interuniversitario Nazionale per l'Informatica (CINI), Roma, Italy
[§] Nexse software engineering, Roma, Italy

Abstract—This paper presents a new cross layer routing protocol for underwater wireless sensor networks. The solution, termed CARP for Channel-aware Routing Protocol, exploits link quality information for cross layer relay determination. Nodes are selected as relays if they have a (recent) history of successful transmissions to their neighbors. CARP combines link quality with simple topology information (hop count), thus being able to route around connectivity voids and shadow zones. The protocol is also designed to take advantage of power control for selecting robust links. The performance of CARP has been evaluated through ns2-based simulations, and compared to the performance of two previously proposed routing protocols, namely, FBR and DBR. Our results show that CARP robust relay selection mechanism enables it to achieve throughput efficiency that is up to twice the throughput of FBR and almost three times that of DBR. CARP also obtains remarkable performance improvements over FBR and DBR with respect to end-to-end packet latency and energy consumption.

Index Terms—Underwater acoustic networks, cross layer design, MAC and routing protocols.

I. INTRODUCTION

Underwater wireless (acoustic) networking is the enabling technology for a wide range of emerging applications, including ocean monitoring for scientific exploration and commercial exploitation, safe CO_2 storage underwater, coastline protection, and prediction of underwater seismic events [1], [2], [3]. Typical underwater applications require multi-hop networks where sensor nodes transmit data to one of more collection points (*sinks*) located at the surface level. Sinks may then forward the received information to onshore control stations (usually via RF transmissions).

The major barriers for deploying underwater wireless sensor networks (UWSNs) come from the very specific environment where they operate. Underwater communications are characterized by long propagation delays, low bandwidth, sound speed variability, slow power signal attenuation, and many other environmental impairments. Moreover, the rapidly changing conditions of the acoustic channel may give rise to time-varying link reliability and to asymmetric links. These challenges affect underwater networking at all levels. In particular, MAC and routing protocols face new design challenges with respect to their terrestrial counterpart, in that channel access mechanisms and routing techniques are highly affected by a link quality that varies so differently from wireless radio links. Most of the solutions proposed for UWSN protocols at the MAC and routing layers address the problem of channel access and multi-hop routing separately [4], [5], [6], [7], [8], [9]. Recently, however, it has been shown that cross layer techniques can impact protocol performance positively, especially in networks with limited resources and/or deployed in challenging environments, like UWSNs [10]. An example of the cross-layer approach to routing is provided by the Focused Beam Routing (FBR) protocol, proposed by Jornet et al. in [11]. Control packets (RTS and CTS as in CSMA/CA-like channel access) are used to reserve the channel and to carry node location information that is used to select as next hop relay the neighbor that is closest to the sink. The use of short packets for channel access and relay selection results particularly effective for routing in the challenged UWSNs environment. This is because their transmissions is relatively fast even at the low rates of acoustic modems, and because they are less subject to bit errors and interferences. However, short control packets and distance information may not be sufficient to determine a relay that could actually receive a longer data packet correctly. In the quickly varying channel conditions, a sequence of bits that can be orders of magnitude longer than the length of a control packet can be easily affected by bit errors. In other words, in solutions like FBR link quality is not considered when it comes to selecting the next hop relay. This is a problem that affects many other cross layer solutions (see Section II).

In this paper, we present a new distributed cross-layer Channel-aware Routing Protocol (CARP) for UWSNs for multi-hop delivery of data to the sink. While still reaping the benefits of cross layer design (short packets for robust channel access and relay selection), CARP obviates to the drawbacks of other solutions in that link quality is explicitly taken into account in the cross layer relay selection. Relay are selected if they have a history of successful transmissions to their neighbors in routes to the sink. CARP also combines

link quality with simple topology information (hop count) for routing around connectivity holes and shadow zones. Among viable relays, a CARP node selects the neighbor with the highest residual energy, and that can receive the larger number of packets. CARP is also designed to take advantage of modem power control, when available, by selecting transmission powers in such a way that shorter control packets experience similar Packet Error Rate (PER) of longer data packets. This allows nodes to select robust and reliable links: When a relay has been selected through the initial handshake, the power is increased so that the following transmission of data packets has a similar likelihood of being successful as the handshake itself.

CARP has been evaluated through ns2-based simulations. In particular, we have compared its performance to that of FBR [11] and to the performance of the Depth-Based Routing (DBR) protocol [12], a popular scheme that has inspired the design of many underwater routing protocols where routes are found based on the depth of the nodes. The Bellhop ray tracing model with historical environmental data has been used to model the underwater acoustic channel. Historical environmental data were downloaded from databases available online (bathymetry, sound speed profile, sea-floor sediment). The use of bellhop has provided a more accurate description of the underwater acoustic channel behavior, computing the frequency-dependent acoustic path loss of each source-destination pair at a given location, as well as the spatially-varying interference induced by all active nodes. Our simulations, in scenarios with 100 nodes, show that CARP is an effective solution for transmitting packets through time-varying channels, capable of routing around connectivity holes and shadow zones, and to maintain a high packet delivery ratio for increasing traffic. CARP outperforms FBR and DBR with respect to key metrics such as throughput efficiency, end to end latency and energy consumption. Its link quality-based channel selection and the ability of selecting robust links through maintaining similar PER for control and data packets, allows its throughput to be always over 80%, independently of the wide variety of traffic we considered. This constitutes a remarkable improvement over the two other protocols, for which already at moderate traffic the same metric is always below 60%. More specifically, throughput improvements over FBR and DBR are 100% and 300%, respectively, at the highest considered traffic load. Our results also show that CARP achieves end-to-end packet latency up to 20% lower than that experienced by packets routed by FBR, and and up to 25% lower than that of packets routed by DBR. In terms of energy consumption, we observed that CARP obtains up to 70% energy savings with respect to FBR, while the improvement over DBR are up to sixfold. Our experimental evaluation is concluded by an assessment of the impact of idle listening (i.e., listening to the channel while waiting for packets to transmit of for incoming packets) on energy consumption. We observed that when modems are equipped with wake up mechanisms (low power devices that wake up the modem transceiver for incoming transmissions) all protocols show superior energy performance.

The rest of the paper is organized as follows. Previous work on underwater multi-hop routing is summarized in Section II. In Section III we describe CARP in details. The following Section IV illustrates our simulation results. Finally, Section V concludes the paper.

II. Related Work: Underwater Routing

Protocols for underwater communications have recently received noticeable attention [13], [14]. Among the first protocols to tackle the problem of finding routes from an underwater sensor node to the sink is the Vector-Based Forwarding (VBF) protocol presented by Xie et al. [15]. Nodes forward packets by broadcasting them to nodes residing in a constrained "pipe" of predefined radius (the radius is pre-selected) in the direction of the sink. The pipe surrounds a virtual line (a *vector*) between the packet source node and the sink. The efficiency of the protocol, especially its throughput, depends on the critical determination of the radius of the pipe: If the radius is too small, few or no relays can be found in the pipe; if it is too large, too many nodes might receive the packet, whose retransmission increases interference, overhead, and duplicate packets. VBF has been improved by Nicolaou et al. [16] by introducing multiple vectors, namely, from each relay to the sink. In this way, a hop-by-hop vector system is used to forward a packet (hence the name of the protocol: HH-VBF), which increases the probability to find a relay in the pipe, especially in sparse networks. Although HH-VBF has been shown to outperform VBF, it still depends on the correct determination of the radius of the forwarding pipe, and suffers from high overhead.

Another constrained flooding and geographic-based approach is presented by Jornet et al. [11]. The protocol, named *Focused Beam Routing* (FBR) assumes that each node knows its own location and that each source knows the location of the sink. FBR also assumes that nodes can choose their transmission power within a set of different power levels P_1 through P_n. Routing happens in a cross-layer fashion: When a node has a data packet to sent, it first transmits a control packet (e.g., a request-to-send, RTS packet) at power P_x, reaching only nodes d_x meters away. Only nodes that lie within a cone centered on the line between the transmitter and the sink (bounded directional flooding) are candidate relays. If there are no nodes within the transmission beam corresponding to P_x, the transmission power is increased to the next power level P_{x+1}. (If no relay is found with any of the available transmission powers, the node moves the beam to the right and then to the left, until a relay is found.) Each candidate relay node that receives an RTS packets replies with a clear-to-send (CTS) packet containing its position and other information. Among all nodes that have replied, the one closer to the sink is chosen as the next hop relay.

Directional flooding of data packets is the approach selected by Shin et al. [17] for their *Directional Flooding-based Routing* (DFR). Data packets are broadcast by each node S to all its neighbors. Based on directional information, namely, by the angle SFD between the sender S, the forwarder F

and the destination D (i.e., the sink), a node decides whether to forward the packet or not. The decision is made by comparing the angle SFD with a BASE_ANGLE carried by the packet. The varying conditions of the underwater channel are addressed by changing the BASE_ANGLE on a hop-by-hop basis based on the link quality: The better the latter, the smaller the flooding zone.

Since determining geographic information underwater can be problematic, or could require high cost/overhead, some protocols use partial geographic information. This kind of information, such as *depth* (distance from the surface), can be easily determined and at a greater accuracy [18]. This is the case of DBR, the *Depth Based Routing* protocol, presented by Yan et al. [12]. Each node that received a data packet forwards it only if its depth is less than that of the packet sender (and if it has not already sent the same packet before). Before forwarding the data packet, a node waits for a time (holding time) that depends on the difference between its own depth and that of the sender. In particular, the larger the vertical distance, the smaller the holding time, so that nodes that are closer to the surface (where the sink is) are the first to forward the data packet. While holding, a node listens to the transmissions on the channel. If it overhears that the data packet it is about to broadcast is transmitted by another node, the node drops the packet. The protocol uses also a depth threshold to reduce or to increase the number of nodes that can forward a packet. More specifically, a node that received a data packet forwards it only if the difference between its depth and that of the sender is greater than the selected depth threshold. If the threshold is small, a higher number of nodes retransmit the packets, which likely increases the packet delivery ratio, as well as energy consumption. Using a larger threshold reduces the number of forwarders, with possibly more troubles in delivering packets, while saving energy. The depth threshold can thus be selected to obtain a desired trade-off between packet delivery ratio and energy consumption.

Depth is the basic concept of protocols such as HydroCast (Lee et al. [19]) and VAPR (Noh et al. [20]). The idea of these solutions is similar to that of DBR: A node will forward a packet only if other nodes closer to the sink cannot send it. HydroCast tries to find a set of possible relays that maximize the Expected Packet Advance (Zeng et al. [21]), while limiting the number of nodes involved in the forwarding so to reduce redundant transmissions, packet collisions and therefore co-channel interference and the impact of hidden terminal phenomena. The protocol provides route recovery strategies in case packets get stuck at local maxima.[1] Recovery routes are discovered through a limited hop search of a 2D surface of a convex hull around a void zone. Node localization is centralized, performed by an off-line monitoring center to which nodes send their coordinates periodically. VAPR uses the same forwarding set selection algorithm of HydroCast. Nodes know their next-hop neighbor towards the sink thanks to the surface

---

[1] A node is a local maximum if there is no neighboring node with a lower pressure level.

reachability information. This information is propagated by the sink via periodic beaconing. Beacons contain information such as depth, hop count, and data forwarding direction. Each node that receives the beacon updates the information (e.g., hop count) and re-broadcasts the beacon. Using these directional trails the protocol performs a *local opportunistic directional flooding* to deliver the data. So, differently from HydroCast, VAPR relies on no recovery fallback for path maintenance, thus incurring inferior overhead. The effectiveness of VAPR in delivering packets to the sink is demonstrated via simulations, through which it is also shown that it outperforms HydroCast, DBR and a generic routing protocol where routes are only based on the distance in hops from the sink.

Further details on the cited protocols (and more results) can be found in recent surveys on the subject [9], [22].

III. PROTOCOL DESCRIPTION

At network set up, `HELLO` packets are flooded from the sink through the network. In this way, every node x acquires its hop count $HC(x)$, i.e., its distance, in hops, from the sink. Each `HELLO` packet carries information on its source node and the hop count information. The sink generates the first `HELLO` packet, setting its hop count field to 0, and broadcast it to its one hop neighbors. Each node x that receives an `HELLO` packet checks whether its $HC(x)$ is greater than the hop count carried by the packet plus 1. If this is the case, x updates its hop count to the value in the `HELLO` packet plus 1, and re-transmits the packet increasing its hop count field by 1. Otherwise, the `HELLO` packet is dropped. By the end of this flooding process a node has acquired its hop distance from the sink, as well as information about its neighbors towards the sink.

When a node x has one or more data packets to forward it chooses a suitable relay among its neighbors. The search for the relay is initiated by x by broadcasting a control packet, called `PING`, which carries the following information.

$$< src, num_{pkt} >.$$

The field src is x unique identifier, and num_{pkt} is the number of packets that x has to transmit. If $num_{pkt} > 1$ a *train* of packets is transmitted back to back.

A node y that receives the `PING` packet immediately replies with a `PONG` packet, directly transmitting it to the `PING` source x (unicast communication). The `PONG` packet carries the following information.

$$< src, dst, hop, queue, energy, lq >.$$

The fields src and dst contain the identifiers of nodes y and x, respectively. The field $queue$ indicates the available buffer space at y, i.e., the number of packets that y can store in its incoming data queue. The field hop contains $HC(y)$. The parameter $energy$ indicates the residual energy available at node y. The parameter lq is an indication of the quality of the links outgoing from y (see detailed description below).

Relay selection happens as follows. After sending a `PING` packet, node x awaits for `PONG` replies 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. After the time δ, node x uses the link quality information lq_y sent in the PONG packets from all its available neighbors y, 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:

$$goodness_y = lq_y lq_{x,y}.$$

The node y with the highest ratio $\frac{goodness_y}{HC(y)}$ is chosen as the relay, and the (train of) data packet(s) is sent directly to it. In so doing, nodes 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 (as encoded by $queue$). Further ties are broken by using the node unique identifiers. Upon receiving a train of data packets, a node y replies with a cumulative ACK, acknowledging each single packet in the train (bit mask). Upon receiving an acknowledgment from y, node x updates its hop count to $HC(y) + 1$. 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 to re-transmit only those that it has not forwarded already.

A. Computing the link quality lq

The $goodness_y$ that each node x computes for all its neighbors y that replied with a PONG packet 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 link quality lq_y is computed by y based on the success of past transmissions to its neighbors. It is defined as an exponential moving average, where the weight of transmissions back in the past are less influential than recent ones in assessing the goodness of the link for transmissions. This enables CARP to take into account the time varying nature of the channel, giving more importance to what has happened recently. More formally, for each data packet transmissions t to one of its neighbors z, node y computes:

$$lq_z^t = \alpha Y_z^t + (1 - \alpha)lq_z^{t-1}.$$

The coefficient $\alpha \in (0, 1)$ is the constant smoothing factor through which we can control how quickly the influence of older transmissions decreases. For instance, a higher α could be used for very variable underwater channels, as it discounts older transmissions faster. Y_z^t is the success ratio of the tth transmission from y to z, defined as the ratio between the number of packets correctly received by z (i.e., that z acknowledges positively) and the number of packets sent in the train of that transmission. lq_z^{t-1} is the value of the moving average after $t - 1$ transmissions from y to z. Since this definition is recursive, we define lq_z^1 as the success ratio

of the first transmission. The value lq that node y transmits in its PONG packet to x is the best among the lqs to all its neighbors, based on the (different) data transmissions with each of them. The value $lq_{x,y}$ used by x for computing $goodness_y$ is computed similarly, considering data packet transmissions from x to y.

B. CARP and power control

In general, cross-layer protocols based on a handshake mechanism for joint channel access and relay selection (such as CARP and FBR) determine a next hop relay based on the correct exchange of control packets. Once a neighbor has been selected as relay, the channel is reserved and used for data transmission. However, an acceptable PER for short control packets might result in a (too) high PER for data packets, which are usually considerably longer. Although not all commercial modems allow the selection of any given transmission power, CARP is designed to take advantage of power control, when available, to obtain similar desirable PER for both control and data packets. The power used to transmit PING packets is computed so to obtain a PER corresponding to a given channel BER. Once a relay has been selected, the power for sending data packets is increased so that the corresponding PER is the same experienced by the PING/PONG exchange through which the relay has been determined. More precisely, the transmission power P for transmitting packets at a given PER is computed as follows. Using a BPSK modulation the probability to transmit correctly a packet that is l bits long is $= (1 - BER)^l$. (Assuming no forward error correction.) The BER is computed as $\frac{1}{2}\text{erfc}(\sqrt{SNR})$, where erfc() is the complementary error function. The SNR is given by $\frac{P/A(r,f)}{N(f)\Delta f}$, where P is the required transmission power, $A(r, f)$ is the attenuation in the underwater channel over a distance r for a signal of frequency f, $N(f)$ is the noise power spectral density, and Δf is the receiver noise bandwidth [23].

IV. PERFORMANCE EVALUATION

This section describes the comparative performance evaluation of CARP and two other previously proposed protocols for UWSN routing, namely, FBR [11] and DBR [12], described earlier. CARP, FBR and DBR have been implemented using ns2-MIRACLE [24] on top of ns-2 [25], connected to the Bellhop propagation simulator [26] via the WOSS interface [27]. Bellhop is used to compute acoustic path loss at a given location, as well as the spatially-varying interference induced by node transmissions. Real environmental data are used, from an area located in the Mediterranean sea off the coast of the Pianosa island (Tuscan archipelago), with the coordinate $(0, 0, 0)$ of the surface located at $42°, 32′, 0″N$ and $10°, 22′, 0″E$. Sound speed profiles (SSP), bathymetry profiles and information on the type of bottom sediments of the selected area are obtained from the World Ocean Database [28], from the General Bathymetric Chart of the Oceans (GEBCO) [29] and from the National Geophysical Data Center's Deck41 data-base [30], respectively. The SSP is

retrieved by WOSS from the World Ocean Database (average of measurements from September 2009).

We start by describing the selected scenarios and protocol parameters settings (Section IV-A), we discuss the metrics that we have investigated (Section IV-B) and we finally report the results of our simulation-based experiments (Section IV-C).

A. Simulation scenarios and settings

We consider scenarios similar to those we explored in [31]. Specifically, our simulations concern UWSNs with 100 nodes (99 nodes plus the sink) statically placed in a region with surface 4km × 4km. Sensor nodes are placed randomly and uniformly in the region at different depths, ranging from 20 to 100m. Every node has an average of 15 neighbors. Each packet that makes it to the sink traverses an average of 2.4 hops (the maximum number of hops is 4). The sink is placed centrally on the surface with the transducer 10m below. Data packet payload is 3000 bytes long. This is the optimized value of packet size for the desired BER of 10^{-6} of links on the routes to the sink [31]. The carrier frequency is 24kHz for a bandwidth of 2000Hz. Bandwidth efficiency is set to 1bps/Hz. We assume BPSK modulation. Topology construction ensures that each node has at least one route to the sink going through robust links (with respect to SNR). For the selected value of the bandwidth and the chosen packet size the transmission power is set to 3.3W, resulting in an average BERs on the routes of 10^{-6}.

Idle, reception and transmission power were estimated based on the energy consumption of existing acoustic modems. In order to have similar PERs for control packets and data packets, CARP control packets are transmitted with a transmission power of 1.5W, corresponding to a source power level at 1m of 5dB re μPa less than the transmission power used for data packets. The FBR protocol uses 4 levels of transmission power corresponding to 3.3W, 4.0W, 5.6W and 8W.

Traffic is generated according to a Poisson process with aggregate (network-wide) rate of λ packets per second. Once a packet is generated, it is associated with a source selected randomly among all nodes (but the sink). The destination of all packets is the sink. We define the normalized packet rate as $\overline{\lambda} = \lambda T_{\text{pack}}$, whose values are considered in the range 0 to 1 packets per packet time. Packet time is expressed as $T_{\text{pack}} = N_b/R_b$, where N_b is the packet size in bits and R_b is the bit rate. The results presented here concern very low traffic ($\overline{\lambda} = 0.01$), low traffic ($\overline{\lambda} = 0.1, 0.15$), medium traffic ($\overline{\lambda} = 0.2, 0.25$) and high traffic ($\overline{\lambda} = 0.3$).

The total size of a data packet is given by the payload (3000B) plus the headers added by the different layers. The physical layer header contains all the information needed by the modem to correctly start receiving a packet (synchronization preamble, delimiters, etc.). At the physical layer, nodes need a synchronization peering time which is taken to be on the order of 10ms (the physical header overhead changes according to the data rate). The DBR protocol uses a CSMA MAC protocol without ACKs. The CSMA header contains the sender and the destination IDs, and the packet type. Its length

is 3B. The size of the DBR routing header is 4B long. FBR RTS and CTS packets are 6B long; its ACK and WARNING packets are 3B long. The size of PING and PONG packets is 5B; CARP ACK packets are 4B long, while the HELLO packets are 3B long. The CARP MAC header is 4B long. Finally, the smoothing factor α used for computing CARP link quality has been set to 0.7. Parameter setting and topology properties are summarized in Table I.

B. Simulation metrics

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

- *Throughput efficiency*, i.e., the ratio between the average bit rate successfully delivered to the sink and the average offered bit rate, $N_b\lambda$.
- *End-to-end latency per meter*, defined as the time between the packet generation and the time of its correct delivery at the sink, divided by the distance between source and destination. Normalization by distance is used so as to unify the performance over varying deployment areas (a larger area will entail proportionally larger propagation delays).
- *Energy per bit*, i.e., the energy consumed by the network to correctly deliver a bit of data to the sink.

C. Simulation results

Throughput efficiency. Figure 1 shows the throughput efficiency of the three considered protocols for increasing traffic $\overline{\lambda}$. As expected, when the traffic increases the packet deliv-
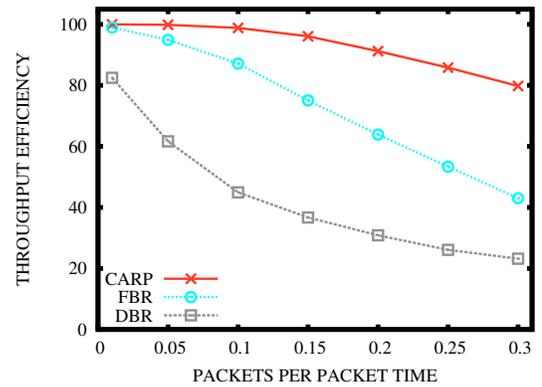
Figure 1: Throughput efficiency.

ery ratio decreases. The increased traffic results in higher probability of packet collisions and re-transmissions as well as in a higher number of times the nodes find the channel busy. CARP clearly outperforms both FBR and DBR. This is because of its link quality-based relay selection and also because data are forwarded on links that are robust for both control and data packets (Section III-B). When $\overline{\lambda} < 0.17$ CARP delivers more than 95% of the generated data to the sink. When the traffic load increases up to 0.3 packet per packet time, CARP throughput efficiency stays above 80%, an acceptable value for most monitoring applications. FBR is

Table I: Simulation parameters and topology properties.

Number of nodes (N)	100
Deployment area surface	4000m $\times$ 4000m
Deployment area depth	100m
Latitude of coordinate (0,0,0) of the deployment area	$42°, 32', 0''N$
Longitude of coordinate (0,0,0) of the deployment area	$10°, 22', 0''E$
Average network density	15
Average/maximum route length	2.4/4 hops
Data payload (N_b)	3000 bytes
Control packet size (all protocols)	$\leq$ 6B
Carrier frequency	$\sim$ 24kHz
Bandwidth	2000Hz
Bandwidth efficiency	1 bps/Hz
Detection threshold	1dB
Idle power	0.085W
Reception power	0.62W
CARP transmission power for BER $= 10^{-6}$ for data / control packets	3.3W / 1.5W
FBR transmission power levels	3.3W, 4W, 5.6W and 8W
Bit rate (R_b)	2000bps
Traffic (packets per packet time $\overline{\lambda}$)	0.01, 0.05, 0.1, 0.15, 0.2, 0.25 and 0.3
CARP smoothing factor α for link quality computation	0.7

able to correctly deliver more than 80% of the data packet only when $\overline{\lambda} < 0.12$. However, as soon as the traffic increases ($\overline{\lambda} = 0.3$) its throughput efficiency shows a node dive decrease to 40%. This is because although, like CARP, FBR exploits power control for finding paths towards the sink, once a power level is selected, it is used for both control and data packets. In FBR, relay selection is based on the correct transmission and reception of RTS and CTS packets, which are much shorter than data packets. This means that the link to the selected relay has low probability of error for short control packets. However, when long data packets are sent over the chosen link the probability of error gets higher, and the selected link can turn out to be unreliable for data. If transmissions at power P_x are not correctly acknowledged, power is increased to the next level P_{x+1}, and relay selection is started again at this power level. If the link was unreliable for transmitting data packet at P_x, using higher power decreases the probability of data error *on that link*. However, the new search at P_{x+1} could yield to the selection of a new relay, closer to the sink (because of FBR greedy approach to relay selection), different from the one chosen at P_x. Therefore, although the link to the new relay is reliable for control packets, it might again be unreliable for longer data packets, resulting in further incorrect data communications. Switching to a higher level of power for this reason is quite common for FBR. Figure 2 shows the average number of times that FBR nodes increase their power within a relay search due to incorrect data transmissions. We observe that for $\overline{\lambda} > 0.14$ FBR goes through at least two power increments, which affects its throughput efficiency (and the protocol overall performance, see below).[2]

The poor throughput performance of DBR, even at low traffic, is due to its relay selection mechanism. Of the different

[2] Recall that once exhausted all power levels for lack of finding a reliable relay/link, an FBR node enlarges its search beam and starts the search again at the lower power. While we observed frequent increases of power levels, we noticed sporadic beam widenings, and only at very high traffic.
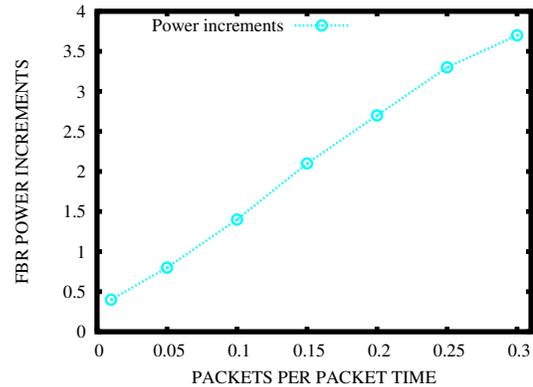
Figure 2: Number of FBR power increments.

flavors of DBR, we have implemented the one where a node chooses a relay among all neighbors at equal or lower depth. According to the investigation in [12], this is the version that maximizes channel utilization. However, this relay choice is irrespective of whether the relay is in the direction of the sink or not. If the packet reaches a node that has no neighbors closer to the surface, and this node cannot directly communicate to the sink, the packet is discarded. This is a problem of both FBR and DBR. CARP, instead, by using hop count information, can find routes to the sink independently of whether the advancement towards the sink provided by the selected node is positive or negative.

End-to-end latency per meter. The average latency per meter experienced by data packets successfully routed by the three considered protocols is shown in Figure 3.

All three protocols show good performance, especially considering that each data packet transmission lasts at least 12 seconds and that each packet successfully delivered to the sink travels, on average, multiple hops. Increasing the traffic brings degraded performance, as expected. This happens even
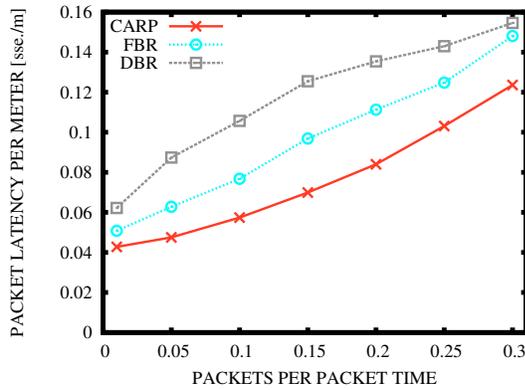
Figure 3: Latency per meter.

if the average number of hops traveled by each packet stays relatively stable, as depicted in Figure 4. Each hop imposes
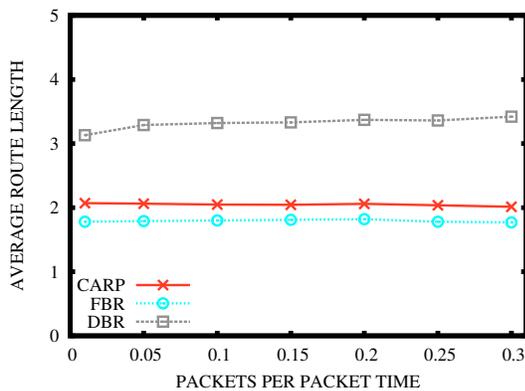
Figure 4: Route length.

longer delays because of the increased number of collisions and retransmissions. CARP shows the best latency per meter performance, due to its relay selection based on link quality and the fact it considers robust links for both control and data packets, which keeps interference and retransmission at bay. FBR performance is beset by many power increments and retransmissions to find a suitable relay, which impose longer times. We observe that, however, increasing the power enables selecting relays that provide higher advancements toward to the sink, which explains why FBR has the shortest averages routes to the sink (Figure 4). The reason why DBR latency performance is inferior to that of the other two protocols has to do with the longer DBR routes. While packets that are generated by node in the central area of the considered scenarios reach the sink quickly, going vertically from their source to the sink in very few hops, packets generated by nodes at the network edges travel vertically first, towards the surface, and then horizontally towards the sink. This makes routes longer, on average (Figure 4), and imposes, as a consequence longer delays.

Energy consumption per bit. The final set of results we present concerns the energy spent for delivering one bit to

the sink correctly. Figure 5 shows the energy consumed for each data bit successfully delivered to the sink.
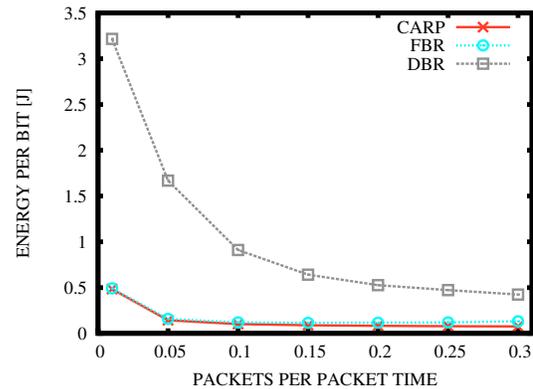
Figure 5: Energy consumption per bit: Case with idling.

These results concern the case where nodes are always on, namely, we consider energy spent for transmitting and receiving a bit as well as that spent when a node just listens to the channel (idling). When the traffic is low, the time spent idling is much longer than the time spent transmitting and receiving. Furthermore, the number of bits delivered to the sink is low. Consequently, energy consumption per bit is higher. When traffic increases more bits are correctly delivered to the sink, and although energy consumption increases and packet delivery ratio decreases, the consumption per bit decreases, as shown by the curves for each protocol.[3] DBR consumes more than CARP and FBR for delivering a bit. This is because, being a flooding-based protocol, it incurs a higher number of data packet transmissions. Moreover, since it correctly delivers to the sink a lower number of bits and since each bit travels longer routes than those of CARP and FBR, its energy demands are higher.

Figure 6 zooms into CARP and FBR energy efficiency.
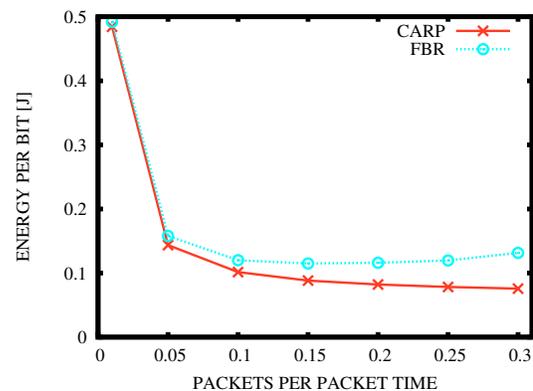
Figure 6: Energy consumption per bit: CARP and FBR (zoom).

---

[3] Although throughput efficiency decreases for increasing traffic (Figure 1), the actual number of bits generated and correctly delivered to the sink increases as well.

When traffic is low both protocols show very similar performance. This is because they deliver correctly very similar amounts of bits to the sink, and because the number of power increments of FBR is still low. As soon as the traffic increases, CARP delivers more information to the sink, and the number of packet re-transmissions for FBR increases. We observe that for new retransmission FBR uses a higher power level (up to the highest one). This explains why CARP saves 20% more energy when $\overline{\lambda} = 0.1$ and up to 70% when $\overline{\lambda} = 0.3$. We also notice that FBR energy performance decreases as the traffic increases (as CARP and DBR), but then, differently from the other protocols, it starts increasing. This is due to the fact that at higher traffic FBR needs a higher number of retransmissions, effectively delivering each packet at the highest of its power levels (Figure 2). In this case, the fact that a greater number of bits are correctly delivered to the sink is not enough to lower the energy consumption per bit, as for CARP and DBR.

Idling is a major culprit of energy expenditure, especially at low traffic. In order to limit this detrimental blow to performance, recent commercial transducers are being built that have "wake up" capabilities. Modems are now endowed with very low power devices capable of alerting a node of upcoming transmissions. Considerable results are being seen for terrestrial radio nodes [32], [33], and similar research is ongoing for underwater modems. For instance, Teledyne Benthos modems [34] feature low power wake up, and Develogic Subsea System Ham.Node [35] implements a very low power sleep mode as well as a low power acoustic stand by mode. The last set of our simulation results concerns the investigation of energy consumption per bit in networks with nodes equipped with "wake up modems," where idling effects are negligible. Results are shown in Figure 7.
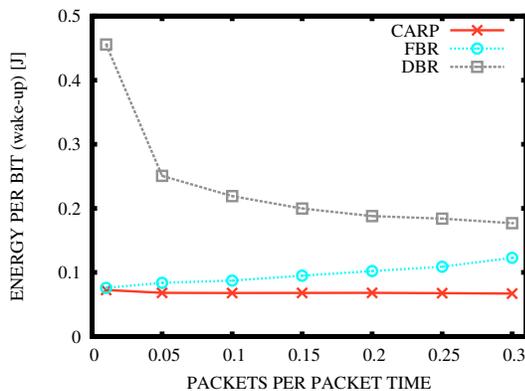
Figure 7: Energy consumption per bit: Case with "wake up modems."

The observed improvement is remarkable for all protocols, indicating the importance of providing modems with wake up capabilities. For each given $\overline{\lambda}$, correctly delivering more bits results in spending energy more effectively, with a corresponding better energy per bit performance. Being lightweight and able to correctly deliver packets to the sink,

CARP significantly outperforms other protocols, resulting in an energy per bit that is a half of that of FBR and one third of that of DBR when the traffic is high ($\overline{\lambda} = 0.3$).

V. CONCLUSIONS

In this paper we presented CARP as an efficient new protocol for UWSNs. CARP follows the cross layer design paradigm in that it efficiently exploits short control messages to perform joint channel access and relay selection. The well know approach is enriched by CARP with the introduction of link quality information in the cross layer relay selection. Robustness of the selected link is also achieved by computing the transmission power so to obtain similar PER for short control packets and longer data packets, thus allowing to exploit the short control packet exchange to identify links, which result in reliable data transmissions. A comparative simulation-based performance evaluation of CARP, FBR and DBR reveals that including link quality explicitly into relay selection is key to obtain superior throughput efficiency, end-to-end latency and energy consumption.

ACKNOWLEDGMENTS

REFERENCES

[1] J. Kalwa, "The RACUN-project: Robust acoustic communications in underwater networks—an overview," in *Proceedings of IEEE/OES OCEANS 2011*, June 6–9 2011.

[2] K. Kawaguchi, E. Araki, S. Kaneko, T. Nishida, and T. Komine, "Subsea engineering ROV and seafloor observatory construction," in *Proceedings of the IEEE Symposium on Underwater Technology, UT 2011*, April 5–8 2011.

[3] N. Meratnia, P. J. M. Havinga, P. Casari, C. Petrioli, K. Grythe, T. Husoy, and M. Zorzi, "CLAM—collaborative embedded networks for submarine surveillance: An overview," in *Proceedings of IEEE/OES OCEANS 2011*, June 6–9 2011.

[4] J. Yackoski and C. Shen, "UW-FLASHR: Achieving high channel utilization in a time-based acoustic MAC protocol," in *Proceedings of the third ACM International Workshop on UnderWater Networks (WUWNet '08)*, San Francisco, California, USA, September 15 2008, pp. 59–66.

[5] N. Chirdchoo, W.-S. Soh, and K. C. Chua, "ALOHA-based MAC protocols with collision avoidance for underwater acoustic networks," in *Proceedings of IEEE INFOCOM 2007, the 26th IEEE International Conference on Computer Communications*, Anchorage, Alaska, USA, May 6–12 2007.

[6] B. Peleato and M. Stojanovic, "Distance aware collision avoidance protocol for ad-hoc underwater acoustic sensor networks." *IEEE Communications Letters*, vol. 11, no. 12, pp. 1025–1027, December 2007.

[7] A. Syed, W. Ye, and J. Heidemann, "Comparison and evaluation of the T-Lohi MAC for underwater acoustic sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 9, pp. 1731–1743, December 2008.

[8] M. Chitre, M. Motani, and S. Shahabudeen, "A scheduling algorithm for wireless networks with large propagation delays," in *Proceedings of IEEE OCEANS 2010*, Sydney, Australia, May, 24–27 2010, pp. 1–9.

[9] M. Ayaz, I. Baig, A. Abdullah, and I. Faye, "A survey on routing techniques in underwater wireless sensor networks," *Elsevier Journal of Network and Computer Applications*, vol. 34, no. 6, pp. 1908–1927, June 2011.

[10] R. Jurdak, A. G. Ruzzelli, and G. M. P. O'Hare, "Design considerations for deploying underwater sensor networks," in *Proceedings of SensorComm 2007*, October 14–20 2007, pp. 227–232.

[11] J. M. Jornet, M. Stojanovic, and M. Zorzi, "Focused beam routing protocol for underwater acoustic networks," in *Proceedings of the third ACM International Workshop on UnderWater Networks, WUWNet 2008*, San Francisco, California, September 15 2008, pp. 75–82.

[12] H. Yan, Z. J. Shi, and J.-H. Cui, "DBR: Depth-based routing for underwater sensor networks," in *Proceedings of the 7th international IFIP-TC6 NETWORKING 2008*, May 5–9 2008, pp. 72–86.

[13] D. Pompili and I. F. Akyildiz, "Overview of networking protocols for underwater wireless communications," *IEEE Communications Magazine*, vol. 47, no. 1, pp. 97–102, January 2009.

[14] P. Casari and M. Zorzi, "Protocol design issues in underwater acoustic networks," *Elsevier Computer Communications*, vol. 34, no. 17, pp. 2013–2025, November 2011.

[15] P. Xie, J. H. Cui, and L. Lao, "VBF: Vector-based forwarding protocol for underwater sensor networks," in *Proceedings of IFIP-TC6 NETWORKING 2006*, Coimbra, Portugal, May 15–19 2006.

[16] N. Nicolaou, A. See, X. Peng, J.-H. Cui, and D. Maggiorini, "Improving the robustness of location-based routing for underwater sensor networks," in *Proceedings of MTS/IEEE OCEANS 2007*, June 18–21 2007.

[17] D. Shin, D. Hwang, and D. Kim, "DFR: An efficient directional flooding-based routing protocol in underwater sensor networks," *Wireless Communication and Mobile Computing*, January 2011.

[18] B. Jalving, "Depth accuracy in seabed mapping with underwater vehicles," in *Proceedings of MTS/IEEE OCEANS 1999*, September 13–16 1999.

[19] U. Lee, P. Wang, Y. Noh, L. Vieira, M. Gerla, and J.-H. Cui, "Pressure routing for underwater sensor networks," in *Proceedings of IEEE INFOCOM 2010*, March 15–19 2010.

[20] Y. Noh, U. Lee, P. Wang, B. S. C. Choi, and M. Gerla, "VAPR: Void aware pressure routing for underwater sensor networks," *IEEE Transactions on Mobile Computing*, February 2012.

[21] K. Zeng, W. Lou, J. Yang, and D. R. Brown, "On geographic collaborative forwarding in wireless ad hoc and sensor networks," in *Proceedings of the International Conference on Wireless Algorithms, Systems and Applications, WASA 2007*, Washington, DC, 2007, pp. 11–18.

[22] A. Wahid and K. Dongkyun, "Analyzing routing protocols for underwater wireless sensor networks," *International Journal of Communication Networks and Information Security*, vol. 2, no. 3, pp. 253–261, December 2010.

[23] M. Stojanovic, "On the relationship between capacity and distance in an underwater acoustic communication channel," *ACM SIGMOBILE Mobile Computing and Communications Review, MC2R*, vol. 11, no. 4, pp. 34–43, October 2007.

[24] N. Baldo, F. Maguolo, M. Miozzo, M. Rossi, and M. Zorzi, "ns2-MIRACLE: A modular framework for multi-technology and cross-layer support in network simulator 2," in *Proceedings of the 2nd International Conference on Performance Evaluation Methodologies and Tools, ValueTools 2007*, Nantes, France, October 23–25 2007, pp. 1–8.

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

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

[27] F. Guerra, P. Casari, and M. Zorzi, "World ocean simulation system (woss): a simulation tool for underwater networks with realistic propagation modeling," in *Proceedings of the Fourth ACM International Workshop on UnderWater Networks*, ser. WUWNet '09, Berkeley, California, USA, 3 November 2009, pp. 1–8.

[28] "World ocean atlas." [Online]. Available: www.nodc.noaa.gov/OC5/WOA05/pr_woa05.html

[29] "General bathymetric chart of the oceans." [Online]. Available: www.gebco.net

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

[31] S. Basagni, C. Petrioli, R. Petroccia, and M. Stojanovic, "Optimized packet size selection in underwater WSN communications," *IEEE Journal of Oceanic Engineering*, 2012, to appear.

[32] L. Gu and J. A. Stankovic, "Radio-triggered wake-up for wireless sensor networks," *Real-Time Systems*, vol. 29, no. 2–3, pp. 157–182, March 2005.

[33] G. Lu, D. De, M. Xu, W.-Z. Song, and B. Shirazi, "A wake-on sensor network," in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems, SenSys 2009*, November 4–6 2009, pp. 341–342.

[34] T. Benthos, "Teledyne benthos undersea systems and equipment." [Online]. Available: http://www.benthos.com/acoustic-telesonar-modem-product-comparison.asp

[35] Develogic Subsea System, "Develogic Subsea System Ham.Node." [Online]. Available: http://www.develogic.de/products/underwater-communication-systems/ham-node/