Mobile Underwater Backscatter Networking

By

Purui Wang

S.B., Peking University (2021)

Submitted to the Department of Electrical Engineering and Computer Science in Partial Fulfillment of the Requirements for the Degree of

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2024

Authored by: Purui Wang
Department of Electrical Engineering and Computer Science
May 15, 2024

Certified by: Fadel Adib
Associate Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by: Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

Mobile Underwater Backscatter Networking

by

Purui Wang

Submitted to the Department of Electrical Engineering and Computer Science
on May 10, 2024 in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

ABSTRACT

Underwater backscatter is a recently introduced technology for ultra-low-power underwater networking. Despite advances in this technology, existing systems are limited to static environments and cannot operate reliably under mobility. This thesis presents EchoRider, the first system that enables reliable underwater backscatter networking under mobility. EchoRider's design introduces three new components. The first is a robust, chirp-based downlink protocol that brings the benefits of LoRa wireless networks to underwater backscatter, while accounting for the ultra-low-power nature of the backscatter sensor nodes. The second is a novel NACK-based backscatter retransmission algorithm, which enables reliable and efficient underwater backscatter. The third is a Doppler-resilient backscatter decoding pipeline on the uplink that features adaptive equalization, polar coding, and an equalizer retraining mechanism. We implemented an end-to-end prototype of EchoRider and compared it to a state-of-the-art baseline. Our evaluation across more than 1,200 real-world experimental trials in real-world environments demonstrates that EchoRider outperforms the state-of-the-art baseline by more than 160× in BER under mobility, and that it can sustain typical underwater goodput (around 0.5kbps) in scenarios where the baseline's goodput drops to zero at speeds as low as 0.1m/s. Finally, we demonstrate EchoRider in an example application involving an underwater mobile drone and a backscatter sensor node.

Keywords: Underwater Acoustic Backscatter, Mobility, Channel Estimation, Phase-Locked Loop (PLL), Automatic Repeat reQuest (ARQ)

Thesis supervisor: Fadel Adib
Title: Associate Professor of Media Arts and Sciences and Electrical Engineering and Computer Science

Acknowledgments

This journey would not have been possible without the unwavering support and guidance of many people who have made both my research and this dissertation a reality. First and foremost, I must extend my deepest gratitude to my research advisor, Prof. Fadel Adib, whose guidance through the complexities of my research topic has been invaluable. Not only has he been a mentor in the realms of academic inquiry and exploration, but his passion and diligence towards research have also served as a constant source of inspiration, shaping my approach to scientific inquiry.

I am also immensely thankful to my labmates–Sayed Saad Afzal, Ahmed Allam, Jack Rademacher, and Waleed Akbar–for their collaborative spirit, assistance, and support throughout the countless experiments and the writing process. Their camaraderie and shared insight have not only enhanced the quality of my work but have also made the arduous journey enjoyable and memorable.

My friends and family deserve special recognition for their unwavering love, encouragement, and belief in my abilities. Their endless support and sacrifices have not only sustained me during challenging times but have also been the foundational pillars upon which my accomplishments stand.

Lastly, I would like to express my profound appreciation to the Massachusetts Institute of Technology. The institution has not only been a place of academic growth but has also offered an environment where I could thrive amongst peers and mentors who share a commitment to excellence and innovation. The opportunities, resources, and communities I have encountered here have been pivotal in my development as a researcher and as an individual.

Contents

List of Figures

Chapter 1

Introduction

Low-power underwater sensor networks have important applications in long-term monitoring of ocean climate change, offshore seafood farms, and underwater infrastructure (such as data center networks, offshore wind farms, oil and gas sites, etc.) [1, 2, 3, 4]. A canonical operation environment for these systems involves deploying a network of sensor nodes on a river bed or ocean floor, and having these sensors upload their data to a mobile node (such as an underwater drone, surface vessel, or ship) as it approaches each of these sensor nodes [5, 6].

The recent introduction of underwater backscatter paves the way for long-term deployments of such networks due to the significantly lower power consumption of underwater backscatter in comparison to traditional underwater communication technologies (by about 5-6 orders of magnitude). Underwater backscatter nodes differ from traditional underwater acoustic networks[1] in that they communicate by reflecting rather than generating their own acoustic signals. As a result, underwater backscatter nodes communicate at net-zero power since the energy-consuming process of generating acoustic signals (to be reflected by the nodes) is offloaded to an access point placed on a drone or ship that has a dedicated energy source. This new asymmetric underwater networking architecture extends the battery life of underwater sensor nodes (since their energy can be used entirely for sensing rather than power-hungry acoustic communication), enabling longer-term deployments of large-scale ultra-low-power sensor networks.

Despite important advances in underwater backscatter, existing systems still cannot work with mobility. The inability to operate under mobility prevents their use in scenarios like those described earlier, where backscatter sensor nodes need to upload their data to a mobile node. In particular, state-of-the-art systems have only been demonstrated in static or near-static settings [7, 8, 9, 10, 11], where the access point (or reader) and backscatter nodes were affixed to rods underwater or suspended in the water from static locations.

Enabling underwater backscatter to operate reliably under mobility faces two key challenges:

- **Doppler:** Unlike traditional RF backscatter communication systems (e.g., RFID or ambient backscatter), underwater backscatter networks suffer from a significant Doppler shift when they are mobile. The Doppler effect refers to the shift in frequency

[1] Underwater communication relies on sound/acoustics rather than radio frequency (RF) signals since RF decays exponentially in water.

that occurs due to mobility. A well-known example is how we hear an ambulance siren go faster (higher frequency) as it approaches and slower (lower frequency) as it goes away. The Doppler shift in frequency is inversely proportional to the speed of propagation of the carrier wave. As a result, the frequency shift experienced by underwater acoustic backscatter is $200,000\times$ more significant than that experienced by RF backscatter, and thus cannot be neglected.[2] Leaving this Doppler shift unaddressed causes significant errors in synchronization, packet detection, and decoding. While this problem has been addressed in previous underwater communication systems, existing solutions are complex and power-hungry, making them infeasible to deploy on ultra-low-power underwater backscatter nodes.

- **Lack of robust link layer:** The second challenge is that underwater acoustic networking is inherently prone to higher bit error rates than traditional wireless communications, a phenomenon that is well documented in the literature [12, 13]. Here too, the ultra-low-power nature of underwater backscatter makes traditional methods for robust link-layer underwater acoustic networking too complex and power-hungry for these networks. Yet, to enable efficient and robust underwater networking, especially under mobility, these systems need a link layer that accounts for the energy and complexity constraints of underwater backscatter nodes. It is worth noting that this problem is not just limited to underwater backscatter under mobility, but in static settings as well. Since past research has focused primarily on the physical layer of underwater backscatter (with a shim link layer focused on addressing), a reliable link layer remains missing.

To overcome these challenges, we introduce EchoRider, the first system that enables reliable underwater networking under mobility.[3] Our system design introduces three key components to overcome the above limitations:

- **Robust, chirp-based downlink protocol:** EchoRider's first component is a robust downlink protocol for communicating between the reader and the backscatter node. The protocol is inspired by recent advances in LoRa wireless networks [14], which exploit frequency diversity (through a chirp-based modulation scheme) to operate in multipath-dense and mobile settings. However, in contrast to typical LoRa chips, underwater backscatter nodes are more energy-constrained and difficult to recharge. As a result, incorporating classical solutions for de-chirping like frequency mixers and/or high sampling rate ADCs (with digital de-chirping) would significantly increase the power and complexity of these nodes. Thus, EchoRider introduces a new sub-Nyquit decoding mechanism that enables it to harness the robustness of LoRa-type modulation on ultra-low-power underwater backscatter nodes. Even though sub-Nyquist sampling introduces aliasing, it preserves the frequency diversity of the (chirp) channel, which provides robustness to frequency selectivity. In §3.1, we describe this method in detail, and how EchoRider deals with the aliasing from sub-Nyquist and extends this method with a protocol for frame synchronization and robust decoding.

[2]The speed of RF propagation is 3×10^8m/s, while underwater acoustics is around $1,500$m/s.

[3]This work does not raise any ethical issues.

- **NACK-based backscatter retransmission algorithm:** EchoRider's second component is a retransmission protocol that enables efficient recovery from packet errors. Unlike typical wireless networks, where sending acknowledgments has negligible overhead, the process of acknowledging packets in underwater backscatter is challenging due to the long delay spread of the acoustic channel coupled with the full-duplex (and asymmetric) nature of backscatter communication. To overcome this challenge, EchoRider introduces a NACK-based retransmission algorithm, whereby the backscatter node opportunistically transmits on the uplink until it receives a NACK from the reader. In §3.2, we describe the algorithm in detail, how EchoRider can decode the NACKs during full-duplex backscatter, and how it employs an active recovery approach to efficiently and robustly communicate.

- **Doppler-resilient backscatter decoding pipeline:** Finally, to deal with the time-varying channel with mobility on the uplink, EchoRider's reader methodically designs and implements a pipeline to robustly decode the backscatter packets. The pipeline features an adaptive equalization algorithm (with a phase-locked loop for phase-tracking), polar coding, and an equalizer retraining mechanism for underwater backscatter.

We implemented a prototype of EchoRider and tested it in a river using a custom-built speed-control platform and an off-the-shelf underwater drone. Our backscatter nodes are fabricated in-house following existing open-source designs for the acoustic transducers, matching networks, and control hardware [15]. The downlink decoding pipeline and backscatter modulation was implemented on the micro-controller of the backscatter node, while the downlink retransmission protocol and adaptive equalization were implemented on the reader. The end-to-end system includes a reader (consisting of a projector and a multichannel receiver array with 6 hydrophones) and a backscatter node.

We evaluated EchoRider in over 1,240 real-world experimental trials spanning different ranges, node speeds, coding rates, and access point power levels. We compared its performance to a state-of-the-art underwater backscatter system as baseline. Our results demonstrate the following:

- While both EchoRider and the baseline can achieve a goodput around 1.4kb/s when static (which is typical for underwater acoustic communication), the baseline's goodput drops to zero beyond 0.1m/s, whereas EchoRider's goodput remains around 770bps at speeds up to 0.6m/s (which are typical speeds for underwater drones).

- EchoRider achieves an uncoded median BER of 10^{-3} at a throughput of 2.5 kbps when the node is moving with a speed of 0.4m/s. In contrast, at the same throughput and power level, a baseline has a median BER of 0.16 even at a relatively low speed of 0.05m/s. This shows that EchoRider outperforms the state-of-the-art baseline by over 160× in terms of BER under mobile settings.

- Interestingly, even in static environments, EchoRider outperforms the baseline in low-SNR conditions by about 75% in terms of throughput due to its reliable retransmission protocol.

Contributions: EchoRider is the first underwater backscatter system that can operate reliably under mobility. The design of EchoRider introduces multiple innovations including robust ultra-low power downlink decoding, a NACK-based retransmission protocol, and a Dopple-resilient backscatter decoding pipeline. The thesis also contributes a prototype implementation and real-world evaluation of EchoRider, including the first demonstration of underwater backscatter communication involving a drone.

By enabling underwater backscatter under mobility, this thesis gets the technology closer to long-term real-world deployments with mobile nodes like underwater drones and surface vessels. As the research evolves, it would be valuable to continue improving the technology to achieve higher throughputs and node speeds and to extend it to large-scale underwater backscatter networks.

Chapter 2

Background & Challenges

(a) Underwater Backscatter. Underwater backscatter is an asymmetric networking technology, where the access point or reader enables communication and arbitrates medium access for a large number of ultra-low-power nodes. A canonical example is shown in Fig. 2.1 with a single backscatter node for simplicity. A communication session starts when the reader transmits an acoustic signal from its projector on the downlink. The signal arrives at the backscatter node, which communicates by modulating the reflections of the downlink signal; for example, it can switch its reflection coefficient between two states (e.g., reflective and non-reflective) to encode packets in binary. A hydrophone at the reader senses the changes in the reflection and uses them to decode the backscatter packets. Past research has studied and demonstrated underwater backscatter in practical environments (pools, lakes, river, and the ocean) [7, 16], demonstrated the potential to communicate with multiple nodes [17], studied the link budget theoretically [18], and developed different applications using the technology [8, 19].

Unfortunately, past work on underwater backscatter has been limited to static test setups and still cannot operate under mobility. To better understand the challenge with mobility, we ran an experiment with underwater backscatter in a river (evaluation environment is further described in §4.3). In this experiment, we reproduced a state-of-the-art past system for a single-transducer node [16].[1] The node was programmed to backscatter packets on the

[1]For simplicity, we did not consider nodes that consist of multiple arrayed transducers, since the main benefit of the array is extending the range.

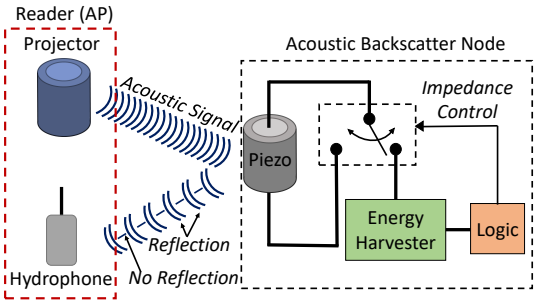


Figure 2.1: Piezo-Acoustic Backscatter.

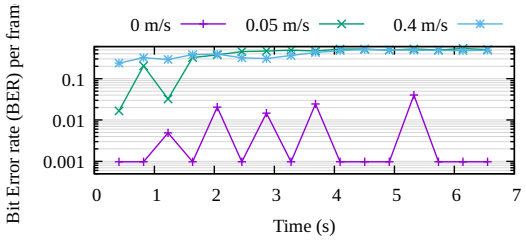


Figure 2.2: BER degradation over time at mobility.

uplink at a rate of 2.5 kb/s. We compared two scenarios: the first in which the node was static, and the second in which it was moved at a speed of 0.05 m/s and 0.4 m/s.

Fig. 2.2 plots the per-frame bit error rate (BER) as a function of time with and without mobility (at speeds of 0m/s, 0.05m/s and 0.4m/s). The plot shows that under mobility, the BER for underwater backscatter fluctuates between 0.02 and 10^{-3} when the node is static, which are expected BERs for uncoded bits in underwater acoustic communication. However, when the node is moving, the BER quickly rises to 0.5 (i.e., becomes undecodable) even at low speeds such as 0.05 m/s. This is because existing underwater backscatter decoders still cannot deal with rapid channel fluctuations. Moreover, due to the lack of channel tracking mechanisms, even if the starting BER is acceptable in the first frame of a packet after the header (e.g., 10^{-2}), the estimated channel from the header quickly diverges due to mobility and the frames become undecodable. This demonstrates the need for a new design that is robust to mobility and dynamic channel conditions.

(b) Underwater acoustic communication under mobility. Underwater communication primarily relies on acoustic signals because radio frequency (RF) signals decay exponentially in water. The underwater acoustic channel is known to be one of the most challenging channels due (a) to the dense multipath particularly in shallow-water environments, where signals repeatedly bounce off the seabed and surface, and (b) the Doppler shift caused by mobility.

In classical underwater networking systems, the two communicating nodes are symmetric, and they employ complex modulation and decoding schemes, such as OFDM and adaptive equalizers to deal with the severe multipath and Doppler shifts [20, 21, 22]. Unfortunately, these schemes cannot be easily adapted for underwater backscatter for three reasons:

- First, due to the low-power nature of underwater backscatter nodes, they can neither transmit nor decode complex modulations like OFDM.
- Second, these nodes also cannot employ complex equalization schemes to decode the downlink on the backscatter node, because they are complex and computationally intensive for backscatter nodes.
- Finally, the reader itself in underwater backscatter cannot use existing equalization schemes out-of-the-box because they are meant to work with active modulation.

(c) Link Layer Challenges. In addition to the physical layer challenges under mobility, achieving a robust and efficient link layer for underwater backscatter is also challenging. In standard networked systems, a reliable link layer can be achieved by exchanging acknowledgments (ACKs). However, in the context of underwater backscatter, requiring the reading to ACK each packet incurs significant overhead. This is due to two reasons: first, each transmission requires a long training sequence (for the equalizers to learn the channel), which is proportional to the delay spread of the underwater channel; thus, pausing a backscatter transmission to wait for ACKs incurs overhead for both the waiting and the retraining times. Second, backscatter communication is full-duplex, which means that the delay spread is a complex function of both the downlink and uplink channels. The combination of these two factors makes designing an efficient and robust link layer for underwater backscatter

challenging, due to the uniqueness of these factors even when compared to the closest counterparts of underwater backscatter: underwater acoustic communication (which is typically half-duplex) and RF backscatter [23, 24] (which has orders of magnitudes smaller delay spread in comparison to underwater acoustics).

Chapter 3

Design

EchoRider is an underwater backscatter system with a robust link-layer protocol that operates reliably in static and mobile settings. Fig. 3.1 shows the end-to-end system architecture for EchoRider. The system architecture has three key components: a robust downlink protocol, a retransmission protocol, and a Doppler-resilient uplink decoder. In this chapter, we explain each of these sub components in detail.

3.1 Underwater Backscatter Downlink with Mobility

The first component of EchoRider is a downlink protocol. The purpose of the downlink signal is to query the backscatter node. This command can also be used to wake up the backscatter nodes, contributing to their energy conservation; i.e. since these nodes are inactive most of the time, they can conserve energy and respond with the backscatter signal only when prompted. Past underwater backscatter-based systems used a single frequency on the downlink where they relied on modulation schemes like pulse width modulation (PWM) to convey data. However, using a single frequency proves impractical for underwater environments especially under mobility, primarily due to the high frequency selectivity and rapid fading of underwater channels. Unfortunately, low-power backscatter nodes cannot decode these because they cannot apply complex equalization methods to remove the impact of the channel.

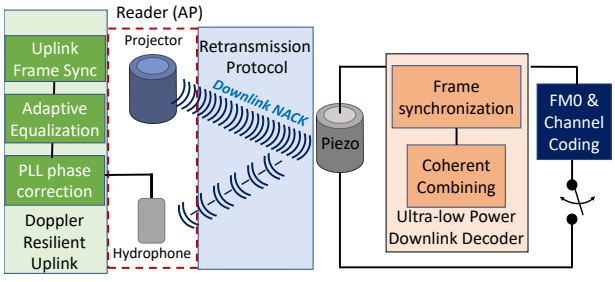


Figure 3.1: An overview of EchoRider's architecture.

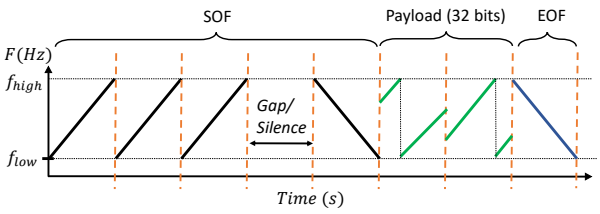
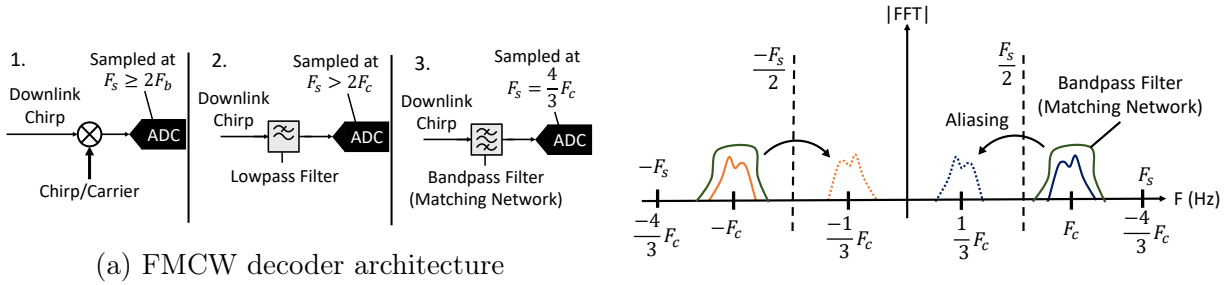


Figure 3.2: EchoRider's Downlink Protocol.



(a) FMCW decoder architecture

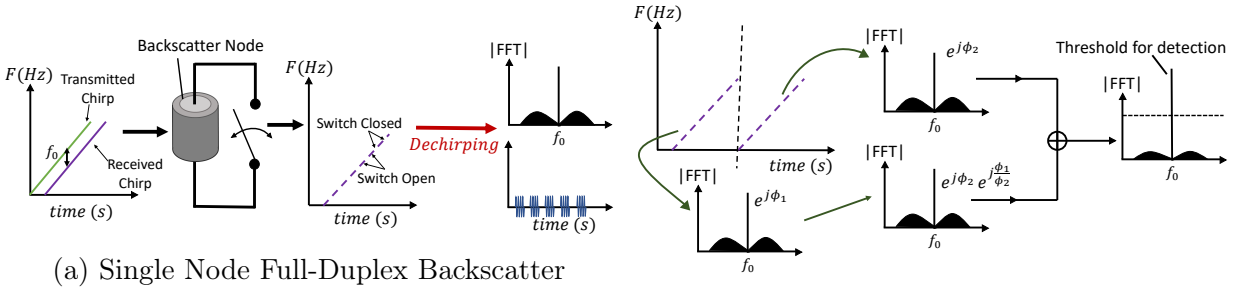
(b) Sampling at $\frac{4}{3}F_c$

Figure 3.3: Doppler and Underwater Backscatter.

To overcome this, we developed a chirp-based protocol for downlink communication. The interesting thing about chirp is that instead of transmitting over a single frequency we transmit over a wideband which makes it resilient to mobility and frequent fluctuations in wireless channels Furthermore, Doppler effects do not significantly impact the downlink chirp-based transmission. This is attributed to two main factors. Firstly, adding Doppler shift (i.e. frequency shift) to chirps is the same as introducing a time delay which means that the chirps do not undergo severe distortion. Secondly, the downlink operates at a considerably lower throughput of 60bps, which further reduces susceptibility to Doppler effects. Fig. 3.2 shows EchoRider's downlink protocol. The protocol comprises three components: the start of frame (SOF), the payload (containing query data), and the end of frame (EOF). EchoRider decodes the downlink signal through a process that starts with utilizing any two of the three up-chirps in the SOF for synchronization. Next, it detects the EOF, allowing for the dechirping of the payload data to decode the query.

(a) Ultra-low power decoding. While chirps-based downlink communication exhibits resilience to mobility and frequency-selective underwater channels, the hardware design for decoding these signals is typically complex and requires high computational and processing resources that consume substantial power. In Fig.3.3(a), three different decoding options for the received downlink chirps are illustrated. The first option involves using a mixer to down-convert the received chirp using the center frequency of the chirp or by generating another chirp and dechirping the received signal. The mixed signal passes through a low-pass filter and is then fed to an ADC which samples it at a rate that is more than twice the bandwidth of the chirp to adhere to Nyquist criteria, after which it undergoes post-processing. While this idea conserves energy with the ADC sampling at a lower rate, downconverting (or dechirping) in the analog domain by generating another copy of the chirp or carrier is power-intensive and unsuitable for an ultra-low-power design. Another option is to pass the entire downlink chirp through a lowpass filter to eliminate out-of-band noise and sample at a rate twice the bandwidth of the chirp. However, this is sub-optimal as it requires the ADC to operate at extremely high sampling rates, consuming a significant amount of power.

Given that the first two options are unsuitable for an ultra-low-power design, EchoRider relies on the third option, which involves using a bandpass (matching network) filter followed by an ADC sampling at $\frac{4}{3}F_c$. EchoRider adopts this architecture because it allows the system to obtain the received chirp while sampling at a much lower rate, even below the Nyquist limit, thereby conserving power. This might seem counter intuitive because sampling below



(a) Single Node Full-Duplex Backscatter

(b) Coherent Combining

Figure 3.4: Doppler and Underwater Backscatter.

the Nyquist limit would result in aliasing. However, EchoRider leverages aliasing to shift the entire spectrum into the frequency band of interest. Fig.3.3(b) illustrates this concept in the frequency domain: by sampling the chirp at $\frac{4}{3}F_c$, the entire signal undergoes aliasing, and the aliased copies move closer to the DC frequency, now centered at $\frac{1}{3}F_c$. It is essential to note that after aliasing, the signal spectrum is reversed and appears to have a negative phasor rotation, similar to the negative frequency spectrum. Nevertheless, the aliased spectrum can be reversed by effectively swapping the roles of I and Q, thus reversing the rotational direction of the signal's phasor. Another interesting aspect is that this approach would not have worked if the signal had a significant amount of out-of-band noise, as it would also alias into the signal, reducing the overall SNR. This is why EchoRider relies on a bandpass filter (matching network) to allow all signals within the chirp's bandwidth to pass through, while attenuating signals outside the band [1]. Although EchoRider can obtain time-domain samples of the chirp through this method, there are still some challenges that need to be addressed before the system can perform full-duplex communication:

(b) Downlink Frame Synchronization: To facilitate downlink communication between the transmitter and the backscatter node, it is necessary to synchronize to the beginning of the transmitted chirps. A straightforward solution for synchronization involves correlating with the transmitted chirps and identifying the time index that maximizes the correlation. However, this approach is not extremely efficient due to the high computational cost, i.e., $O(N^2)$. Instead, EchoRider adopts an alternate method where it mixes the received chirp with the transmitted chirp in the digital domain (on the backscatter node) [2]. It then utilizes the peak frequency in the FFT of the product to determine the amount of offset. This process requires $O(N)$ time, making it more feasible, especially considering that the node needs to detect these downlink commands in near real-time.

Fig. **??**(a) depicts the processing pipeline for achieving synchronization with the downlink chirps. The backscatter nodes receive the downlink chirp, potentially while already transmitting uplink data. As the backscatter node is unaware of the received downlink, it continues switching between "on" and "off" states, and the downlink chirp gets modulated with the uplink data. Upon performing the dechirping operation on this modulated

[1] This hyper-Nyquist bandpass sampling technique has been implemented in an number of RF systems, e.g. [25], but we adopt it for the acoustic signals.

[2] Note that this consumes less power than dechirping in the analog domain

received chirp, followed by a low-pass filter, EchoRider obtains a modulated single tone. Subsequently, it proceeds to take the FFT of the signal, retrieves the frequency f_o with the maximum amplitude, and estimates the offset t_s using $t_s = \frac{f_o}{s_c}$ where s_c is the slope of the chirp. Employing this technique, EchoRider achieves synchronization to the beginning of the chirp, enabling accurate decoding of the payload bits.

(c) Reliable Downlink Decoding: So far, we have explored how EchoRider synchronizes to the start of the uplink modulated chirp using the frequency of the peak amplitude in the FFT. The peak amplitude serves an additional purpose—by applying a threshold over the peak value, EchoRider can determine whether a downlink signal was transmitted. However, as the chirp is modulated with some uplink data, the carrier's amplitude is reduced after dechirping due to some power being distributed across the bandwidth of the uplink data. Consequently, in noisy channels, the system might fail to detect the downlink chirp (or experience false alarms), significantly impacting system goodput. One potential solution is to enhance the SNR by adding two up-chirps before the dechirping operation. However, this approach is impractical because the two received chirps will have different phase offsets due to node movement, especially in higher frequency components of the chirp.

To tackle this issue, EchoRider transmits three up-chirps in the Start of Frame (SOF) and extracts two copies from the received chirps. Subsequently, it individually de-chirps each copy, identifies the peak amplitude for each, and estimates the phases ϕ_1 and ϕ_2 from the FFT's maximum amplitude peak for both copies. The second copy of the chirp is then multiplied by $e^{j\frac{\phi_1}{\phi_2}}$. After this, EchoRider combines the two chirps coherently by adding them together. Fig.**??**(b) illustrates how EchoRider enhances SNR through coherent combining. By adopting this approach, EchoRider ensures that only the carrier, after dechirping, combines coherently, while the uplink data does not. This results in improved SINR, making it easier for the node to detect a downlink command and reducing the chance of false alarms or missed commands. Once EchoRider successfully detects the two chirps, the logic controller directs the backscatter node to cease the ongoing uplink during the gap/silence period of the downlink. This interruption means that the downlink chirp is no longer modulated with the uplink data. Subsequently, EchoRider can proceed with decoding the remaining downlink using the standard LoRa protocol[14].

3.2 Retransmission Protocol

To enable a reliable link layer, EchoRider employs a retransmission protocol. In standard wireless networks, this is realized by having the receiver send acknowledgments for each packet (or set of packets) it decodes correctly. In the context of underwater backscatter, this can be achieved by requiring the backscatter node to interrupt its streaming (e.g., of camera images [8]) to wait for an acknowledgment from the reader before it continuing its transmissions.[3]

[3]Since the reader arbitrates medium access in backscatter systems (like RFIDs), it does not typically require the backscatter node to acknowledge its downlink transmission for two reasons. First, the downlink is primarily used for low-throughput querying, while the uplink backscatter is needed for higher-through sensor data streaming. Second, the mere fact of the backscatter node activating and responding to it implies that the downlink query was received and decoded correctly. If the reader does not receive a response within

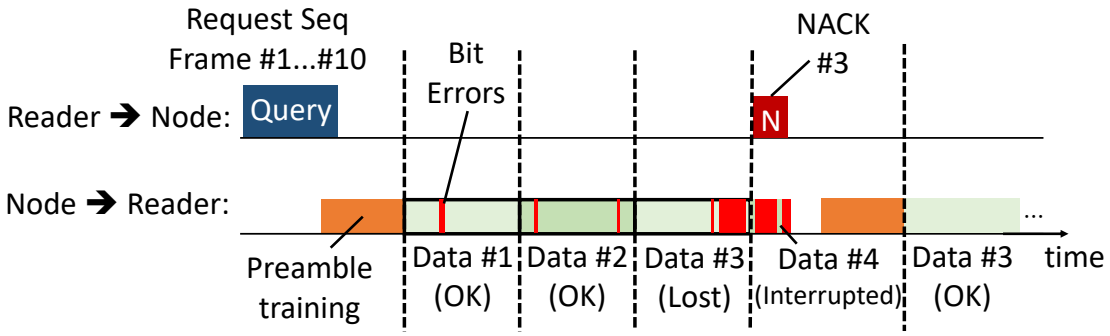


Figure 3.5: **EchoRider's Retransmission Protocol.**

However, a standard acknowledgment-based link-layer protocol is undesirable for underwater backscatter since it leads to significant overhead. In particular, for the reader to decode correctly, it requires a long training sequence, which is usually embedded in the header of the underwater backscatter packet on the uplink (see §3.3 for more details on decoding). This means that each time the underwater backscatter node interrupts its transmission to wait for an acknowledgment from the reader, it must retransmit the entire training sequence, which can lead to up to 50% reduction in throughput.[4] Thus a more ideal approach is to send a NACK instead of ACK where the node is asked to stop and retransmit the prior frame if the reader had issues decoding.

Fig. 3.5 illustrates the retransmission protocol employed by EchoRider. The process begins with the reader initiating a query on the downlink, requesting the initial frames from the backscatter node. The backscatter node responds by sending a training sequence followed by the actual data frames. Upon receiving this, the reader utilizes the training sequence to train the decoder, then proceeds to decode the data. While decoding, potential bit errors may arise, but these can be addressed through Error Correction Coding (ECC), ensuring accurate data retrieval. In cases where errors persist beyond the FEC correction capability (as seen in Data #3 in the figure), EchoRider issues a NACK command, prompting the backscatter node to retransmit the preceding frame. This interruption affects the decoding process for the subsequent frame at the reader (Data #4). Upon receiving the retransmission request, the backscatter node resends the known preamble training sequence along with the previous frame (Data #3). Next, the backscatter node resumes normal operation, transmitting subsequent frames on the uplink.

3.3 Robust Uplink & Mobility

So far we have discussed how EchoRider enables a robust chirp based downlink protocol and how it employs an active recovery approach to communicate efficiently. In this section, we

a certain timeout window, it sends a query repeat command.

[4]These training packet are of the order of several 100 milliseconds [16].EchoRider employs a frame size of 1024 bits along with a 512-bit training sequence. Transmitting at a data rate of 2.5 kbps translates to a training duration of 205 milliseconds. This training sequence is commonly utilized for linear equalizers to facilitate effective channel tracking.

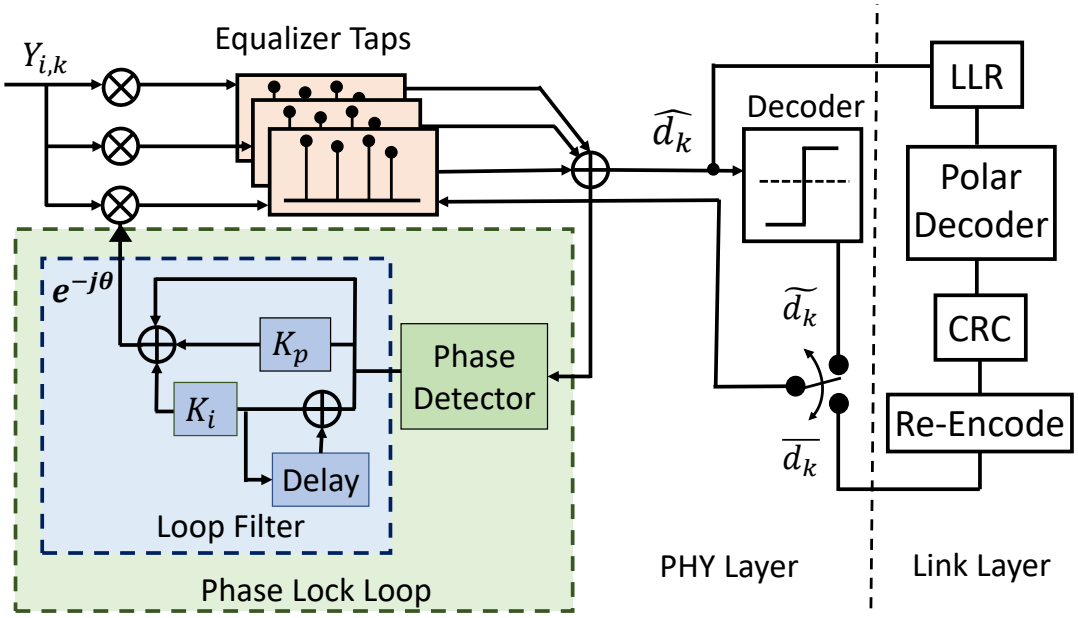


Figure 3.6: **EchoRider's Uplink Decoding Pipeline.** In EchoRider , the received symbols are passed through an equalizer to remove impact of ISI, then fed to a PLL to track and compensate for phase changes.

will discuss how EchoRider tackles the challenge of swift phase rotations in symbols within the backscatter uplink induced by mobility.

For uplink communication, EchoRider transmits a single frequency on the downlink and records the signal that has been reflected and modulated by the backscatter node.[5]. One method for monitoring changes in Doppler-induced phase in the backscatter response involves utilizing an adaptive decision feedback equalizer (DFE). DFEs are commonly employed to effectively counteract the impact of multipath extending across several tens of symbol intervals, making them a fundamental component in many coherent communication receivers. While DFEs can handle signals with moderate Doppler shift, they would need to make rapid adjustments to the equalizer filter taps. This process is computationally intensive and introduces adaptation noise. Typically, the tap gains should not change by more than a few percent from one symbol interval to another and using the DFE to correct the rapid variation in phase may eventually cause the equalizer taps to diverge. As pointed out in [27], relying solely on the equalizer to compensate for backscatter phase shift can result in an excess mean squared error (MSE) during tap weight adjustments. This issue arises mainly because the tracking gain must be set higher to rectify the rapidly changing phase.

To address this issue, EchoRider relies on a improved DFE algorithm containing an embedded phase-locked-loop (PLL) to remove phase shift due to Doppler. Fig. 3.6 shows the block diagram for EchoRider's uplink decoding pipeline. The baseband symbols received from the i^{th} hydrophone channel at the k^{th} time index, denoted as $Y_{i,k}$, undergo a process

[5]Note that our design uses a single switching rate in the uplink for backscatter communication. One option to mitigate Doppler is to use LoRa backscatter [26] (where we send chirps by switching at different rates to communicate data over a wider band) however due to the limited bandwidth of underwater transducers and the issue of damping, chirps are undesirable for uplink since it severely limits the throughput

where they are multiplied by a complex exponential to eliminate the excess phase introduced by Doppler effects. Subsequently, these symbols are equalized using filter taps, which are updated at each time index k. Following equalization, they are coherently combined (expressed as $\hat{d}_k$) and then processed through a decoder to determine the transmitted symbol, distinguishing between bit 1 or 0.

To estimate the excess phase from Doppler, EchoRider exploits the characteristic that backscatter modulation relies on two states encoded as two phases of 0 and π. The ISI corrected symbols $\hat{d}_k$ are squared to eliminate the 0, π variations. Subsequently, EchoRider acquires the phase of the product and normalizes it by 2 to extract the excess phase ϕ_k introduced by Doppler. The estimated phase is fed as an input to a second-order loop filter with proportional and integral gains K_p and K_i. The second order loop filter keeps track of the changes in the phase ϕ_k and produces the final phase θ_{k+1} used to eliminate Doppler effects in the incoming symbol in the subsequent time step, namely: $\phi_k = \frac{angle(\hat{d_k}^2)}{2}$ and $\theta_{k+1} = \theta_k + K_p\phi_k + K_i \sum_{p=0}^{k} \phi_p$

By separating the task of equalization from the Doppler phase offset estimation, we ensure proper equalizer filter updates despite substantial phase fluctuations in underwater acoustic channels. However, there are additional challenges that may arise while decoding uplink backscatter data in a practical implementation, outlined below:

(a) Uplink Frame Synchronization: The mobility of underwater nodes introduces significant time scaling in the data. This implies that, depending on the node's velocity relative to the receiver, the received backscatter data may undergo time stretching or shrinking, posing challenges in correlating for the beginning of the frame. While sending training bits once for the initial frame might succeed in synchronization for that frame, however in successive frames (without training bits), EchoRider could lose track of the frame's beginning, leading to high BER.

To address this, one potential solution is to add training bits at the start of every frame. However, this approach is impractical as it reduces the overall throughput. EchoRider tackles this challenge in two steps: first, the node encodes the uplink backscatter data using channel coding (polar code), then upon receiving the coded backscatter symbols, the reader attempts to decode the data to obtain raw bits. A CRC check is employed to verify if the resulting frame has a 0 BER. Subsequently, EchoRider's reader re-encodes the data using polar coding and correlates the received packets with the sequence of these coded bits to determine the beginning of the next frame. In cases where the CRC check fails, it follows the retransmission protocol to receive the correct frame, which is discussed in §3.2.

(b) Adaptive Equalization: As previously mentioned, the equalizer employed by EchoRider requires updating at every time step k. However, there could be scenarios where a few samples belonging to a certain symbol (or a few symbols in a frame) are corrupted due to the fast-fading underwater channel. This corruption can result in errors in the equalized output $\hat{d}_k$. If this situation persists over several frames, the errors would accumulate, eventually causing the equalizer to lose track of the channel.

To mitigate this issue, EchoRider leverages the link layer to retrain the equalizer. The processing pipeline for EchoRider's MAC layer is depicted in Fig. 3.6. After receiving the equalized output $\hat{d}_k$, EchoRider converts it into a log-likelihood ratio (LLR) using a sliding-

window noise estimator and stores it in a buffer. Once all the samples corresponding to a frame (which contains multiple symbols) are obtained, EchoRider passes these LLRs through a polar decoder [6] and performs a CRC check of the uncoded bits. If the CRC check passes, it re-encodes the bits and uses this re-encoded output to retrain the equalizer taps. By applying the training process twice, EchoRider ensures that, in the worst case, errors are only propagated until the end of a frame and not beyond it. This approach significantly enhances the equalizer's robustness to the challenges posed by the fast-fading underwater channel.

(c) Wideband Self Interference: Enabling the downlink protocol introduces additional challenges, particularly for uplink decoding at the reader, attributed to wideband self-interference. Self-interference occurs when the transmitter's signal directly interferes with the signal from the backscatter node. As mentioned in the system overview in §3.1, EchoRider utilizes a chirp for downlink communication. Due to ISI in the transmitted chirps, EchoRider may face difficulty decoding the initial frames, given the strong self-interference from the previous downlink transmission.

Traditionally, self-interference is mitigated using a band-stop filter. However, in this scenario, it is unfeasible as the band of the chirp overlaps that of the uplink. Alternatively, we may introduce gaps or silent periods between uplink frames. However, this proves to be ineffective: the underwater channel experiences rapid changes, which means that adjusting equalizer taps using a frame that was sent some time ago is futile. This would lead to the equalizer consistently removing ISI for an outdated channel, failing to mitigate errors induced by multipath.

EchoRider addresses this challenge by introducing a fixed duration delay to the uplink sequence. This delay allows the uplink sequence to reach the hydrophone after the ISI from the chirp has significantly attenuated, preventing interference with the weak backscatter. It is important to note that this delay is introduced only once per downlink command.

To summarize the design of EchoRider, the reader transmits a single tone on the downlink for standard backscatter communication and employs a DFE (with an integrated PLL) to address distortions coming from Doppler. In the scenario where it detects an error (after applying error correction) in the uplink data, it sends a LoRa chirp on the downlink to request a retransmission. The downlink chirp is decoded using a sub-Nyquist decoding mechanism that enables EchoRider to harness the robustness of LoRa-type modulation on ultra-low power backscatter nodes. This completes the loop for the E2E EchoRider system.

[6]The encoding uses frame-size 1024 and coding parameter in 3GPP TS 38.212 [28]. At the receiver, we perform the standard CRC-aided list decoding [29] on the LLRs.

Chapter 4

Implementation

4.1 Backscatter Node

Fig. 4.1(a) shows the underwater backscatter node used by EchoRider. The backscatter node architecture includes a underwater transducer, transformer-based matching network, and a on-board tag firmware.

(a) Transducer. The transducer's core element is a customized piezoelectric cylinder (PZT-42-⌀50.8 mm *⌀44.704 mm * 38.1 mm) from Zibo Yuhai Electronic Ceramics Co., Ltd, with a nominal resonance frequency of 22 kHz in radial mode. The transducer was fabricated using a process similar to that detailed in prior works [7, 30].

(b) Matching Network. We implemented wide-band matching for our backscatter node for maximum power transfer to ADC which improves both downlink SNR and uplink communication range. As shown in Fig. 4.1d, the matching consists of a 4th order LC ladder network to create a wideband filter. We made use of 1206 C0G capacitors and RM-4 ferrite core for the inductors. The values of the capacitors C_2,C_3,C_4 and inductors L_1,L_2,L_3,$L_{4,pri}$ were selected using an min-max optimization process where different values were selected iteratively to minimize the reflection coefficient S_{11} parameter of the network within a certain bandwidth. This can be expressed mathematically as:

$$\min_{L_{1,2,3,4},C_{2,3,4},Z_0} \max_{f\in[f_l,f_h]} |S_{11,\text{ref}\,Z_0}|$$

where Z_0 is the load impedance expected at the transformer primary $L_{4,pri}$ and f_l, f_h represent the limits of the band of interest. After determining these component values, $L_{4,sec}$ is selected so that the Z_0 get scaled and the voltage is maximized at the ADC input (without causing saturation). Since the ADC cannot sample negative voltages, the input is biased with a DC offset of $V_{DD}/2$. This optimization process ensures a matched $|S_{11}| < -10$dB over the frequency range of 22kHz $\pm$ 4kHz.

(d) The MCU and firmware. The firmware of our backscatter runs on a STM32 microcontroller (MCU) and it incorporates an FM0 and a Polar code encoder for uplink communication. To generate the uplink payload, our firmware runs a Maximum-Length Sequence (MLS) pseudorandom generator, and on top of it, it encodes with Polar encoder and FM0 modulator,[1] which is subsequently clocked out via low-power SPI3 that drives the backscat-

[1]We need this additional subcarrier-modulation (FM0) to avoid self-interference at the reader. Although

ter Tx switch (MOSFETs). For downlink reception, the ADC4 monitors the voltage of the switching port using the aliasing sampling frequency (see Sec. 3.1), and the subsequent dechirping and FFT operations are performed in real-time in the ADC's DMA interrupt.

4.2 Reader

Fig. 4.1(b) shows our reader consisting of 4 main components:

(a) Transmitter. We use an underwater piezo cylinder transducer with a resonance frequency of 22 kHz enclosed within a customized thin-wall stainless steel housing. The interior of the housing is filled with liquid paraffin, providing insulation and relaying the acoustic signal. The use of these materials increases the durability during transmission compared to prior designs that relied on polymer-encapsulated transducers.

(b) Transmitter Chain. In order to transmit the wideband chirp signal, the transmitting chain employs a high-order matching network designed using a similar methodology as in 4.1, with the exception of using higher power-rated inductance (PM50-39 [31]) and capacitance (X2 Film [32]) components. The transformer secondary coil matches the impedance to the 4Ω standard speaker load impedance. This matched load is driven by a Texas Instruments TPA3245 power amplifier and an ESS es9018 DAC.

(c) Receiver Array. The receiving system is built around six low-cost Aquarian Audio H1C hydrophones, arranged in a vertical array above the projector. An AD8656 preamp is used specifically for the H1C hydrophones, in conjunction with a Cirrus Logic CS5381 ADC. A Foam board that reflects sound wave is placed between the hydrophone and projector to prevent hydrophone saturation due to self-interference of the excitation carrier.

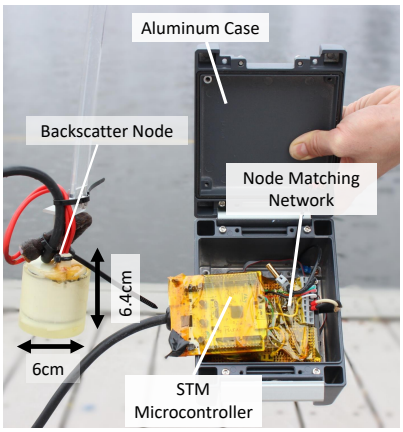
(d) Digital Backend and Software:. The digital aspect of our reader incorporates eight channels of ADCs and DACs connected to an XC7A200T FPGA on a custom-designed PCB This system is connected to a computer responsible for generating the transmit carrier, enabling downlink trigger and recording/processing the hydrophone data. We carefully designed the software to enable the real-time retransmission protocol: we implemented sample streaming from/to the FPGA as a hardware abstraction layer in a C++ gRPC service. The rest of the reader was implemented using python client scripts that transmit and receive chunks of samples for downlink generation and uplink decoding.

4.3 Evaluation

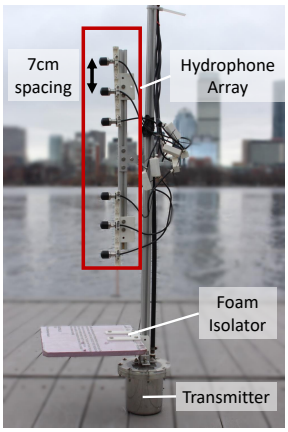
We evaluate our system in a realistic setting, detailed below:

(a) Motion mechanism:. To evaluate underwater backscatter under mobility, we created a towed buoy mechanism to move the node in a controlled trajectory. As shown in Fig.4.1c, the backscatter transducer was mounted on a buoy made of XPS foam, using a rigid acrylic bar. A closed-loop, rope-pulley system was designed to drive the buoy. A 400W AC servo motor [33] was utilized, driving two coaxially mounted winches that contained a continuous

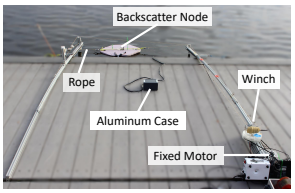
we've explained our receiver using BPSK assumption, the receiver's equalization of FM0 and BPSK are interchangeable given a preprocessing step [30].



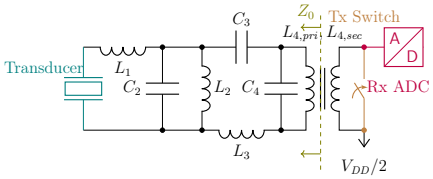
(a) Underwater Backscatter Node.



(b) Reader.



(c) Sample Experimental Setup.



(d) Backscatter Node Matching Network.

Figure 4.1: System Implementation.

loop of ⌀1.6 mm Kevlar rope. This mechanism can achieve a maximum speed of 0.6 m/s with a positioning accuracy of around 5 cm. We also tested the performance of our system using a BlueROV Drone to evaluate the system in practical settings

(b) Mobility Environment, dataset and metrics: . For the evaluation of EchoRider, we created a dynamic test environment in a 6m deep river, using a dock-mounted pulley system to achieve a 15m travel range for the backscatter node. We tested with speeds ranging from 0m/s to 0.6 m/s, and divided our data into two range groups: 5-10 m and 10-15 m from the reader. Fig. 5.1a shows the different locations where we tested in the river.

Chapter 5

Results

5.1 Main results

We evaluated the per-frame BER performance, which is critical for assessing the decodability of underwater backscatter communication when Forward Error Correction (FEC) is enabled. In this experiment, we move the node at different speeds and at different ranges. We evaluated the BER for EchoRider and compared it to a baseline system that employed only an adaptive equalizer. Fig. 5.2a plots a CDF of the BER for every 1024-bit frame of EchoRider (solid lines) and the baseline (dotted lines). This result shows that EchoRider exhibits superior resilience in high-mobility environments. Specifically, at 0.4 m/s, EchoRider maintains a median per-frame BER of 2e-3. In contrast, the baseline system, yields a substantially higher median BER above 0.17 even at 0.05 m/s. This demonstrates the importance of EchoRider's Doppler tracking and channel tap retraining for improving the BER in mobile scenarios.

We further evaluated the end-to-end Goodput of EchoRider as a function of mobility level and communication range, employing the full system with retransmission. In this experiment, the system attempts to reliably deliver 22 frames over 30 runs at different combinations of distances and speeds. The goodput is computed from the total time the system takes to transmit the frames. We compared the performance of EchoRider to a baseline system that does not use the retransmission strategy, transmitting 16 frames per run over a fixed duration of 8.6 seconds. The baseline's goodput is calculated using the number of CRC-correct frames. Both systems use a 3/4-rate Polar code. Fig. 5.2b plots the goodput of EchoRider (solid) and the baseline (dotted) as a function of mobility level. We show results across both close ranges (5-10m) and far ranges (10-15m). We observe that despite an increase in mobility levels up to 0.4m/s, the system maintains a 60% Goodput at closer ranges and 40% at farther ranges. In contrast, the baseline's Goodput shows a significant drop, halving its capacity in the face of any mobility, due to the Doppler phase changes. This result shows the importance of EchoRider's techniques for ensuring robustness in challenging mobile underwater environments.

5.2 Microbenchmarks

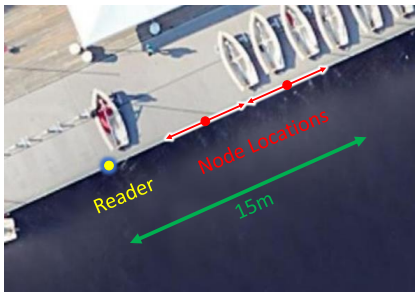
(a) Downlink reliability results and comparison: To validate the effectiveness of our chirp-based downlink design in EchoRider, we evaluated the downlink BER across various distances from the reader, ranging from 3m to 20m. For this microbenchmark, we used a transmit power of 3.2W and we transmitted a total of 2540 downlink bits at a rate of 60bps for each range location. We compare the performance of our system to a state-of-the-art baseline (using a trace driven simulation [1]) for the Pulse Width Modulation (PWM) decoder and a single downlink frequency [7]. As illustrated in Fig. 5.2c, the chirp-based downlink of EchoRider maintains a BER of zero across all distances, demonstrating no detectable errors. In contrast, the PWM baseline has significant fluctuations in BER with increasing distance. This is due to sensitivity to multipath when transmitting a single-frequency on the downlink, which suffers from nulls due to destructive interference at certain ranges. In contrast, EchoRider's approach harnesses frequency diversity (from the wideband chirp) to achieve low BER vs range

(b) Impact of Mobility level: We evaluated the BER (per frame) performance of EchoRider as a function of different speeds. To this end, we performed experiments where we varied the speed of the backscatter from 0m/s to 0.6m/s. As depicted in Fig. 5.2d, our results indicate that at lower speeds (up to 0.2 m/s), 80% of frames maintain a raw BER below 0.01, which is within the decodable range of code rates around 3/4. However, we observe a degradation in performance at increased speeds of 0.4 m/s and 0.6 m/s. These higher error rates demonstrate the importance of our retransmission and recovery mechanisms to have reliable communication in mobile environments.

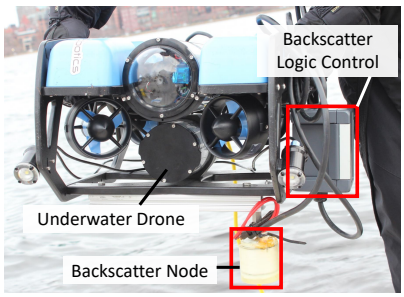
(c) Impact of coding rate. To understand the trade-off between robustness and efficiency, we evaluate the frame error rate (FER) performance of the EchoRider system using different Polar coding rates. The experiments were performed by varying coding ratesfrom 1/8 to 7/8, and changing underwater mobility speeds from 0.0 m/s to 0.6 m/s. As shown in Fig. 5.2e, we can observe that the frame error rate decreases with increased redundancy due to more robust coding. However, at higher mobility speeds, we observe a non-zero frame error rate even at the lowest coding rates, this could be due to several reasons: loss of channel tracking with the Doppler estimator, issues in frame synchronization, or errors in the equalization process. This demonstrates the need for link-layer recovery mechanisms in conjunction with FEC.

(d) Impact of retransmission strategy: To evaluate the impact of our retransmission strategy, we evaluate a partial implementation that incorporates our PLL tracking, but not our retransmission strategy. We compared the goodput of this partial implementation to the full implementation across two operational ranges: 5–10m and 10–15m. The mobility levels ranged from 0.05m/s to 0.6m/s as shown on the x-axis, and the goodput performance is depicted on the y-axis. As demonstrated in Fig. 5.2f, at lower mobility levels (0.05m/s to 0.1m/s), the partial implementation exhibits similar goodput to EchoRider, suggesting that both systems can cope with modest mobility through carrier tracking. However, at higher mobility levels (notably at 0.4m/s and 0.6m/s), we observe a significant decline

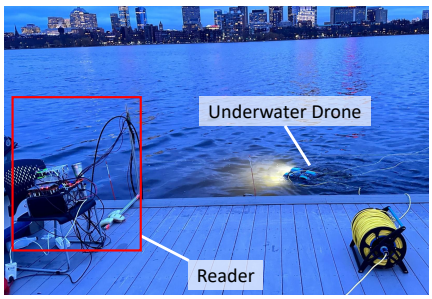
[1]For the simulation we collected transducer and channel responses for a carrier frequency of 23kHz and simulated the envelope detector, assuming a forward biased voltage (Vf) of 0.1V.



(a) Controlled experiment layout with node-reader positioning.



(b) Close-up of EchoRider-equipped underwater drone components.



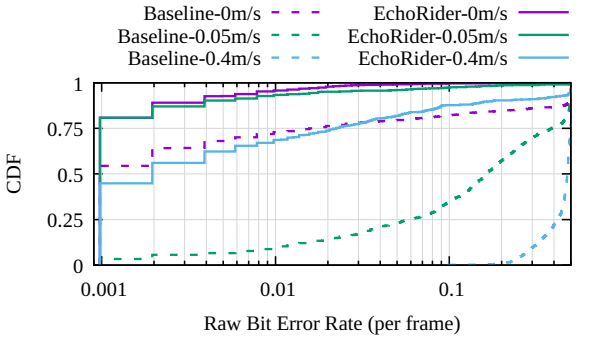
(c) EchoRider drone during open-water operational test.

Figure 5.1: EchoRider experiments: Controlled evaluation setup, drone details, and open-water drone test.

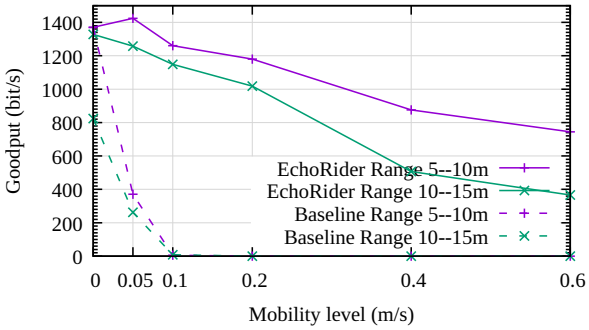
in goodput for the partial implementation, ranging from 10% to 80%, when compared to EchoRider. This degradation emphasizes the importance of EchoRider's immediate recovery mechanism. In high mobility scenarios, the PLL-based system is prone to losing channel tracking after a frame error occurs, which can then affect subsequent frames. In contrast, EchoRider mitigates this issue by reinitiating channel training immediately in the next uplink (after retransmission), effectively preventing the error propagation seen in the partial implementation.

5.3 Underwater drone test

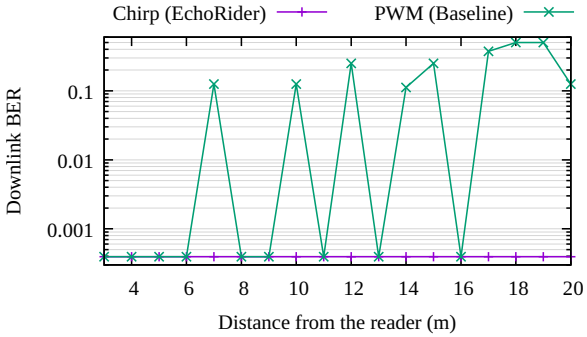
In addition to the controlled experiments with the towed buoy, we mounted the EchoRider backscatter node, along with its waterproof circuit assembly, onto an underwater drone (as illustrated in Figure 5.1b and Figure 5.1c). During a series of seven end-to-end throughput tests, the drone—equipped with the node—moved at a speed of approximately 0.5 m/s and maintained a distance of roughly 10 m from the reader. The results demonstrated that the tag could successfully communicate with the reader, achieving an average throughput of ∼800 bps. These findings are consistent with the results from our controlled experiments, reinforcing EchoRider's efficiency in practical underwater deployment scenarios.



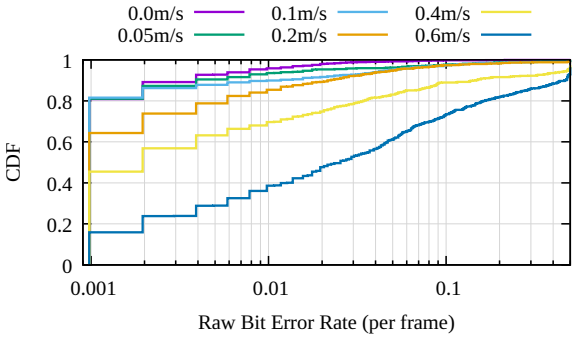
(a) BER versus mobility level.



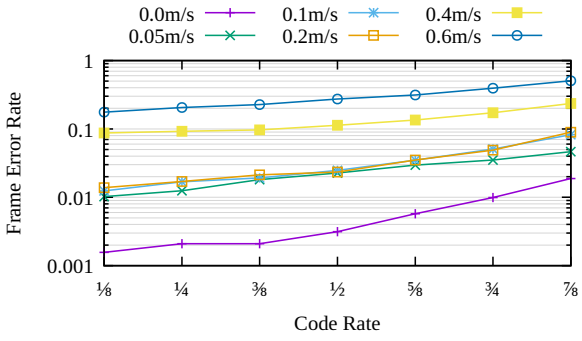
(b) Goodput versus mobility level.



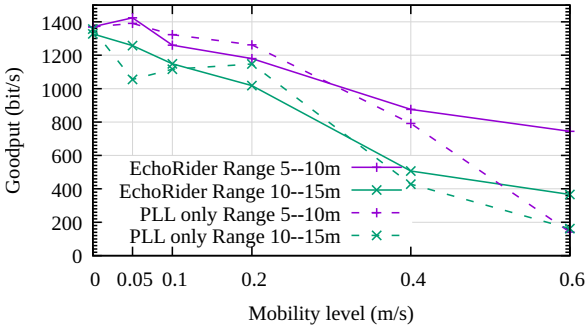
(c) Downlink BER versus distance.



(d) Additional mobility speeds.



(e) FER versus coding and mobility.



(f) Retransmission strategy.

Figure 5.2: EchoRider Evaluation experimental results.

Chapter 6

Related Work

(a) Underwater Backscatter: Past work on underwater backscatter has demonstrated the feasibility of the technology [7], higher-throughput [17, 34], longer-range [16], link budget [35], and new applications in imaging [8] and localization [19]. However, all past work focused on static environments. EchoRider builds on this past literature and is the first to enable reliable underwater backscatter under mobility.

(b) Reliable Underwater Networking: Various techniques have been developed to deal with the challenging conditions of the underwater communication channel, specifically the high delay spread from multipath reflections and Doppler effects stemming from mobility relative to the slow propagation speed of acoustic waves. All of these techniques have been developed for symmetric communication links, and include: RLS combined with PLL carrier tracking [36, 37], symbol-based timing recovery [38], advanced techniques like Time-Update RLS with complex exponential BEM spreads [39], and differential decoding LMS with fewer parameters [40]. EchoRider builds on these past approaches and brings them to the underwater backscatter problem. It is worth noting that although high mobility (5-15m/s) can be addressed using advanced TX waveforms like OFDM [13] and chirp [41], these waveforms are infeasible for due to the inherent binary switching of backscatter, and therefore, we focus on the single-carrier BPSK uplink modulation, and improve its mobility resilience with a mixture of Link and Physical-layer techniques.

(c) Error Recovery and Downlink Communication: Past research has focused on different error recovery methods to counter the unique challenges (i.e., poor quality, asymmetric connectivity, long propagation delays etc.) in underwater acoustic channels. For example, the application of turbo-code-based Forward Error Correction (FEC), turbo equalization, and turbo channel estimation methods to mitigate errors due to low received SNR[42, 43, 44, 45, 46], use of rateless-code for ACK-free correction [47], and ARQ schemes for dealing with extremely long propagation delays [48, 49]. A more detailed overview of these methods can be found in a comprehensive survey [50]. EchoRider builds on similar methods and brings them to underwater backscatter communication where it uses them to decode backscatter data (or request a retransmission in case of an error) to mitigate the impact of Doppler shifts coming from node mobility.

Chapter 7

Conclusion

In this thesis, we present EchoRider a novel system that significantly surpasses the capabilities of existing underwater backscatter systems under challenging mobile environments. Unlike the current state-of-the-art which relies on Pulse Width Modulation (PWM) for the downlink and on equalizer adaptation methods to mitigate channel effects, EchoRider introduces a low power chirp-based downlink strategy that is robust to Doppler coming from underwater mobility. Furthermore, we employ a NACK-based retransmission protocol that promotes effective data recovery, and our Doppler-resilient equalization technique maintains signal integrity in the face of rapid mobility-induced channel variations. Our comprehensive real-world experimentation, featuring over 1,200 trials, demonstrates EchoRider's improvement in both bit error rates by 160x and in sustaining reasonable goodput under speed conditions that render other systems inoperative. We believe this advancement paves the way for EchoRider to enhance not only underwater backscatter networking but potentially influence a broad range of underwater IoT and acoustic communications systems.

References

[1] Sustainable fisheries and aquaculture for food security and nutrition. https://www.fao.org/3/i3844e/i3844e.pdf.

[2] Dana R Yoerger, Annette F Govindarajan, Jonathan C Howland, Joel K Llopiz, Peter H Wiebe, Molly Curran, Justin Fujii, Daniel Gomez-Ibanez, Kakani Katija, Bruce H Robison, et al. A hybrid underwater robot for multidisciplinary investigation of the ocean twilight zone. *Science Robotics*, 6(55):eabe1901, 2021.

[3] Alexander G Rumson. The application of fully unmanned robotic systems for inspection of subsea pipelines. *Ocean Engineering*, 235:109214, 2021.

[4] Jörg Kalwa, A Pascoal, Pere Ridao, A Birk, M Eichhorn, L Brignone, M Caccia, J Alves, and RS Santos. The european r&d-project morph: Marine robotic systems of self-organizing, logically linked physical nodes. *IFAC Proceedings Volumes*, 45(27):226–231, 2012.

[5] John G Proakis, Ethem M Sozer, Joseph A Rice, and Milica Stojanovic. Shallow water acoustic networks. *IEEE communications magazine*, 39(11):114–119, 2001.

[6] John Heidemann, Milica Stojanovic, and Michele Zorzi. Underwater sensor networks: applications, advances and challenges. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 370(1958):158–175, 2012.

[7] Junsu Jang and Fadel Adib. Underwater backscatter networking. In *Proceedings of the ACM Special Interest Group on Data Communication*, pages 187–199, 2019.

[8] Sayed Saad Afzal, Waleed Akbar, Osvy Rodriguez, Mario Doumet, Unsoo Ha, Reza Ghaffarivardavagh, and Fadel Adib. Battery-free wireless imaging of underwater environments. *Nature communications*, 13(1):1–9, 2022.

[9] Waleed Akbar, Ahmed Allam, and Fadel Adib. The underwater backscatter channel: Theory, link budget, and experimental validation. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, pages 1–15, 2023.

[10] Ananya Bhardwaj, Ahmed Allam, Alper Erturk, and Karim G Sabra. Ultrasound-powered wireless underwater acoustic identification tags for backscatter communication. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 2023.

[11] Alper Bereketli. Interference-free source deployment for coverage in underwater acoustic backscatter networks. *Peer-to-Peer Networking and Applications*, 15(3):1577–1594, 2022.

[12] Milica Stojanovic and James Preisig. Underwater acoustic communication channels: Propagation models and statistical characterization. *IEEE communications magazine*, 47(1):84–89, 2009.

[13] Baosheng Li, Shengli Zhou, Milica Stojanovic, Lee Freitag, and Peter Willett. Multicarrier communication over underwater acoustic channels with nonuniform doppler shifts. *IEEE Journal of Oceanic Engineering*, 33(2):198–209, 2008.

[14] Reza Ghanaatian, Orion Afisiadis, Matthieu Cotting, and Andreas Burg. Lora digital receiver analysis and implementation. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1498–1502. IEEE, 2019.

[15] Van-Atta Backscatter Github Repository. https://github.com/signalkinetics/vab.

[16] Aline Eid, Jack Rademacher, Waleed Akbar, Purui Wang, Ahmed Allam, and Fadel Adib. Enabling long-range underwater backscatter via van atta acoustic networks. In *Proceedings of the ACM SIGCOMM 2023 Conference*, pages 1–19, 2023.

[17] Reza Ghaffarivardavagh, Sayed Afzal, Osvy Rodriguez, and Fadel Adib. Ultra-wideband underwater backscatter via piezoelectric metamaterials. In *Proceedings of the ACM Special Interest Group on Data Communication*, 2020.

[18] Waleed Akbar, Ahmed Allam, and Fadel Adib. The underwater backscatter channel: Theory, link budget, and experimental validation. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, pages 1–15, 2023.

[19] Reza Ghaffarivardavagh, Sayed Saad Afzal, Osvy Rodriguez, and Fadel Adib. Underwater backscatter localization: Toward a battery-free underwater gps. In *Proceedings of the 19th ACM Workshop on Hot Topics in Networks*, HotNets '20, page 125–131, New York, NY, USA, 2020. Association for Computing Machinery.

[20] Kai Tu, Dario Fertonani, Tolga M Duman, Milica Stojanovic, John G Proakis, and Paul Hursky. Mitigation of intercarrier interference for ofdm over time-varying underwater acoustic channels. *IEEE Journal of Oceanic Engineering*, 36(2):156–171, 2011.

[21] Baosheng Li, Shengli Zhou, Milica Stojanovic, Lee Freitag, and Peter Willett. *Non-uniform Doppler compensation for zero-padded OFDM over fast-varying underwater acoustic channels*. IEEE, 2007.

[22] Milica Stojanovic. *Low complexity OFDM detector for underwater acoustic channels*. IEEE, 2006.

[23] Pengyu Zhang, Dinesh Bharadia, Kiran Joshi, and Sachin Katti. Hitchhike: Practical backscatter using commodity wifi. In *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*, pages 259–271, 2016.

[24] Vincent Liu, Aaron Parks, Vamsi Talla, Shyamnath Gollakota, David Wetherall, and Joshua R Smith. Ambient backscatter: wireless communication out of thin air. *ACM SIGCOMM Computer Communication Review*, 43(4):39–50, 2013.

[25] Renjie Zhao, Purui Wang, Yunfei Ma, Pengyu Zhang, Hongqiang Harry Liu, Xianshang Lin, Xinyu Zhang, Chenren Xu, and Ming Zhang. Nfc+ breaking nfc networking limits through resonance engineering. 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*, pages 694–707, 2020.

[26] Vamsi Talla, Mehrdad Hessar, Bryce Kellogg, Ali Najafi, Joshua R Smith, and Shyamnath Gollakota. Lora backscatter: Enabling the vision of ubiquitous connectivity. *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies*, 1(3):1–24, 2017.

[27] Milica Stojanovic. Underwater acoustic communication. *Wiley Encyclopedia of Electrical and Electronics Engineering*, pages 1–12, 1999.

[28] European Telecommunications Standards Institute. 5G; NR; Multiplexing and Channel Coding. ETSI TS 138 212 V16.2.0, ETSI, July 2020.

[29] Ido Tal and Alexander Vardy. List decoding of polar codes. *IEEE transactions on information theory*, 61(5):2213–2226, 2015.

[30] ALine Eid, Jack Rademacher, Waleed Akbar, Purui Wang, Ahmed Allam, and Fadel Adib. Enabling long-range underwater backscatter via van atta acoustic networks. In *Proceedings of the ACM SIGCOMM 2023 Conference*, 2023.

[31] TDK PM-50-39 ferrite core. https://www.tdk-electronics.tdk.com/inf/80/db/fer/pm_50_39.pdf.

[32] KEMET Electronics Corporation. Metallized polypropylene film EMI suppression capacitors R46, class X2, 310 VAC, 125°C. https://content.kemet.com/datasheets/KEM_F3122_R46_X2_310_125C.pdf.

[33] T6 Series 400W AC Servo Motor Kit 3000RPM 1.27Nm 17-bit Encoder IP65 – STEPPERONLINE. https://www.omc-stepperonline.com/t6-series-400w-ac-servo-motor-kit-3000rpm-1-27nm-17-bit-encoder-ip65-t6-rs400h2a3-m17s, 2023.

[34] Sayed Saad Afzal, Reza Ghaffarivardavagh, Waleed Akbar, Osvy Rodriguez, and Fadel Adib. Enabling higher-order modulation for underwater backscatter communication. In *Global Oceans 2020: Singapore–US Gulf Coast*, pages 1–6. IEEE, 2020.

[35] Waleed Akbar, Ahmed Allam, and Fadel Adib. The underwater backscatter channel: Theory, link budget, and experimental validation. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, pages 1–15, 2023.

[36] Milica Stojanovic, Josko A Catipovic, and John G Proakis. Phase-coherent digital communications for underwater acoustic channels. *IEEE journal of oceanic engineering*, 19(1):100–111, 1994.

[37] Mark Johnson, Lee Freitag, and Milica Stojanovic. Improved doppler tracking and correction for underwater acoustic communications. In *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 575–578. IEEE, 1997.

[38] Rolf Weber, Andreas Waldhorst, Florian Schulz, and JF Bohme. Blind receivers for msk signals transmitted through shallow water. In *MTS/IEEE Oceans 2001. An Ocean Odyssey. Conference Proceedings (IEEE Cat. No. 01CH37295)*, volume 4, pages 2183–2190. IEEE, 2001.

[39] Trym H Eggen, Arthur B Baggeroer, and James C Preisig. Communication over doppler spread channels. part i: Channel and receiver presentation. *IEEE journal of oceanic engineering*, 25(1):62–71, 2000.

[40] Milica Stojanovic. An adaptive algorithm for differentially coherent detection in the presence of intersymbol interference. *IEEE Journal on selected areas in communications*, 23(9):1884–1890, 2005.

[41] Chengbing He, Jianguo Huang, Qunfei Zhang, and Kaizhuo Lei. Reliable mobile underwater wireless communication using wideband chirp signal. In *2009 WRI International Conference on Communications and Mobile Computing*, volume 1, pages 146–150. IEEE, 2009.

[42] EM Sozer, John G Proakis, and F Blackmon. Iterative equalization and decoding techniques for shallow water acoustic channels. In *MTS/IEEE Oceans 2001. An Ocean Odyssey. Conference Proceedings (IEEE Cat. No. 01CH37295)*, volume 4, pages 2201–2208. IEEE, 2001.

[43] Tommy Oberg, Bernt Nilsson, Niten Olofsson, Magnus Lundberg Nordenvaad, and Erland Sangfelt. Underwater communication link with iterative equalization. In *OCEANS 2006*, pages 1–6. IEEE, 2006.

[44] Jun Won Choi, Thomas J Riedl, Kyeongyeon Kim, Andrew C Singer, and James C Preisig. Adaptive linear turbo equalization over doubly selective channels. *IEEE journal of oceanic engineering*, 36(4):473–489, 2011.

[45] Atulya Yellepeddi and James C Preisig. Adaptive equalization in a turbo loop. *IEEE Transactions on Wireless Communications*, 14(9):5111–5122, 2015.

[46] Yuriy Zakharov, Benjamin Henson, Roee Diamant, Yuan Fei, Paul D Mitchell, Nils Morozs, Lu Shen, and Tim C Tozer. Data packet structure and modem design for dynamic underwater acoustic channels. *IEEE Journal of Oceanic Engineering*, 44(4):837–849, 2019.

[47] Mandar Chitre and Mehul Motani. On the use of rate-less codes in underwater acoustic file transfers. In *OCEANS 2007-Europe*, pages 1–6. IEEE, 2007.

[48] Mingsheng Gao, W-S Soh, and Meixia Tao. A transmission scheme for continuous arq protocols over underwater acoustic channels. In *2009 IEEE International Conference on Communications*, pages 1–5. IEEE, 2009.

[49] Mandar Chitre and Wee-Seng Soh. Reliable point-to-point underwater acoustic data transfer: To juggle or not to juggle? *IEEE Journal of Oceanic Engineering*, 40(1):93–103, 2014.

[50] Shengming Jiang. On reliable data transfer in underwater acoustic networks: A survey from networking perspective. *IEEE Communications Surveys & Tutorials*, 20(2):1036–1055, 2018.