

Underwater Messaging Using Mobile Devices

Tuocho Chen,[◊] Justin Chan[◊] and Shyamnath Gollakota

[◊]Co-primary student authors

Paul G. Allen School of Computer Science & Engineering

University of Washington, Seattle, WA, USA

underwatermessaging@cs.washington.edu

ABSTRACT

Since its inception, underwater digital acoustic communication has required custom hardware that neither has the economies of scale nor is pervasive. We present the first acoustic system that brings underwater messaging capabilities to existing mobile devices like smartphones and smart watches. Our software-only solution leverages audio sensors, i.e., microphones and speakers, ubiquitous in today's devices to enable acoustic underwater communication between mobile devices. To achieve this, we design a communication system that in real-time adapts to differences in frequency responses across mobile devices, changes in multipath and noise levels at different locations and dynamic channel changes due to mobility. We evaluate our system in six different real-world underwater environments with depths of 2-15 m in the presence of boats, ships and people fishing and kayaking. Our results show that our system can in real-time adapt its frequency band and achieve bit rates of 100 bps to 1.8 kbps and a range of 30 m. By using a lower bit rate of 10-20 bps, we can further increase the range to 100 m. As smartphones and watches are increasingly being used in underwater scenarios, our software-based approach has the potential to make underwater messaging capabilities widely available to anyone with a mobile device.

Project page with open-source code and data can be found here:

<https://underwatermessaging.cs.washington.edu/>

CCS CONCEPTS

• **Networks** → **Mobile networks**; • **Applied computing** → *Environmental sciences*; • **Human-centered computing** → **Ubiquitous and mobile devices**;

KEYWORDS

Underwater communication, ocean sciences, SOS beacons, mobile phones, smart watches, underwater exploration

ACM Reference Format:

Tuocho Chen,[◊] Justin Chan[◊] and Shyamnath Gollakota. 2022. Underwater Messaging Using Mobile Devices. In *ACM SIGCOMM 2022 Conference (SIGCOMM '22)*, August 22–26, 2022, Amsterdam, Netherlands. ACM, New York, NY, USA, 15 pages. <https://doi.org/10.1145/3544216.3544258>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGCOMM '22, August 22–26, 2022, Amsterdam, Netherlands

© 2022 Copyright held by the owner/author(s). Publication rights licensed to the Association for Computing Machinery.

ACM ISBN 978-1-4503-9420-8/22/08...\$15.00

<https://doi.org/10.1145/3544216.3544258>

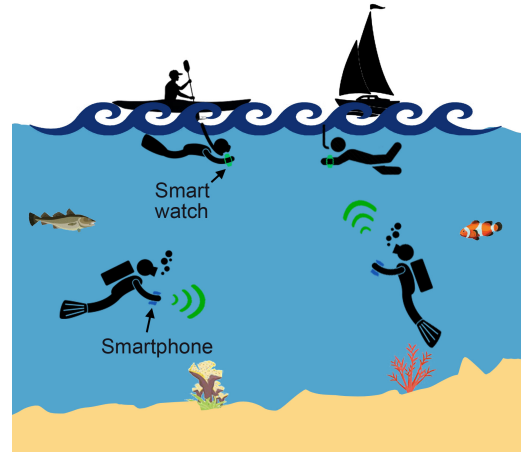


Figure 1: Mobile devices for underwater messaging.

1 INTRODUCTION

Each year, tens of millions take part in underwater activities like snorkeling in lakes, rivers, and oceans [14]. Millions more participate in recreational scuba diving in the United States alone [15]. Underwater exploration is also important in observation of marine life, water pollution and to document the biological, geological, and archaeological aspects of underwater environments [23]. Effective communication during these underwater activities is critical for safety and navigation [4, 18, 19]. Hand signals are commonly used to communicate intention, convey direction and maintain safety. In addition to the 10-20 signals commonly used in recreational settings [18], professional divers use more than 200 hand signals to communicate with each other or with surface members of the dive team [19]. This includes information about oxygen level, aquatic life or operations that require cooperation [4]. Given the number of hand signals and their visual nature, however, this mode of communication is ineffective in low-visibility scenarios (e.g., turbid waters) and is limited in its communication range and reliability.

To address this problem, commercial efforts have designed hardware that enables two-way text messaging to send from a set of pre-defined messages or SOS beacons [1]. DARPA also initiated the AMEBA effort to build custom hardware that would enable divers to communicate at a low bit-rate with each other via text messages or with nearby relay buoys [3]. These prior efforts require custom hardware that is neither ubiquitous nor has the economies of scale.

In this paper, we take a different approach and explore if one can enable underwater messaging capabilities on mobile devices like smartphones and smartwatches. Smartphones are increasingly being used with diving-proof cases (\$30-40) for underwater photography, video logging and in lieu of a dive computer [12, 13]. These



Figure 2: User interface of our messaging system on a phone within a water-proof case.

diving-proof smartphone cases are rated to work at depths of 15-40 m depending on their price [11, 21]. The latest smartwatches are also water-resistant and can be used during shallow water activities like snorkeling [6, 10]. Ideally, anyone with a mobile device should be able to download a software app and communicate underwater during snorkeling, recreational diving or scientific exploration, without additional hardware. Such an approach would leverage the ubiquity of mobile devices to democratize underwater communication and make it available to anyone with a mobile device.¹

We present *AquaApp*, the first software-only solution that enables underwater acoustic communication and networking on commodity mobile devices. While the most common communication modality on mobile devices is to use radios (e.g., Wi-Fi and LTE), these frequencies are not suitable for underwater communication – 2.4 GHz Wi-Fi signals can attenuate as much as 169 dB per meter in seawater [36, 46]. Our experiments show that two smartphones separated by just a few inches in fresh water could not connect using Wi-Fi or Bluetooth. Our solution instead is to leverage audio sensors, i.e., microphones and speakers, that are ubiquitous in today’s devices to enable acoustic underwater communication. In contrast to RF signals, acoustic signals have much better propagation properties underwater [36].

Repurposing microphones and speakers on commodity mobile devices for underwater communication is challenging for three reasons: (1) Unlike custom hydrophone hardware that is specially designed to operate underwater, acoustic sensors in smartphones and watches are designed for in-air operation and the hardware components used can vary across device manufacturers. This results in different frequency responses across devices (Fig. 3a,b). The transmit power on these mobile devices is also typically limited compared to underwater hydrophones. In addition, the severe multipath in underwater scenarios can result in the signal strength varying by as much as 10-20 dB within a few kHz. (2) Even with the same smartphone model used across users, the SNR profiles on the forward and backward paths can be different (Fig. 3c,d), resulting in the need to use different frequency bands on the two paths. (3) The bit rate can vary an order-of-magnitude from 100 bps to 1.8 kbps with distance and multipath, requiring an adaptation algorithm to minimize packet error rate. However, mobility inherent to diving and snorkeling results in varying channel across packets, making adaptation challenging. This requires a real-time protocol to pick

¹Recent reports note Apple’s R&D interest in enabling iPhones to communicate in underwater environments [7, 8].

the correct bit rate before transmitting data, without incurring significant overhead.

At a high level, we use orthogonal frequency division multiplexing (OFDM) to communicate underwater between mobile devices. Our real-time system adapts the acoustic frequencies used to encode data in each packet transmission, as a function of frequency response, distance and signal-to-noise ratio (SNR) of each frequency bin. This ensures that when the devices are a few meters apart and have a high SNR, the system adapts in real-time to use all the OFDM bins to encode data and achieve a high bit rate. On the other hand, it uses a smaller frequency band as the distance increases. This ensures that more power is allocated to a right set of frequencies, thus, increasing the SNR and the packet delivery rate.

Our design has three key components:

- *Post-preamble feedback.* In high-data rate systems (e.g., Wi-Fi), rate adaptation is performed across packets since packet sizes are less than a millisecond, and the channel coherent time is more than a few packets. In contrast, our underwater acoustic system is low rate making packets much longer and resulting in the channel changing between consecutive packets. To do this, we perform per-packet adaptation by splitting the preamble/header and data portions of each packet. Say Alice wants to send a packet to Bob. In our design, Alice first broadcasts the preamble and header with Bob’s address and stays silent for Bob’s feedback before transmitting the data portion of the packet (Fig. 5). Bob estimates the SNR for each OFDM bin using the preamble, runs a frequency adaptation algorithm and in real-time sends information back to Alice, embedded in a single OFDM symbol. Alice uses this feedback to transmit the data portion of the packet using the right frequency band.
- *Frequency band adaptation.* Per-frequency rate adaptation is ideally performed using the water-filling algorithm to allocate different power and modulation to each OFDM bin [65]. In a low-data rate system, however, conveying fine-grained feedback about 60 OFDM bins requires at least $O(60)$ bits which is a significant overhead. To minimize the feedback Bob sends back to Alice, we design a low-overhead frequency band adaptation algorithm (§2.2.2). At a high level, we first compute the SNR in each bin using the received preamble. If not all the OFDM bins are above an SNR threshold, we drop the lowest SNR bin and reallocate power to the remaining bins. We repeat this process until we find the largest contiguous band where all frequencies are above the SNR threshold. Bob sends back only information about the start and end frequencies, f_{start} and f_{end} , of this contiguous band.
- *Feedback encoding method.* Bob encodes this feedback using a single OFDM symbol. Our encoding method allocates all the power to the two OFDM bins corresponding to the start and end frequencies, f_{start} and f_{end} , output by our frequency band adaptation algorithm. Alice can extract this information by performing a sliding window and picking the top-2 OFDM bins with the highest power. This is a reliable encoding method since all the transmit power is being allocated to these two OFDM bins. Alice then uses this band to send data by setting the OFDM bins outside it to zero.

We implemented our software system in real-time on the Android platform so it can be used with various smartphone models and smart watches. Since underwater multipath can have a large

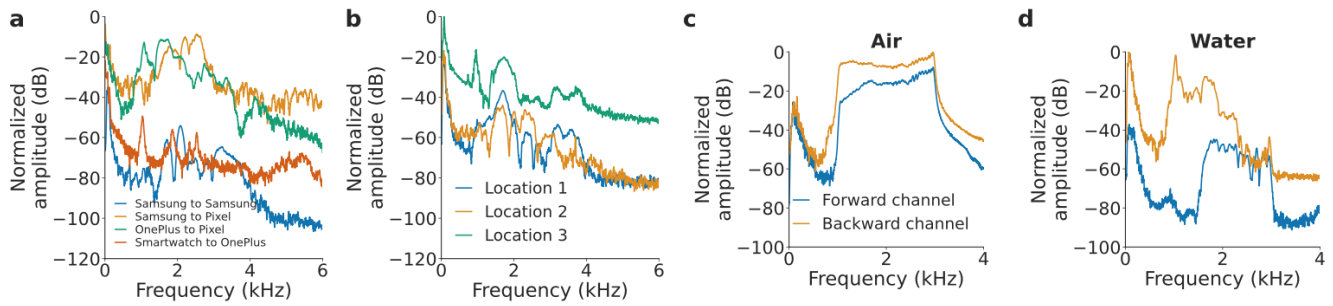


Figure 3: (a,b) Frequency selectivity underwater in a lake at a 5 m distance across devices and locations. (c,d) Channel reciprocity on the forward and backward path in (c) air and (d) water separated by 2 m.

delay spread, we implement time-domain equalization to reduce the cyclic prefix duration to only 7% of the OFDM symbol. To address mobility, we use differential coding across consecutive OFDM symbols to reduce errors (§2.3). We also design a carrier sense based MAC protocol to support a network of multiple underwater mobile devices (§2.4). Using our communication system, we implement a messaging app where users can transmit one of the 240 common messages. Finally, by using a lower bit rate, we design a longer-range SoS beacon messaging system.

We evaluate our system in six different real-world underwater environments. Our findings are as follows.

- Our system adapts the frequency band to achieve bit rates of 100 bps-1.8 kbps at distances up to 30 m. The bit rate scales with distance and multipath. By reducing the bit rate to 10-20 bps, our system can increase its range to 100 m.
- Our real-time adaptation algorithm reduces the average packet error rate (PER) from 38-70% to 3% compared to fixed bandwidth schemes across 5 to 30 m.
- Our system achieves a PER of 4% and 7% in the presence of slow and fast motion. Further, it adapts its frequency band to operate reliably with different phone orientations.

Contributions. The last few decades have shown that software can bring technology to the masses more rapidly than custom hardware. We present the first acoustic-based system that enables underwater messaging on commodity mobile devices, using only software. To this end, we designed a communication system that in real-time adapts to variations in frequency responses across mobile devices, changes in multipath due to mobility and SNR variations from severe frequency diversity, to minimize packet error rate. We evaluated our system in underwater environments and built a messaging app that allows users to send messages and SoS beacons. Finally, by making our system code publicly available at publication, we believe that this work has the potential to make underwater communication technology accessible to everyone with a mobile device, by just downloading software.

2 SYSTEM DESIGN

We first characterize the properties of acoustic signals from mobile devices in water. We then present our real-time protocol as well as our data encoding and decoding algorithms. We open-source our code and data at <https://underwatermessaging.cs.washington.edu/>.

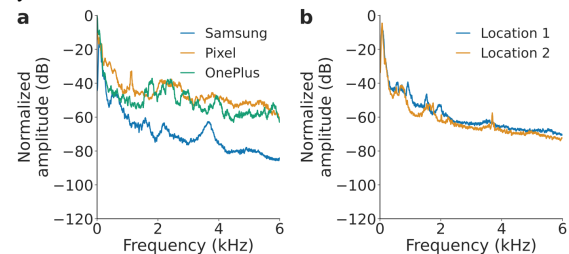


Figure 4: Underwater ambient noise measurements. Normalization is done across plots in the same graph.

2.1 Characterizing mobile devices in water

We characterize the acoustic properties of commodity smart devices (e.g., phones, watches) in underwater scenarios.

Frequency selectivity. As the speakers and microphones differ between smart devices, the frequency selectivity of an acoustic signal can vary between different transmitting and receiving phones. To evaluate this, we selected four different smart devices: Samsung Galaxy S9, Google Pixel 4, OnePlus 8 Pro, and Samsung Galaxy Watch 4. We placed each device pair underwater 5 m away from each other. We placed each smartphone in a waterproof pouch [22] and submerged the device to a depth of 1 m in a 2 m deep lake. The transmitter sends a 1–5 kHz chirp with a duration of 500 ms. Fig. 3a shows that the frequency response varies between device pairs. It is uneven and exhibits deep notches with the frequencies where notches occur varying across device. The plot also shows that the frequency response diminished above 4 kHz, which suggests that acoustic communication above this frequency on mobile devices may be challenging.

Next, we fix the transmitting and receiving smart devices to both be a Samsung Galaxy S9. We repeat the same experiment as before at a distance of 10 m. Fig. 3b shows the variation in frequency response as a result of the multipath characteristics at the different locations. Specifically, we observe that multipath causes the notches of the response to occur at different frequencies. Thus the frequencies ideal for underwater communication may vary with location.

Channel reciprocity. Next, we analyze the frequency response of the forward and backward channel in air and underwater. We use two smartphones of the same model (Samsung Galaxy S9) and measure the frequency response of a 1–3 kHz chirp with a 1 s duration. The first phone is set to send a chirp to the second phone, and two seconds later, the second phone sends a chirp back to

the first phone. We separate the phones by 2 m and perform the measurement over the air and underwater. In Fig. 3c we see that the frequency response of the chirp sent over the air is similar across both phones. However, the frequency response of the chirp underwater in Fig. 3d differs significantly. This suggests that the optimal frequencies for communication are different for the forward and backward channel. So an explicit feedback signal would need to be sent from the receiving device to the transmitting device.

Ambient noise. Finally, we measure underwater ambient noise as recorded on different smart devices for five seconds at the same location. Fig. 4a shows the ambient noise across these devices for different frequencies. Each noise profile is normalized to the maximum amplitude across all measured frequencies. The plot shows that across devices, the amplitude of noise is high below 1 kHz. Furthermore, noise can also be seen at higher frequencies up to 4.5 kHz. In our measurement location, underwater noise was due to the sound of water flowing and the movement of air bubbles. Additionally, the plot shows that the noise profiles vary across smartphones. We also measured the level of ambient noise at different locations using a Samsung Galaxy S9. Fig. 4b shows that the level of noise between 0–6 kHz can vary by 9 dB between locations. This is expected as the amount of water flowing or other acoustic interference will vary across locations. Across both these experiments, the high noise level below 1 kHz suggests that acoustic communication in that frequency range may be challenging. This suggests that we need to measure both signal strength and noise across frequencies to determine the right frequencies.

Design requirements. So our requirements are:

- *Works across smart devices with different frequency responses.* Our system should work across speakers and microphones on smart devices with different frequency responses.
- *Robust to multipath across different locations and distances.* Underwater environments can have challenging multipath due to reflections from the surface, floor and from the coast. Our system should be robust to these effects.
- *Tolerates mobility.* Strong water currents or waves can cause devices to drift away quickly or toward each other within a few seconds, resulting in Doppler shifts. Our system should be able to work in these real world environments.
- *Adapts to different noise profiles across environments.* The underwater environments can suffer from significant noise from ships, boats and animals. Loud sounds outside the water such as airplanes or helicopters also contribute to the noise.

2.2 Post-preamble feedback protocol

Fig. 5 shows our protocol that splits the packet into preamble/header and data. Alice first sends the preamble and the receiver ID. Alice remains silent for feedback from Bob but keeps its OFDM symbol timer ON. The preamble is composed of eight OFDM symbols from 1 to 4 kHz. When Bob detected this preamble, we first check the ID. It then runs our SNR estimation algorithm on the preamble for each subcarrier between 1-4 kHz, which it uses in our frequency band adaptation algorithm. Instead of a fine-grained adaptation for each bin, we select a contiguous band and send back only the start and end frequencies of this band to reduce overhead. Alice

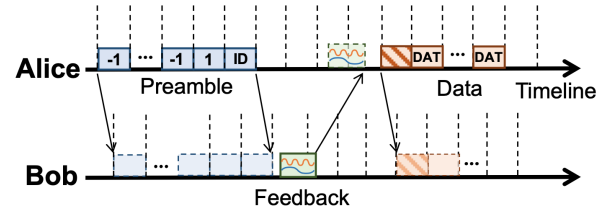


Figure 5: Protocol sequence diagram.

uses this information to encode bits within this selected frequency band and transmits data to Bob. To do this, Alice transmits the data symbols at the beginning of the next OFDM time interval as determined by its OFDM symbol clock. This ensures that the preamble synchronization performed by Bob can also be used for data. The first OFDM symbol in the data transmitted by Alice however is a known training symbol to track changes in channel since preamble transmission. In practice, the post-preamble silence period at Alice is assigned zero values to keep the speaker buffer full. This ensures that the symbol timing can be maintained for data symbols. Bob uses cross-correlation and energy detection in every OFDM symbol interval to detect the arrival of the first known data symbol from Alice. Bob can use the preamble to synchronize the data symbols since the propagation time for preamble and data symbols is similar. Note that safe human motion under water is usually lower than $1 - 2m/s$ [86] and the time interval between the preamble and data transmission in Alice side is several OFDM symbols (including feedback propagation, processing time). Assuming the interval is 5-symbol, the change in the propagation time is only 0.6% of the OFDM symbol duration.

2.2.1 Preamble design. The preamble has three purposes: packet detection, symbol synchronization, and channel estimation. For a real-time system, the preamble design and detection algorithm have two main requirements: detection robustness and low computational burden. Linear frequency modulation (LFM) signals and cross-correlation-based detection are proposed for underwater communication [58, 79, 88]. However, in our real-world experiments, LFM was not robust enough at long distances or with severe multipath. We instead use a data-aided approach [53] where we fill the OFDM bins with a CAZAC sequence. These sequences have a very good auto-correlation property and have unit Peak to Average Power Ratio (PAPR) [85]. Further, the good autocorrelation properties of a CAZAC sequence also makes it suitable to be used for channel estimation [90]. We concatenate eight such identical OFDM symbols and multiply each with a PN sequence with different signs $([-1, 1, 1, 1, 1, 1, -1, 1])$, to provide steeper fall off to the correlation timing metric and alleviate the side-lobes problem for a more robust detection and accurate synchronization [85].

Preamble detection and synchronization. Our detection algorithm is composed of two parts: coarse detection and fine-grained detection. The coarse detection algorithm applies cross-correlation between the received signal and preamble. In the presence of a preamble, this results in a correlation peak. However, the cross-correlation peak varies with SNR and spiky noise like underwater bubbles could also cause a very high correlation peak. To address this, we also use sliding correlation [53] where instead of calculating the correlation between the transmitted preamble and received

Algorithm 1: Frequency band selection algorithm

```

for  $L \leftarrow N_0$  to 1 do
  for  $m \leftarrow 0$  to  $N_0 - L$  do
     $SNR_{min} = \min_{m \leq k < m+L} \{SNR_k + \lambda \cdot 10\log_{10}(\frac{N_0}{L})\}$ 
    if  $SNR_{min} > \epsilon_{SNR}$  then
       $n \leftarrow m + L - 1$ 
      return  $(m, n)$ 

```

signal, we apply a sliding window on the received signal and divide the sliding windows into 8 segments each the length of an OFDM symbol. Then we multiply each segment by the PN sequence and calculate the correlation between the two nearby segments. Finally, we sum them up and divide the sum by the received energy within this sliding window. When the real preamble arrives, the sliding correlation would have a high peak (> 0.6). The peak height of this normalized sliding correlation is not sensitive to SNR changes. Moreover, spiky noise is unlikely to have this specific encoded data pattern and thus its sliding correlation value is low (< 0.2).

However, the sliding correlation method increases computational burden. So we perform the sliding correlation on the candidate signals only after the coarse detection. In addition, we increase the step size of sliding correlation to 8 to balance the computational burden and synchronization resolution. When a valid preamble is detected, we select the peak index in the sliding correlation curve as the beginning of the preamble for synchronization. Even if our preamble structure can improve the peak prominence in sliding correlation, a synchronization offset is still inevitable, which may affect OFDM data transmission quality [54]. As described later, we use time-domain equalization in our decoder and a cyclic prefix in our encoder to address this issue.

2.2.2 Frequency band adaptation algorithm. Our frequency band adaptation algorithm is composed of two main steps.

SNR estimation per frequency bin. We use the 8 OFDM symbols in the preamble to estimate the channel. We compute the SNR for each subcarrier by applying frequency-domain channel estimation. Specifically, we denote the transmitted data in subcarrier k of the eight preamble symbols by a vector $x(k)$ and the received data in subcarrier k of 8 training symbols by a vector $y(k)$. We apply a minimum mean square error (MMSE) estimator to compute the channel response $H(k)$ for each subcarrier k . We then compute the SNR in the k^{th} bin as, $SNR_k = 20\log_{10} \frac{\|H(k)x(k)\|_2}{\|y(k) - H(k)x(k)\|_2}$

Frequency band selection. Our goal is to find the optimal frequency range based on the SNR distribution between 1-4 kHz. The basic idea is that we drop the bin with the lowest SNR and reallocate power to the remaining bins until the SNR in all remaining bins surpass the preset SNR threshold. Say, there are totally N_0 bins between 1-4 kHz, and the estimated SNR in the k^{th} bin is SNR_k . When we only select bins between m and n , the power in the discarded bins could be re-allocated to the remaining bins. Hence, the SNR value in the remaining bins increases by $10\log_{10}(\frac{N_0}{L})$, where

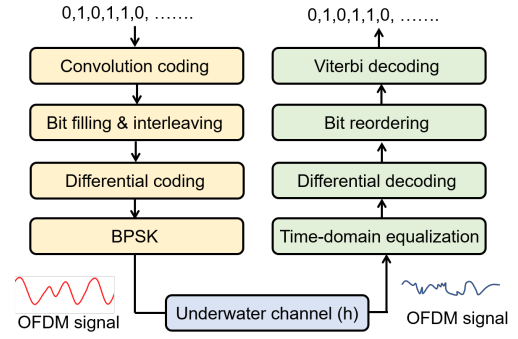


Figure 6: Data encoding and decoding.

$L = n - m + 1$. So, our optimization problem is as follows,

$$\begin{aligned} \max_{m,n} \quad & L = n - m + 1 \\ \text{s.t.} \quad & SNR_k + \lambda \cdot 10\log_{10}(\frac{N_0}{L}) > \epsilon_{SNR}, \forall k \in [m, n] \end{aligned}$$

where ϵ_{SNR} is the preset SNR threshold (in our implementation we set ϵ_{SNR} to 7). λ is a conservative factor between 0 to 1 (we select it empirically to 0.8), since in real-world implementation the power re-allocation is not exact. We set, ϵ_{SNR} and λ , a bit conservatively since we need to account for imperfect SNR estimation and because the channel may still change due to mobility. m and n from the above optimization give us f_{begin} and f_{end} .

2.2.3 Encoding feedback. We encode the frequency bins f_{begin} and f_{end} , in a single OFDM symbol by assigning all the power only to the two corresponding bins. By allocating all the power from the transmitter into two frequency bins, Alice can still decode the feedback signal reliably even when channel estimates of the backward path from Bob to Alice are unknown. Specifically, Alice can decode the signal by extracting the frequencies with the top-2 SNRs. The OFDM symbol from Bob would arrive at Alice after a delay of around a round-trip time. Since this specific OFDM symbol is effectively the same as transmitting two frequency tones, Alice performs an FFT over a sliding window with the same length as the OFDM symbol. The sliding window computation is performed up to the maximum round trip time corresponding to 30 m, to search for Bob's OFDM symbol.

2.3 Data transmission

Multi-path can be severe under water due to reflections from the surface, floor and other objects [74] causing inter-symbol interference (ISI). To address ISI without sacrificing bit rate, we apply time-domain MMSE equalization instead of increasing the cyclic prefix. Dynamic channel changes due to motion of humans, waves and other underwater objects is another challenge. Even within a packet, the channel for the first OFDM symbol may differ from the last symbol, which may deteriorate the equalizer's performance. Here, describe our encoding and decoding mechanisms to address these issues.

2.3.1 Data encoding. Alice encodes the transmitted bits and generates OFDM symbols between f_{begin} and f_{end} . The duration of each OFDM symbol is 960 samples (20 ms duration and 50 Hz

subcarrier spacing with 48000 kHz sampling rate). We also add a 67 sample cyclic prefix (6.9% overhead).

- *Convolutional coding.* We first apply 2/3 convolutional coding with constraint length $K = 7$ to encode data bits. These codes are widely used in GSM and satellite networks [28, 38].
- *Interleaving bits.* The next step is to assign the coded bits to OFDM symbols. Our algorithm has two main rules: (1) assign bits to OFDM bins between f_{begin} and f_{end} and set the remaining bins to zeros and (2) interleave the bits to avoid consecutive bit errors which could cause unresolved errors after convolutional decoding. Our empirical observations show that bit errors usually happen in the OFDM packets at a specific subcarrier or nearby two subcarriers. Our interleaving strategy is to first fill one symbol until all selected subcarriers in this symbol have bits, and then fill the next symbol to avoid consecutive bit errors in a single subcarrier. Within a symbol, after assign a bit in a subcarrier, we leave some subcarriers for the next bit with a step size is one-third of the selected bins. If we use less than three bins then this defaults to not using interleaving.
- *Differential coding.* In our system, we use differential encoding across consecutive OFDM symbols to alleviate the time-varying channel effects. Specifically, say the data in subcarrier k of symbol $i - 1$ is $y_{i-1}(k)$ and b is a coded bit intended for transmission. Then, we apply differential coding by setting the data in subcarrier k of the next symbol i as, $y_i(k) = y_{i-1}(k) \oplus b$. Thus, the bit is encoded as an XOR between two consecutive OFDM symbols. Since channel variations across two consecutive symbols is small, differential coding provides resilience to channel variations, as long as the coherence time is larger than one OFDM symbol.

After assigning the bits to the OFDM subcarriers, we demodulate the bits in each subcarrier using BPSK and use IFFTs to generate the time-domain OFDM signal.

2.3.2 Data decoding. To decode data, Bob first applies a 128-order FIR bandpass filter, with a passband of 1–4 kHz, on the received signal to filter out ambient noise.

- *MMSE Equalization.* Time domain equalization utilizes the equalizer coefficients to recover the transmitted signal and address inter-symbol interference (ISI). The equalizer coefficients is estimated using preset training symbols and the MMSE algorithm. The communication channel model can be written as $y = h * x + n$, where x is the transmitted signal, y is the received signal, h is the channel coefficient vector with length L , and $*$ is the convolution operator. The time-domain equalization model can be written as, $\hat{x} = g * y$, where \hat{x} is the recovered signal, y is the received signal, and g is the equalizer coefficient with length L . MMSE equalizers [69] minimize the mean square error between the transmitted signal and the recovered signal, i.e. $\min \|\hat{x} - x\|_2^2$. In our system, a known training symbol is appended to the front of the data symbols. We utilize this training symbol to estimate our MMSE equalizer with the channel length L of 480 samples. Finally, we apply the estimated equalizer to each received OFDM data symbol for transmitted signal recovery.
- *Differential & convolutional decoding.* After equalization, we apply an FFT on each symbol and acquire the data in the frequency bins. We calculate the phase difference of two consecutive symbols on each OFDM bin within the frequency range of our frequency band

adaptation algorithm. By measuring the phase difference, we can extract the coded bits, 0 or 1. We then re-interleave the bits following the order of the pre-determined interleaver. Finally, we use the Viterbi algorithm with constraint length 7. This provides maximum likelihood values for the data bits.

We note the following points about our design.

- *Doppler shifts.* A concern is that motion could also cause Doppler shifts, leading to inter-channel interference (ICI) within the OFDM symbol. Since our system mainly focuses on underwater human communication, the safe motion speed for humans during scuba diving is usually slower than 1 m/s [86]. Even if the transmitter and receiver are moving in opposite directions, their relative speed is 2 m/s. The typical underwater acoustic speed is 1500 m/s. This results in a Doppler shift of around 5 Hz at our maximum operational frequency. We however note that our OFDM subcarrier spacing is 50Hz, which is much higher than this Doppler shift and thus does not cause significant ICI.
- *Encoding ID and ACKs.* We use a single frequency in the OFDM symbol to encode device ID as well as the ACKs. Specifically, ACKs are encoded by assigning the OFDM bin corresponding to 1 kHz to denote successful packet reception. This mechanism is reliable and does not require a long preamble since it assigns all the transmit power to a single OFDM bin. We also encode the device ID using the different subcarriers in the OFDM symbol. We have a total of 60 subcarriers and when the device transmits the OFDM symbol encoding its ID, it allocated all the power to the corresponding subcarrier. This limits the number of devices in our local network to 60 users, which may be acceptable for underwater human activities like scuba diving and snorkeling. These IDs can be assigned using the app across devices in-air.

2.4 MAC protocol

In a typical use case, we expect our network to be operating in a low load scenario where not all transmitters send data at the same time. However, to support multiple devices that may operate at the same time, we also use carrier sense, similar to Wi-Fi, as a means of mediating access to the channel between multiple transmitters. We implement our carrier sense in realtime using energy detection by measuring the average energy level in the 1–4 kHz band, which is the frequency range used for communication in our system. We perform this measurement every 80 ms. Prior to transmission, each phone measures if the energy level on the channel exceeds a predefined threshold. If it detects that the energy level exceeds the threshold, the phone waits for a random backoff period that is defined in multiples of the packet duration. During this backoff period, the phone continues to listen to the channel. If it detects energy on the channel during this backoff period, it will increase the backoff time by the duration of one packet to ensure the backoff period will not elapse while a packet is being transmitted on the channel. After this additional time has elapsed, the phone will again check if the energy on the channel is below the threshold. After the remainder of the backoff time has elapsed and the channel is idle, the phone is then clear to send a packet. The threshold is computed by measuring the average noise level for a few seconds in each environment before use.

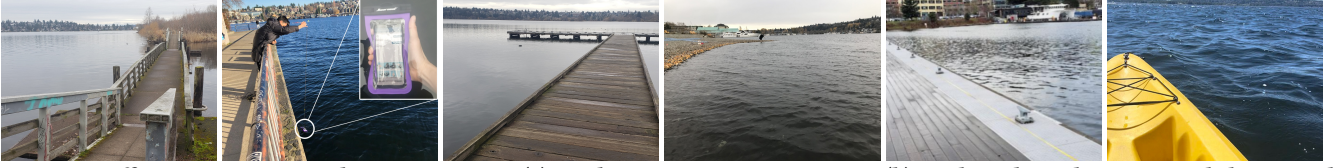


Figure 7: Different experimental environments. (a) Bridge: a quiet environment. (b) Park: A busy location with boats and kayaks passing by regularly. (c) Lake: a busy and noisy location by a fishing dock with animals and other water activity. (d) Beach: a waterfront with a length around 100 m for long range experiments. (e) Museum: a highly occupied location used to dock ships and boats with a depth of 9 m for depth experiments. (f) Bay: a location with 15 m depth for deeper water experiments.

We note two points: 1) In addition to energy detection, Wi-Fi receivers also use preamble detection as part of carrier sense, which we could also incorporate to improve noise resilience. 2) Our post-preamble feedback mechanism can be considered as a light-weight version of RTS-CTS. So we can add a preamble to Bob’s feedback which could be used in lieu of a CTS message to address the hidden terminal problem. Our current implementation does not include either features.

3 EVALUATION

Our communication system has been implemented to run on Android devices with preamble detection running continuously in real-time. The runtime to execute our channel estimation, frequency adaptation, and feedback decoding algorithm are each on average 1–2 ms on a Galaxy S9. Our decoder can perform equalization and Viterbi decoding for each symbol in less than 20 ms, which is the duration for an OFDM symbol. Fig. 2 shows our app interface where users can select from a list of 240 messages corresponding to hand signals used by professional divers. Users can filter the list of messages based on eight different categories. Additionally, the 20 most commonly used hand signals are displayed more prominently for selection by users. Pictorial representations of hand signals are included for reference in the app. The size of our data packet is 16 bits, 24 bits after applying a 2/3 convolutional code. With this, users can choose to send two hand signals in a single packet.

Our system also encodes a user’s 6-bit ID into an SoS beacon using frequency-shift keying. Specifically, we encode a 0 bit with a single frequency tone f_0 and a 1 bit with a single frequency tone f_1 . We design our system to support data rates of 5, 10, and 20 bps and use frequencies in 1.5–4 kHz to transmit these beacons over longer ranges. Using this scheme we may also encode an 8-bit hand signal, which can be transmitted in around a second at these rates.

We evaluated our system in four underwater environments with different multipath effects and noise levels (Fig 7).

- *Bridge*. Under the water of a bridge with a horizontal distance of 20 m. This is a quiet location with still waters.
- *Park*. By the waterfront of a park with a length of 40 m. This is a busy location with boats and strong currents.
- *Lake*. Fishing dock by lake with a 30 m length. The lake had a 5 m depth. This is a busy location with people fishing and kayaking.
- *Beach*. The length of the water here is around 100 m.
- *Museum*. This location has a depth of 9 m. This is a highly occupied location used as a dock for different boats and ships.

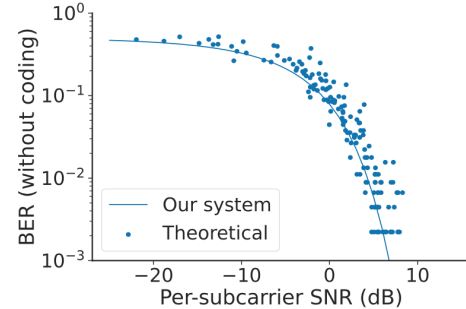


Figure 8: BER measurements are from distances of 5, 10, and 20 m with bandwidth of 1-4 kHz.

- *Bay*. This location has a depth of 15 m. There were a lot of waves at this location and experiments were performed on a kayak.

BER versus SNR. First, we evaluate our decoding algorithm by computing the BER at different subcarrier SNRs. Given the frequency fading nature of underwater links, BER can be different at each of the OFDM subcarriers depending on its SNR. In these experiments we use our real-time implementation on two Samsung Galaxy S9 phones. In all experiments, the phone speaker is set to its maximum volume. The phones were placed in a waterproof case and submerged at a depth of 1 m using a selfie stick as an extension pole. We perform the experiments at a distance of 5, 10, and 20 m at the bridge location. At each distance, we configure the transmitter to send a total of 500 OFDM symbols on the subcarriers between the 1–4 kHz and the modulation scheme for each subcarrier is set to BPSK. We computed the BER as the fraction of mistakenly decoded bits (without coding) to the total transmitted bits over 500 symbols. Fig. 8 plots the BER curve obtained in our experiments in comparison to the theoretical curve for BPSK. The plot shows that the empirical data for our design follows a similar trend to the theoretical estimates.

Effect of different environments. We evaluate our system in three locations: the bridge, park, and lake. These range from quiet and still to noisy and busy, and capture a diversity of environmental effects. We report the bitrate picked by our real-time algorithm and the PER of our system at 5 m.

During the experiment, our system runs in real-time where the transmitting smartphone first sends the preamble and header to the receiver. The receiver continuously listens for the preamble, when it detects it, performs SNR estimation on each OFDM subcarrier, runs our frequency band adaptation algorithm and sends back f_{begin} and f_{end} frequencies to the transmitting phone in a single OFDM

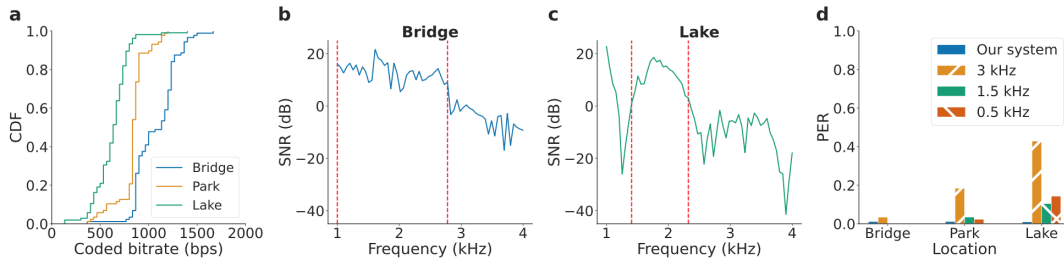


Figure 9: Effect of different environments. (a) CDF of bitrates selected by our algorithm. (b,c) Example frequency responses from two locations. Dashed lines indicate the range of selected frequencies by our algorithm. (d) A PER value computed using all packets at each location for our system and fixed bandwidth schemes.

symbol. The transmitting phone then sends data using OFDM subcarriers between f_{begin} and f_{end} . Each data packet contains 2 bytes of data (24 bits after applying a 2/3 convolutional code) on the selected OFDM frequency bins. This is more than sufficient to be able to encode the range of hand signals used by professional divers. We compare the performance of our frequency adaptation scheme to fixed-rate bandwidth schemes, where the transmitting phone sends the same bits using a fixed bandwidth of 1–4 kHz, 1–2.5 kHz and 1–1.5 kHz, which correspond to 60, 30, and 10 OFDM bins respectively. This procedure was set to repeat 100 times at each of the three locations tested. Every 25 packets, we would pause measurements on the smartphone, retrieve it from the water, and submerge it again, until all 100 packets were transmitted. Even if one bit error occurs at the decoder output, we mark it as an erroneous packet. Fig. 9a shows that the selected bit rate varies across location as well as runs. Within the same location, the selected bitrate changes as the multipath changes with time. The average selected bitrate was highest at the bridge location which is likely because the environmental noise and water currents are lowest in this setting. In comparison, at the park and lake environments, people fishing and kayaking nearby may have affected the channel, resulting in a selection of lower bitrates. Fig. 9b,c shows an example received frequency spectrum in the bridge and lake locations. The dashed lines indicate the range of frequencies selected by our algorithm for sending data. In contrast to the bridge location, the spectrum at the lake location exhibit more frequent and deeper dips. In the lake location, the wall and pillars underwater would reflect the acoustic signals resulting in more frequency selectivity.

Fig. 9d plots the PER obtained for our system, as well as the three fixed bandwidth schemes at the three environments. The plot shows that the PER for the fixed bandwidth schemes increases in response to larger multipath in the environment. The PER is larger in the lake environment which exhibits the highest variability in the received spectrum. In contrast, the PER of our system remains low across all three locations with an average value of 1%. The PER obtained by our system is also lower than those obtained by the fixed bandwidth schemes at the park and lake location.

Effect of different depths. Multipath can change at different depths — close to the surface, multipath interference from the surface of the water can be strong. Similarly, close to the bottom of the water body, it may experience significant multipath from the floor. We perform our experiments in the museum location, which

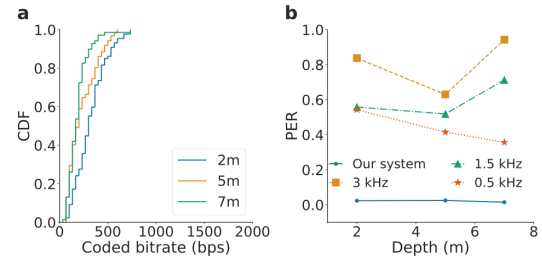


Figure 10: Effect of depth. (a) CDF of selected coded bitrate. (b) PER our system compared to fixed bandwidth schemes. A PER value is computed at each depth location using all packets sent at the location.

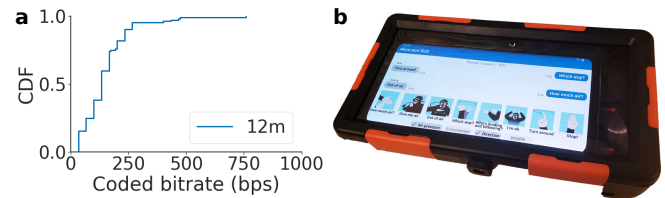


Figure 11: Testing in deeper waters. (a) CDF of selected coded bitrates. (b) Waterproof casing rated for a depth of 15 m.

had a total depth of 9 m. Our experiments were performed at a fixed horizontal distance of 5 m between the two smartphones.

Fig. 10a shows the bitrates selected by our system for three depths of 2, 5 and 7 m and Fig. 10b shows the PER. The PER for our system and the 0.5 kHz fixed bandwidth scheme was highest at a depth of 2 m when the phones are close to the water surface. For the 1.5 and 3 kHz fixed bandwidth scheme, the PER was highest at a depth of 7 m when the phones are close to the bottom of the lake. These results suggest that the environments at a depth of 2 and 7 m are the most challenging multipath environments. This is likely because there are more objects for the signal to reflect from at the surface and bottom of the lake including ducks, fish, and kayaks. At all depths, our system obtained significantly lower PERs than the fixed bandwidth schemes.

Testing in deeper waters. To evaluate our system at deeper depths, we performed an experiment at the bay location with a total depth of 15 m and submerged the phones to a depth of around 12 m. We note that 12 m is in the range of depths associated with basic recreational scuba dives [20]. In this experiment, we used a different waterproof pouch [12] (Fig. 11b) which was rated to work at a

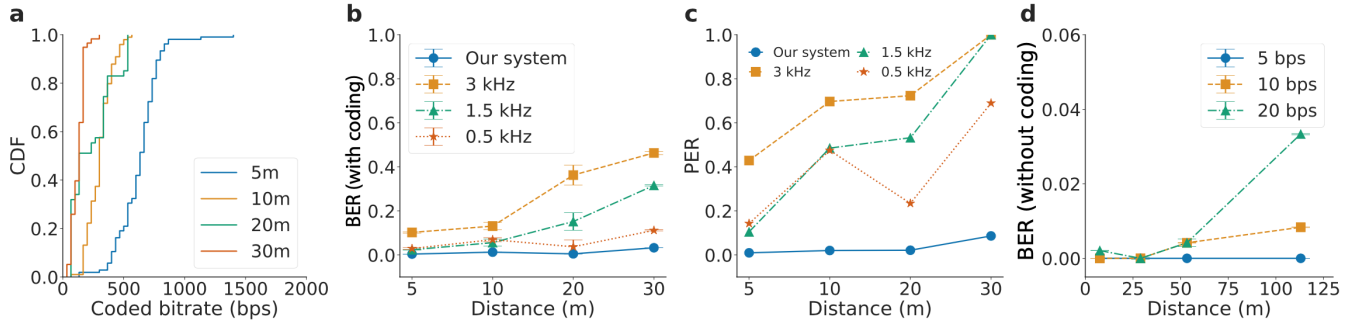


Figure 12: Range evaluation. (a) CDF of selected bitrates. (b,c) BER and PER of our system versus fixed bandwidth schemes. A PER value is computed from all packets at the location. (d) Longer ranges using lower bit rates.

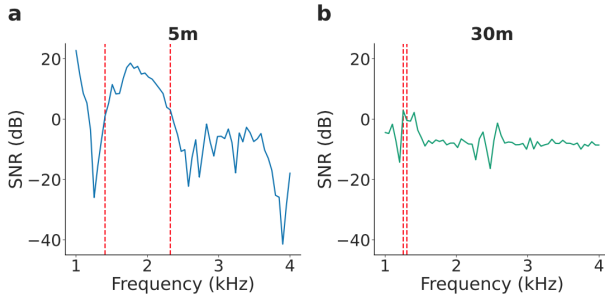


Figure 13: (a,b) Our system uses a smaller band in response to increased attenuation at larger distances.

maximum depth of 15 m. We note that this pouch was made of a hard plastic casing (polycarbonate and thermoplastic polyurethane) which attenuated the sound more than the thin flexible plastic material (polyvinyl chloride) used in the pouch for other experiments. We positioned the transmitter and receiver phone on either side of a two-person kayak (3.5 m length), and weighed down the phones using a pouch of sand to ensure the phones would submerge underwater. Fig. 11a shows that the median bitrate selected by our system at a depth of 12 m was 133 bps. This suggests the phones are able to communicate even at these deeper depths and with a hard plastic casing.

Range evaluation. We perform our range evaluation in the lake location since it had a comparatively long distance of 30 m. We performed our measurements at a depth of 1 m, which as we note in the previous evaluation, has empirically more challenging multipath at the lake location. The phones were submerged into the water using a rope, which caused the phone to sway and rotate slowly during measurements, i.e., the phones were not static.

Fig. 12a shows the bit rate after coding selected by our system at different distances. The plot shows that the selected bitrate generally decreases with distance, with the largest drop occurring between 5 and 10 m. At a distance of 5 m and 30 m, the median selected bitrate was 633.3 and 133.3 bps respectively. In Fig. 12b we also show the BERs for the coded bits transmitted at different distances for our frequency adaptation scheme and the fixed bandwidth schemes. The BERs for the fixed adaptation schemes increase quickly with distance. This is because the fixed schemes will continue to naively transmit bits on low SNR subcarriers and increase the likelihood of bit errors. This is clearer in Fig. 12c where we plot the PERs for these schemes over distance. The plots show that the

PER for the fixed adaptation schemes reach 100% when using a fixed bandwidth of 1.5 and 3 kHz. In contrast, our frequency adaptation scheme has a PER of 7% at a distance of 30 m. This demonstrates that to minimize PER it is essential to pick the appropriate frequency band since even with a fixed low bandwidth of 0.5 kHz, it is likely that some of the frequencies in this narrowband signal are in a deep fade resulting in sustained packet losses despite using coding. In contrast, our real-time adaptive system picks a conservative set of frequencies depending on the frequency profile and SNRs as shown in Fig. 13 which allows it to minimize the packet error rate.

In the above design, each OFDM symbol had a 20 ms duration, limiting the minimum symbol rate to 50 bps. To further reduce the bit rate, we increase the symbol duration to 50, 100 and 200 ms and use a single frequency within each symbol to encode data. This results in a bit rate of 20, 10 and 5 bps respectively, which may result in a longer range. To evaluate this, we perform our experiments in the beach location. Testing was performed at a fixed depth of 1 m. We compute BER as the number of bits that were correctly decoded over all the bits transmitted at each location at each of the three bit rates. Fig. 12d shows the uncoded BER of our system up to a distance of 113 m. The plot shows that the BER is less than 1% for bitrates of 5 and 10 bps up to the maximum tested distance of 113 m. This demonstrates that our system enables communication between smartphones underwater at long ranges albeit at lower bit rates. We also note that a bit rate of 10 bps is sufficient to transmit SoS beacons which would be important at these long ranges during underwater activities. Similarly transmitting a 8 bit packet that is sufficient to encode the 200 hand signals can also be done in around a second after accounting for coding.

Effect of mobility. We perform mobility evaluation in the lake location at a fixed horizontal distance of 5 m and at a depth of 1 m. In this experiment, we measure system performance when the phones are static, and when the transmitting phone is moving slowly and moving quickly. To do this, we move the phone horizontally back and forth, and vertically up and down. Since the phone is attached to the rope, the phones will also rotate randomly during the movement. During these measurements, the raw accelerometer readings on the phone after compensating for gravity were on average, 2.5 and $5.1m/s^2$ for slow and fast motion, respectively.

In Fig. 14c, we calculate the uncoded BER with and without differential coding. Specifically, for received packets without differential coding, we only apply the equalizer to each OFDM symbol and compare the decoded bits with the original bits. The plot shows

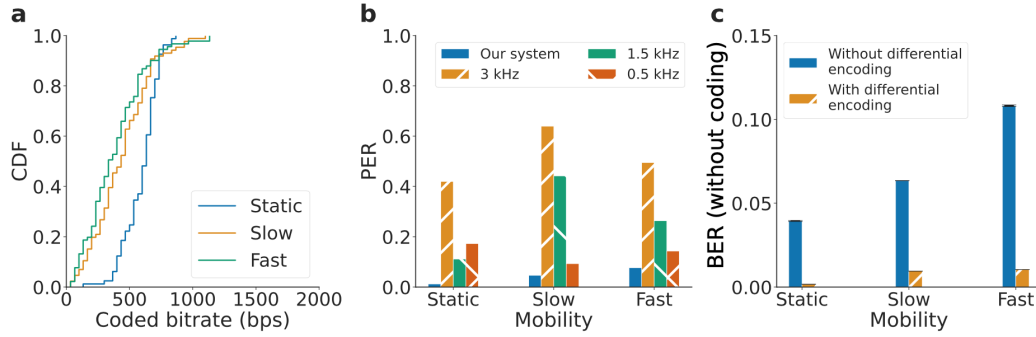


Figure 14: Effect of mobility. (a) CDF of selected bitrates in a static scenario versus one of the smartphones moving slow ($2.5m/s^2$) and fast ($5.1m/s^2$). (b,c) PER and uncoded BER of our system versus other configurations.

that with mobility, the BER without differential coding increases rapidly and easily surpass 10%. When we apply differential coding, the BER decreases dramatically and could still keep around 1% in the fast motion scenario. Fig. 14a shows the CDF of the coded bitrate across these three mobility scenarios. We see that the coded bitrate is highest in the static situation with a median bitrate of 640 bps. In contrast, the median bitrates in the slow and fast moving case decrease to 433 and 336 bps respectively. Fig. 14b shows that our PER increases from 1.2 to 7.6% as mobility increases.

Effect of phone orientation. For this evaluation, we ran our system at the bridge environment with a fixed distance of 5 m and a depth of 1 m. The phones were submerged with the screen facing towards the water surface, and were first set so the speaker and microphones of each phone are directly facing each other. We rotate one phone in the azimuth angle from 0 to 180° in increments of 45° . Fig. 15a shows that the median bitrate decreases with phone orientation from 1067 bps at 0° to 567 bps at 180° . Fig. 15b shows that while the fixed bandwidth schemes have a higher PER at large angles, our frequency adaptation scheme achieves a low PER. This is because our system is able to select a better frequency band and adapt to the channel at different orientations.

Channel stability and SNR. We take a closer look at the effects of mobility from the perspective of channel stability and SNR. Mobility has two effects on our system: (1) the channel may change between the preamble and the data symbols, leading to a different bandwidth selection and, (2) the channel between the first and the last data symbol may be different. For the second effect, as shown in Fig. 14(c), differential coding addresses the channel changes within the data packet. Here, we investigate the stability of the underwater channel between the preamble and data symbols in different motion scenarios. In our system, Alice first transmits a preamble to Bob and Bob selects the bandwidth. To verify the channel stability and its effects on SNR, instead of transmitting the data symbols, we configure Alice to transmit another preamble; Bob used the bandwidth from the first preamble to compute the SNRs of the corresponding OFDM bins using the second preamble. Now, we can evaluate the effect of channel stability on our bandwidth selection algorithm. Specifically, when the SNRs in some of the subcarriers within the selected bandwidth are very low, packet errors can happen. Hence, we select the minimum SNR computed using the second preamble within the selected bandwidth as the metric to evaluate the performance of our bandwidth selection algorithm.

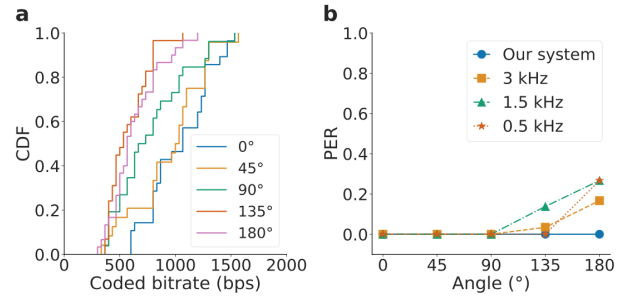


Figure 15: Effect of phone orientation. (a) CDF of selected bitrates for different azimuth offsets between a pair of smartphones. (b) PER at a distance of 5 m for our system and other fixed frequency band schemes.

We performed this experiment at a horizontal distance of 10 m at the lake location when both phones were held static, and when moving them at slow and fast speeds similar to our mobility evaluations. The x-axis in Fig. 16 represents the experiment index number, and the y-axis is the minimum SNR computed using the second preamble within the selected bandwidth. The dashed SNR line of 4 dB is the reference for subcarrier quality evaluation (4 dB may cause 1% BER according to the SNR-BER curve).

In the static scenario, the minimum SNRs are high, likely due to the conservative parameter settings used in our bandwidth adaptive algorithm. In the slow and fast motion scenario, the fluctuation of the minimum SNR values increase and sometimes the bad subcarriers appear in the bandwidth selection algorithm, leading to an increasing PER during fast motion. Despite the existence of channel variance with motion, two components in our system help alleviate the effects of mobility and keep the PER low: (1) the conservative parameter settings of our bandwidth adaptive algorithm with a high SNR threshold and conservative factor λ allows for some room for safe bandwidth selection albeit the inaccurate SNR estimation due to mobility. Due to these conservative settings, only a small proportion of packets would pick up the bad subcarrier (<4 dB). (2) In scenarios where our conservative parameters can fail when the change of channel is very large, convolution coding helps reduce the bit errors. However, when channel drastically changes leading to more deep drops in the selected bandwidth, packet error can occur, which accounts for the increasing PER during fast motion. In

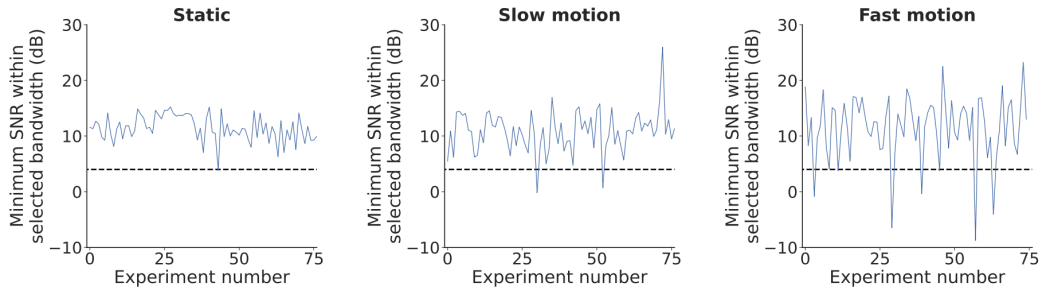


Figure 16: Channel stability and SNR. For consecutively transmitted pairs of preamble signals separated by the feedback duration, the figure shows the minimum SNR computed using the second preamble over the bandwidth picked from the first preamble. We run these experiments in (a) static scenarios and with (b) slow and (c) fast motion. The dashed line at 4 dB marks the SNR threshold at which BER will theoretically be at 1%, representing the threshold for a stable communication link.

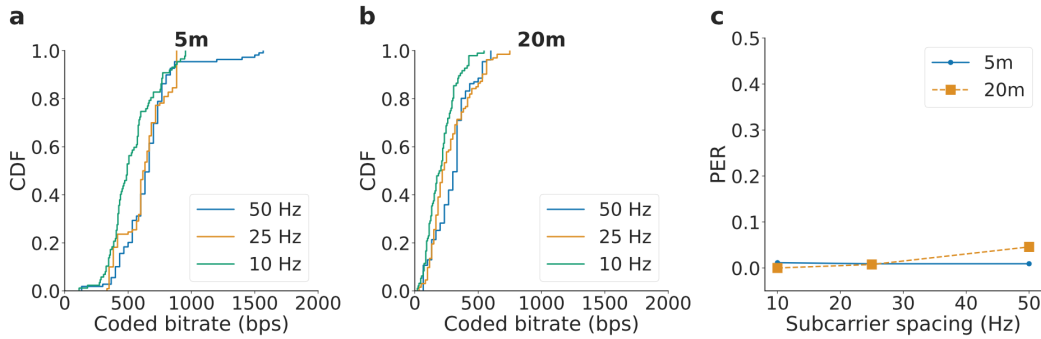


Figure 17: Effect of OFDM subcarrier spacing. CDF of selected bitrates for different subcarrier spacing values at a distance of (a) 5 and (b) 20 m. (c) PER for our system at different subcarrier spacings.

future work, further improvements can be used to make the bandwidth selection algorithm to be even more conservative, which can lead to lower data rates. Another approach to improving the PER is to apply non-linear equalization techniques, which perform better for signal recovery in poor channels with severe distortion [57].

Effect of OFDM subcarrier spacing. Next, we perform experiments with three different OFDM subcarrier spacing: 50 Hz (length of OFDM symbol is 20 ms), 25 Hz (40 ms), and 10 Hz (100 ms). We do not select the OFDM subcarrier spacing lower than 10 Hz due to Doppler shifts as described in §2.3. We perform these experiments in the lake location at a horizontal distance of 5 m and 20 m. During the experiment, for each subcarrier spacing, the OFDM symbol length in the preamble is kept the same as the length of data OFDM symbol to maintain the same frequency resolution. Fig. 17(c) shows that at 5 m, all the PERs are around 1%. At 20 m, the PER with 50 Hz spacing increases to 4.6%, while PERs for 25 Hz and 10 Hz spacings are lower than 1%. Smaller subcarrier spacing can improve PER because (1) smaller spacing can provide high-resolution SNR estimation and more accurate bandwidth selection and (2) smaller spacing can also improve the equalization performance due to the higher frequency-resolution in the training symbol.

Effect of air in water-proof case. To evaluate this, we first expelled as much air as we can from the water-proof case before putting the phones into the water and then measure the frequency response. We then filled the case with air before putting the phones in the water and then analyzed the frequency response. Fig. 18

shows that even if the frequency response of the two curves has some difference, the average power within 1-4 kHz was not significantly different.

Preamble & feedback signal evaluation. We also perform preamble evaluation in the lake location at a depth of 1 m. We transmit 180 preambles at each distance and evaluated the probability that our system could successfully detect the preamble. Our measurements show that our detection rate defined as the fraction of detected and transmitted preambles was 0.99, 1.0, 1.0 and 0.96 at 5, 10, 20 and 30 m. We also measured our system’s ability to correctly decode the feedback signal containing the result of our frequency adaptation algorithm at all the above distances. Frequency error rate is computed as the fraction of feedback signals where the decoded frequencies did not match the transmitted values. Across all tested distances, the error rate was around 0.01, i.e., 1 in 100 packets across all these distances. This is because we allocate all the power to these two frequencies making the signal strength much higher which allows reliable decoding even with frequency fading. In the cases where there are errors, the system confuses it to the adjacent OFDM bins.

MAC protocol evaluation. Finally, we measure the effectiveness of carrier sense at supporting multiple devices underwater. We consider two network deployments with four phones (three transmitters, and one receiver) and three phones (two transmitters, and one receiver) placed underwater in the bridge location at a depth of

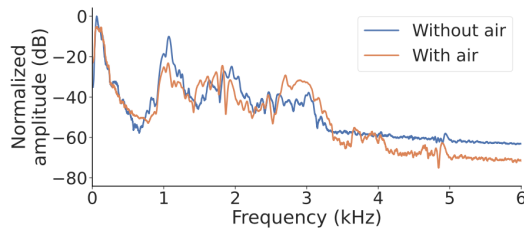


Figure 18: Effect of air in waterproof case.

1 m at distances in the range of 5-10 m from the receiver. As mentioned before, we expect that not all devices will be transmitting continuously underwater. But to test the effectiveness of carrier sense in a heavily used network, we configure the transmitters to send continuously after a random backoff period of multiple seconds, up to a maximum of 120 packets per transmitter. We repeat the experiments with and without carrier sense.

We measure the fraction of collisions for each of the transmitter. To calculate this fraction, we first logged the timestamps at when the transmitting phones sent a packet. Packets that were transmitted within one packet duration of each other was marked as a collision. The fraction of collisions in the network was defined as the number of packets that were involved in a collision divided by the total number of packets sent by all transmitters (dotted horizontal lines). Fig. 19 shows that without carrier sense, the collision rate in the three transmitter network is high with an average of 53%, however, our carrier sense mechanism is able to reduce this to an average of 7%. The figure shows a similar collision reduction for the two transmitter network from 33% to 5%.

4 RELATED WORK

Underwater networking is an active research area where topics across the network stack continue to be explored for custom acoustic modems and sensor networks [24]. There is also prior work in localization and ranging [35, 50], which is not in scope of our work. Below we describe the prior work in more detail.

Acoustic modem hardware. Since acoustic signals propagate well underwater, unlike RF, the common approach is to design custom acoustic modem hardware and use custom amplifiers at high power to achieve long ranges. Given the lack of economies of scale, much of this hardware can be expensive or not as accessible as in-air radios. All prior works that present underwater network simulators [32–34] or protocol stacks [29, 61, 63] use custom hardware with amplifiers and (de)modulators. Recent work integrates a custom acoustic modem with software radios (e.g., USRPs) to design a software-defined platform for underwater networking research [55, 72]. The closest to our work is iSonar [66] that designs a custom acoustic OFDM modem hardware that can connect to the smartphone using its audio jack. In addition to requiring additional hardware, it neither supports band adaptation nor is evaluated with mobility. In contrast, we design the first underwater acoustic communication system that operates on mobile devices without any additional hardware.

Modulation, rate adaptation and MAC protocols. Prior work makes contributions at the physical and MAC layers. [70, 73, 77] have analyzed various modulation techniques. There has also been work on underwater channel estimation for OFDM [56, 82], the use

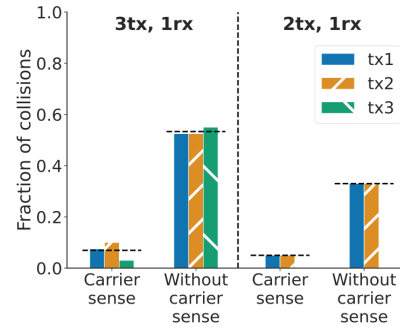


Figure 19: MAC protocol with multiple transmitters.

of pilot symbols to track channel estimates within a packet [44, 76], Doppler estimation [39, 49, 81] and bit rate adaptation [25, 59, 67, 71, 75, 78] for underwater sensor networks. Various hardware MAC protocols [24] have also been explored in prior work. There have also been interest in cognitive underwater spectrum access [9, 26, 89]. All this existing work has been designed for custom underwater hardware. In contrast, we design adaptation algorithms and protocols that can run on commodity mobile devices and operate with a large diversity of frequency responses across hardware.

Underwater sensor networks/IoT. Underwater sensor networks have been an active research area and uses custom sensor hardware [60, 87]. Recent work has designed novel IoT hardware that uses backscatter communication [41, 47] and acoustic energy harvesting to design battery-free underwater wireless sensors [37, 40]. Recent work has also designed creative hardware to communicate across the water and air interface [30, 80]. Optical approaches have also been proposed for short-range underwater communication at 1–2 m [48]. We build on this sensing work but focus instead of using commodity devices to enable underwater communication capabilities without additional hardware. Further, in contrast to prior work that is designed for underwater sensors, our goal is to enable humans to message underwater using their smart devices.

In-air acoustic communication. Prior work has used smartphones to enable acoustic communication and tracking in air at close ranges [45, 51, 52, 68, 83, 84]. However enabling underwater communication using smartphones is challenging for multiple reasons, 1) multipath in underwater environments can be more severe in comparison to in air, 2) even the frequencies that can be used on the forward and backward paths can be different underwater (see §2.1), 3) in-air acoustic systems have a limited communication range of a few meters. Designing an underwater communication protocol that can adapt for a wide range of bit rates and achieve much longer range requires designing a different system.

5 DISCUSSION

We discuss various system-level aspects of our system.

Battery life. To measure the power consumption of our system, we considered a setup with two Samsung Galaxy S9 smartphones as the transmitter and the receiver. We ran our system continuously at maximum sound volume with the screen on and found that the battery power reduced by 32% after a duration of 4 hours. This is sufficient for the application of recreational scuba diving, which have a maximum recommended dive time less than this [16, 17].

Messaging latency. A question that the reader may have is if the bit rates in our design are sufficient for underwater messaging. Since typing underwater may be uncomfortable, our app interface has the user select from one of 240 messages, which translates to around 8 bits (and 12 bits after coding). It takes close to half a second to send this message at 25 bps. At 1 kbps, we can even send a 50 character message in half a second.

Range versus power. The range achieved by custom hydrophones depends on their transmit power. Commercial hydrophones that have a kilometer or more range typically have a high transmit power [37]. The 30 m range for messaging and 100 m range for SoS beacons that we achieve with mobile devices is sufficient for many recreational and professional underwater activities.

Audibility. Underwater acoustic modems use a range of frequencies from multiple kHz to 100s of kHz. Our results show that 1–4 kHz is the optimal set of frequencies for use on commodity smart phones and watches. These are in the audible range of human hearing, which is also true for some of the commercial modems that operate in 7–17 kHz [5, 27]. Also note that human hearing underwater occurs through bone conduction rather than through the air pocket in the ear canal [64]. As a result, some prior studies observe that human hearing extends to ultrasonic frequencies underwater [2, 64]. While fish and sharks have limited sensitivity for frequencies far above 10 kHz [31, 62], sea mammals, e.g., dolphins, seals, and whales are highly sensitive to frequencies up to 150 kHz [42, 43]. Transmissions from every acoustic modem are assumed to be audible to sea mammals in the vicinity [70].

6 CONCLUSION

The last few decades have shown that software-based solutions can transform industries and bring technology to the masses more rapidly than custom hardware. We present the first acoustic system that uses software to bring underwater communication capabilities to commodity mobile devices. We believe that since our system can be downloaded as a mobile software app on billions of devices, it can help democratize underwater communication and has the potential to be used by tens of millions of people who participate in underwater activities like scuba divers and snorkeling every year.

Acknowledgments. The researchers are funded by the Moore Inventor Fellow award #10617 and the National Science Foundation. We thank our shepherd, Haitham Hassanieh, and the anonymous reviewers for their feedback on our submission.

This work does not raise any ethical issues.

REFERENCES

- [1] 2008. Underwater SMS for Divers, UDI Digital Texting. (2008). <https://www.trendhunter.com/trends/utc-udi>
- [2] 2016. Humans Can Hear Much Higher Pitched Sounds Underwater. *National Institute on Deafness and Other Communication Disorders* (2016). <https://www.noisyplanet.nidcd.nih.gov/have-you-heard/humans-hear-high-pitches-under-water>
- [3] 2016. Underwater Radio, Anyone?, DARPA. <https://www.darpa.mil/news-events/2016-12-16>. (2016).
- [4] 2018. 107 Hand signals for scuba divers. (2018). <https://www.youtube.com/watch?v=sVTRbLDtZaE>
- [5] 2018. Evo Logics 7/17 communication and positioning devices. <https://evologics.de/acoustic-modem/7-17>. (2018).
- [6] 2019. Can I snorkel with my apple 4 watch? <https://discussions.apple.com/thread/250139566>. (2019).
- [7] 2020. Magneto-inductive charging and communication in electronic devices, Patent US20210013740A1. <https://patents.google.com/patent/US20210013740A1/>. (2020).
- [8] 2021. Apple Reveals R&D into allowing iPhones to communicate in Underwater and Underground Environments. <https://www.patentlyapple.com/patently-apple/2021/01/apple-reveals-rd-into-allowing-iphones-to-communicate-in-underwater-and-underground-environments.html>. (2021).
- [9] 2021. An underwater cognitive acoustic network strategy for efficient spectrum utilization. *Applied Acoustics* 175 (2021), 107861.
- [10] 2022. 7 Best Snorkeling Watches And Smartwatches. <https://wearholc.com/best-snorkeling-watches-smartwatch/>. (2022).
- [11] 2022. Best Waterproof Phone Cases for Smartphone Photographers. (2022). <https://www.mobiography.net/gear/best-waterproof-phone-cases/>
- [12] 2022. Diving Phone Case for iPhone Samsung, Professional Underwater Photography Housings Case with Lanyard[50ft/15m], Diving Case for iPhone 13 Mini/12 Mini/12/12 Pro. (2022). https://www.amazon.com/gp/product/B086PQQVN3/ref=ppx_yo_dt_b_search_asin_title?ie=UTF8&th=1
- [13] 2022. Diving Phone Case for iPhone Samsung, Professional Underwater Photography Housings Case with Lanyard[50ft/15m], Diving Case for iPhone 13 Mini/12 Mini/12/12 Pro Etc. <https://www.amazon.com/Samsung-Professional-Underwater-Photography-Housings/dp/B086PQQVN3/>. (2022).
- [14] 2022. Fast Facts on Recreational Scuba Diving and Snorkeling <https://www.dema.org/store/download.aspx?id=7811B097-8882-4707-A160-F999B49614B6>. (2022).
- [15] 2022. Global Scuba Diving Equipment Market - Growth, Trends, COVID-19 Impact, and Forecasts (2022 - 2027), Mordor Intelligence. (2022).
- [16] 2022. How Long Can You SCUBA Dive? (2022). <https://www.wetsuitwarehouse.com/blog/how-long-can-you-scuba-dive/>
- [17] 2022. PADI Recreational Dive Table Planner. (2022). <https://www.a1scubadiving.com/wp-content/uploads/2018/06/PADI-Recreational-Dive-Table-Planner.pdf>
- [18] 2022. Scuba Diving Hand Signals. <https://blog.padi.com/scuba-diving-hand-signals/>. (2022).
- [19] 2022. Scuba Hand Signals. <https://lahainadivers.com/maui-scuba-hand-signals/>. (2022).
- [20] 2022. Scuba Q & A: Common Questions Asked By Nondivers. (2022). <https://dtmag.com/thelibrary/scuba-q-a-common-questions-asked-by-nondivers/>
- [21] 2022. Underwater Smartphone SeaLife Scuba Case – Waterproof Photography, Access Camera Controls, Leak Alarms (Without Light). (2022). <https://www.amazon.com/Underwater-iPhone-SeaLife-Scuba-Case/dp/B08KFGPWMT?th=1>
- [22] 2022. Universal Waterproof Case, Waterproof Phone Pouch Compatible for iPhone 13 12 11 Pro Max XS Max XR X 8 7 Samsung Galaxy s10/s9 Google Pixel 2 HTC Up to 7.0", IPX8 Cellphone Dry Bag -2 Pack. (2022). https://www.amazon.com/gp/product/B08S3SG5KF/ref=ppx_yo_dt_b_asin_title_o02_s00?ie=UTF8&psc=1
- [23] 2022. What is Ocean Exploration and why is it important? NOAA Office of Ocean Exploration and Research (2022). <https://oceanexplorer.noaa.gov/backmatter/whatisexploration.html>
- [24] Ahmed Al Guqhaiman, Oluwatobi Akanbi, Amer Aljaedi, and Chinghua Edward Chow. 2021. A Survey on MAC Protocol Approaches for Underwater Wireless Sensor Networks. *IEEE Sensors Journal* 21, 3 (2021), 3916–3932. <https://doi.org/10.1109/JSEN.2020.3024995>
- [25] M.S.M. Alamgir, Mst. Najnin Sultana, and Kyunghi Chang. 2020. Link Adaptation on an Underwater Communications Network Using Machine Learning Algorithms: Boosted Regression Tree Approach. *IEEE Access* 8 (2020), 73957–73971. <https://doi.org/10.1109/ACCESS.2020.2981973>
- [26] A. Ozan Bicen, A. Behzat Sahin, and Ozgur B. Akan. 2012. Spectrum-Aware Underwater Networks: Cognitive Acoustic Communications. *IEEE Vehicular Technology Magazine* 7, 2 (2012), 34–40. <https://doi.org/10.1109/MVT.2012.2190176>
- [27] Joe Borden and Jeffery DeArruda. 2012. Long range acoustic underwater communication with a compact AUV. In *2012 Oceans*. 1–5. <https://doi.org/10.1109/OCEANS.2012.6405091>
- [28] SA Butman, LJ Deutsch, and RL Miller. 1981. Performance of concatenated codes for deep space missions. *The Telecommunications and Data Acquisition Progress Report 42-63, March–April 1981* (1981), 33–39.
- [29] I. Calabrese, Riccardo Masiero, P. Casari, Lorenzo Vangelista, and Michele Zorzi. 2012. Embedded systems for prototyping underwater acoustic networks: The DESERT Underwater libraries on board the PandaBoard and NetDCU. *OCEANS 2012 MTS/IEEE: Harnessing the Power of the Ocean*, 1–8.
- [30] Charles J. Carver, Tian Zhao, H. Zhang, K. Odame, Alberto Quattrini Li, and Xia Zhou. 2020. Amphilight: Direct Air-Water Communication with Laser Light. In *NSDI 2020*.
- [31] Lucille Chapuis, Shaun Collin, Kara Yopak, Robert McCauley, Ryan Kempster, Laura Ryan, Carl Schmidt, Caroline Kerr, Enrico Gennari, Channing Egeberg, and Nathan Hart. 2019. The effect of underwater sounds on shark behaviour. *Scientific Reports* 9 (05 2019). <https://doi.org/10.1038/s41598-019-43078-w>
- [32] Mandar Chitre, Rohit Bhatnagar, and Wee-Seng Soh. 2014. U-netStack: An agent-based software stack and simulator for underwater networks. In *2014 Oceans - St. John's*. 1–10. <https://doi.org/10.1109/OCEANS.2014.7003044>

- [33] Emrecaan Demirors, Bharatwaj G. Shankar, G. Enrico Santagati, and Tommaso Melodia. 2015. SEANet: A Software-Defined Acoustic Networking Framework for Reconfigurable Underwater Networking. Association for Computing Machinery, New York, NY, USA. <https://doi.org/10.1145/2831296.2831316>
- [34] Emrecaan Demirors, Jiacheng Shi, Raffaele Guida, and Tommaso Melodia. 2016. SEANet G2: Toward a High-Data-Rate Software-Defined Underwater Acoustic Networking Platform. Association for Computing Machinery, New York, NY, USA. <https://doi.org/10.1145/2999504.3001112>
- [35] Katherine Domrese, Andrew Szajna, Shengli Zhou, and Jun-Hong Cui. 2014. Comparison of the Ranging Function of Three Types of Underwater Acoustic Modems. In *2014 IEEE 11th International Conference on Mobile Ad Hoc and Sensor Systems*. 743–748. <https://doi.org/10.1109/MASS.2014.36>
- [36] Yuhan Dong, Xuelong Mi, and Yiqing Zhou. 2016. Spatial Channel Model for Underwater Wireless Optical Communication Links: [Extended Abstract]. In *Proceedings of the 11th ACM International Conference on Underwater Networks & Systems (WUWNet '16)*. Association for Computing Machinery, New York, NY, USA, Article 18, 2 pages. <https://doi.org/10.1145/2999504.3001113>
- [37] Reza Ghaffarivardavagh, Sayed Saad Afzal, Osvy Rodriguez, and Fadel Adib. 2020. Ultra-Wideband Underwater Backscatter via Piezoelectric Metamaterials. In *Proceedings of the Annual Conference of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM '20)*. Association for Computing Machinery, New York, NY, USA, 722–734. <https://doi.org/10.1145/3387514.3405898>
- [38] Timo Halonen, Javier Romero, and Juan Melero. 2004. *GSM, GPRS and EDGE performance: evolution towards 3G/UMTS*. John Wiley & Sons.
- [39] Sung-Jun Hwang and Philip Schniter. 2007. Efficient Communication over Highly Spread Underwater Acoustic Channels. In *Proceedings of the Second Workshop on Underwater Networks (WuWNet '07)*. Association for Computing Machinery, New York, NY, USA, 11–18. <https://doi.org/10.1145/1287812.1287817>
- [40] Junsu Jang and Fadel Adib. 2019. Underwater Backscatter Networking. In *Proceedings of the ACM Special Interest Group on Data Communication (SIGCOMM '19)*. Association for Computing Machinery, New York, NY, USA, 187–199. <https://doi.org/10.1145/3341302.3342091>
- [41] Bryce Kellogg, Aaron Parks, Shyamnath Gollakota, Joshua R. Smith, and David Wetherall. 2014. Wi-Fi Backscatter: Internet Connectivity for RF-Powered Devices. *SIGCOMM Comput. Commun. Rev.* 44, 4 (aug 2014), 607–618. <https://doi.org/10.1145/2740070.2626319>
- [42] Darlene Ketten. 1994. Functional analyses of whale ears: Adaptations for underwater hearing. *Oceans Engineering for Today's Technology and Tomorrow's Preservation* 1, 1/264 – 1/270 vol.1. <https://doi.org/10.1109/OCEANS.1994.363871>
- [43] Darlene R. Ketten. 1992. The Marine Mammal Ear: Specializations for Aquatic Audition and Echolocation.
- [44] Anand Kumar and Prashant Kumar. 2020. Pilot-Assisted Maximum-Likelihood Estimation for Underwater Acoustic Communication. In *2020 5th International Conference on Computing, Communication and Security (ICCCS)*. 1–6.
- [45] Hyewon Lee, Tae Hyun Kim, Jun Won Choi, and Sunghyun Choi. 2015. Chirp signal-based aerial acoustic communication for smart devices. In *2015 IEEE Conference on Computer Communications (INFOCOM)*. 2407–2415. <https://doi.org/10.1109/INFOCOM.2015.7218629>
- [46] Chi Lin, Yongda Yu, Jie Xiong, Yichuan Zhang, Lei Wang, Guowei Wu, and Zhongxuan Luo. 2021. Shrimp: A Robust Underwater Visible Light Communication System. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*. Association for Computing Machinery, New York, NY, USA, 134–146. <https://doi-org.offcampus.lib.washington.edu/10.1145/3447993.3448616>
- [47] Vincent Liu, Aaron Parks, Vamsi Talla, Shyamnath Gollakota, David Wetherall, and Joshua R. Smith. 2013. Ambient Backscatter: Wireless Communication out of Thin Air. *SIGCOMM Comput. Commun. Rev.* 43, 4 (aug 2013), 39–50. <https://doi.org/10.1145/2534169.2486015>
- [48] Xinyang Liu, Xinhua Gao, Jiale Li, Yibo Wang, Chi Lin, Zhenquan Qin, and Lei Wang. 2021. UQCom: underwater safety communication based on 3D blue-green QR arrays. In *Proceedings of the SIGCOMM'21 Poster and Demo Sessions*. 36–38.
- [49] Sean Mason, Christian Berger, Shengli Zhou, Keenan Ball, Lee Freitag, and Peter Willett. 2009. An OFDM Design for Underwater Acoustic Channels with Doppler Spread. In *2009 IEEE 13th Digital Signal Processing Workshop and 5th IEEE Signal Processing Education Workshop*. 138–143. <https://doi.org/10.1109/DSP.2009.4785910>
- [50] Martin Monteiro and Arturo C Marti. 2020. Using smartphones as hydrophones: two experiments in underwater acoustics. *Physics Education* 55, 3 (Apr 2020), 033013. <https://doi.org/10.1088/1361-6552/ab8102>
- [51] Rajalakshmi Nandakumar, Krishna Kant Chintalapudi, Venkat Padmanabhan, and Ramarathnam Venkatesan. 2013. Dhvani: Secure Peer-to-Peer Acoustic NFC. In *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM (SIGCOMM '13)*. Association for Computing Machinery, New York, NY, USA, 63–74. <https://doi.org/10.1145/2486001.2486037>
- [52] Rajalakshmi Nandakumar, Vikram Iyer, Desney Tan, and Shyamnath Gollakota. 2016. FingerIO: Using Active Sonar for Fine-Grained Finger Tracking. Association for Computing Machinery, New York, NY, USA.
- [53] Ali A Nasir, Salman Durrani, and Rodney A Kennedy. 2010. Performance of coarse and fine timing synchronization in OFDM receivers. In *2010 2nd International Conference on Future Computer and Communication*, Vol. 2. IEEE, V2–412.
- [54] Leila Nasraoui. 2015. *Advanced Synchronization Techniques for OFDM Systems*. Ph.D. Dissertation. Université de Carthage.
- [55] Youngtae Noh, Dustin Torres, and Mario Gerla. 2015. Software-defined underwater acoustic networking platform and its applications. *Ad Hoc Networks* 34 (01 2015).
- [56] Donghong Ouyang, Yuzhou Li, and Zhizhan Wang. 2021. Channel Estimation for Underwater Acoustic OFDM Communications: An Image Super-Resolution Approach. *CoRR* abs/2103.04345 (2021).
- [57] Seema Paliwal, Dilpreet Kaur Grover, and Jyoti Krayla. 2016. Comparison of Linear and Non-Linear Equalizer using the Matlab. *Communications on Applied Electronics (CAE)* 4, 1 (2016).
- [58] Rothna Pec, Mohammed Saquib Khan, and Yong Soo Cho. 2017. An LFM-based preamble for underwater communication. In *2017 International Conference on Information and Communication Technology Convergence (ICTC)*. 1181–1183. <https://doi.org/10.1109/ICTC.2017.8190891>
- [59] Konstantinos Pelekanakis and Luca Cazzanti. 2018. On Adaptive Modulation for low SNR Underwater Acoustic Communications. In *OCEANS 2018 MTS/IEEE Charleston*. 1–6. <https://doi.org/10.1109/OCEANS.2018.8604521>
- [60] Zheng Peng, Zhong Zhou, Jun-Hong Cui, and Zhijie Jerry Shi. 2009. Aqua-Net: An underwater sensor network architecture: Design, implementation, and initial testing. In *OCEANS 2009*. 1–8. <https://doi.org/10.23919/OCEANS.2009.5422199>
- [61] Chiara Petrioli, Roberto Petrocchia, John R. Potter, and Daniele Spaccini. 2015. The SUNSET Framework for Simulation, Emulation and at-Sea Testing of Underwater Wireless Sensor Networks. *Ad Hoc Netw.* 34, C (nov 2015), 224–238. <https://doi.org/10.1016/j.adhoc.2014.08.012>
- [62] Arthur Popper. 2003. Effects of Anthropogenic Sounds on Fishes. *Fisheries* 28 (10 2003), 24–31. [https://doi.org/10.1577/1548-8446\(2003\)28\[24:EOASOF\]2.0.CO;2](https://doi.org/10.1577/1548-8446(2003)28[24:EOASOF]2.0.CO;2)
- [63] John Potter, João Alves, Thomas Furfaro, Arjan Vermeij, Nicolas Jourden, Diego Merani, Giovanni Zappa, and Alessandro Berni. 2014. Software Defined Open Architecture Modem development at CMRE. In *2014 Underwater Communications and Networking (UComms)*. 1–4. <https://doi.org/10.1109/UComms.2014.7017157>
- [64] Michael K. Qin, Derek Schwaller, Matthew Babina, and Edward A. Cudahy. 2011. Human underwater and bone conduction hearing in the sonic and ultrasonic range. *Journal of the Acoustical Society of America* 129 (2011), 2485–2485.
- [65] Hariharan Rahul, Farinaz Edalat, Dina Katabi, and Charles G. Sodini. 2009. Frequency-Aware Rate Adaptation and MAC Protocols. In *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking (MobiCom '09)*. Association for Computing Machinery, New York, NY, USA, 193–204.
- [66] Francesco Restuccia, Emrecaan Demirors, and Tommaso Melodia. 2017. Isonar: Software-Defined Underwater Acoustic Networking for Amphibious Smartphones. In *Proceedings of the International Conference on Underwater Networks & Systems (WUWNET'17)*. Association for Computing Machinery, New York, NY, USA, Article 15, 9 pages. <https://doi.org/10.1145/3148675.3148710>
- [67] Mohammad Sadeghi, Mohammed Elamassie, and Murat Uysal. 2017. Adaptive OFDM-based acoustic underwater transmission: System design and experimental verification. In *2017 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*. 1–5.
- [68] G. Enrico Santagati and Tommaso Melodia. 2015. U-Wear: Software-Defined Ultrasonic Networking for Wearable Devices. Association for Computing Machinery, New York, NY, USA.
- [69] Phil Schniter. 2008. MMSE Equalizer Design. (2008). https://www2.ece.ohio-state.edu/~schniter/ee501/handouts/mmse_eq.pdf
- [70] Dominik Schott, Andrea Gabbriellini, Wenxin Xiong, Georg Fischer, Fabian Höflinger, Johannes Wendeberg, Christian Schindelhauer, and Stefan Rupitsch. 2021. Asynchronous Chirp Slope Keying for Underwater Acoustic Communication. *Sensors* 21 (05 2021), 3282. <https://doi.org/10.3390/s21093282>
- [71] Satish Shankar and Mandar Chitre. 2013. Tuning an underwater communication link. In *2013 MTS/IEEE OCEANS - Bergen*. 1–9. <https://doi.org/10.1109/OCEANS-Bergen.2013.6607956>
- [72] George Sklivanitis, Emrecaan Demirors, Stella N. Batalama, Tommaso Melodia, and Dimitris A. Pados. 2014. Receiver configuration and testbed development for underwater cognitive channelization. In *2014 48th Asilomar Conference on Signals, Systems and Computers*. 1594–1598. <https://doi.org/10.1109/ACSSC.2014.7094734>
- [73] Milica Stojanovic. 2008. OFDM for underwater acoustic communications: Adaptive synchronization and sparse channel estimation. In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*. 5288–5291. <https://doi.org/10.1109/ICASSP.2008.4518853>
- [74] Milica Stojanovic. 2008. Underwater acoustic communications: Design considerations on the physical layer. In *2008 Fifth Annual Conference on Wireless on Demand Network Systems and Services*. IEEE, 1–10.
- [75] Yishan Su, Yibo Zhu, Haining Mo, Jun-Hong Cui, and Zhigang Jin. 2015. A Joint Power Control and Rate Adaptation MAC Protocol for Underwater Sensor Networks. *Ad Hoc Netw.* 26, C (mar 2015), 36–49. <https://doi.org/10.1016/j.adhoc.2014.10.014>

- [76] Tahsin Tabba. 2020. Pilot Assisted MIMO OFDM-IM Design for Doubly Selective Underwater Acoustic Systems.
- [77] Jun Tao, Liang An, Shuai Yao, Lin Zhou, Xiao Han, and Zhen Qin. 2018. Precoded OFDM over Underwater Acoustic Channels. In *Proceedings of the Thirteenth ACM International Conference on Underwater Networks & Systems (WUWNet '18)*. Association for Computing Machinery, New York, NY, USA, Article 4, 6 pages. <https://doi.org/10.1145/3291940.3291983>
- [78] Filipe Teixeira, Rui Campos, and Manuel Ricardo. 2015. IEEE 802.11 Rate Adaptation Algorithms in Underwater Environment. Association for Computing Machinery, New York, NY, USA.
- [79] Arjun Thottappilly. 2011. *OFDM for underwater acoustic communication*. Ph.D. Dissertation. Virginia Tech.
- [80] Francesco Tonolini and Fadel Adib. 2018. Networking across Boundaries: Enabling Wireless Communication through the Water-Air Interface. Association for Computing Machinery, New York, NY, USA.
- [81] Joël Trubuil and Thierry Chonavel. 2012. Accurate Doppler estimation for underwater acoustic communications. In *2012 Oceans - Yeosu*. 1–5. <https://doi.org/10.1109/OCEANS-Yeosu.2012.6263500>
- [82] Joël Trubuil, Thierry Le Gall, and Thierry Chonavel. 2014. Synchronization, Doppler and channel estimation for OFDM underwater acoustic communications. In *OCEANS 2014 - TAIPEI*. 1–6.
- [83] Anran Wang and Shyamnath Gollakota. 2019. MilliSonic: Pushing the Limits of Acoustic Motion Tracking. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–11. <https://doi.org/10.1145/3290605.3300248>
- [84] Qian Wang, Kui Ren, Man Zhou, Tao Lei, Dimitrios Koutsonikolas, and Lu Su. 2016. Messages behind the Sound: Real-Time Hidden Acoustic Signal Capture with Smartphones. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking (MobiCom '16)*. Association for Computing Machinery, New York, NY, USA, 29–41. <https://doi.org/10.1145/2973750.2973765>
- [85] Yang Wen, Wei Huang, and Zhongpei Zhang. 2006. CAZAC sequence and its application in LTE random access. In *2006 IEEE Information Theory Workshop-ITW'06 Chengdu*. IEEE, 544–547.
- [86] Magdalena Wojtków and Anna Nikodem. 2017. Biomechanics of diving: the influence of the swimming speed on the kinematics of lower limbs of professional divers. *Acta of bioengineering and biomechanics* 19, 4 (2017).
- [87] Yang Xiao. 2009. *Underwater Acoustic Sensor Networks* (1 ed.). Auerbach Publications, USA.
- [88] Yongjun Xie, Xiaoyi Hu, Jing Xiao, Deqing Wang, and Wangwei Lei. 2009. Implementation of timing synchronization for OFDM underwater communication system on FPGA. In *2009 3rd International Conference on Anti-counterfeiting, Security, and Identification in Communication*. IEEE, 568–570.
- [89] Wang Yonggang, Tang Jiansheng, Pan Yue, and Huangfu Li. 2008. Underwater communication goes cognitive. In *OCEANS 2008*. 1–4. <https://doi.org/10.1109/OCEANS.2008.5151898>
- [90] Yongzhao Zhang, Wei-Hsiang Huang, Chih-Yun Yang, Wen-Ping Wang, Yi-Chao Chen, Chuang-Wen You, Da-Yuan Huang, Guangtao Xue, and Jiadi Yu. 2020. Endophasia: Utilizing Acoustic-Based Imaging for Issuing Contact-Free Silent Speech Commands. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 1 (2020), 1–26.