Murphy Also Loves Fish: Lessons Learned from Practical Underwater Communication Trials

Luisa Lux[•○], Eric Wagner[⋆], Ulrike Meyer[○], and Jan Bauer[•],
[•]Fraunhofer FKIE ○RWTH Aachen University ⋆University of Luxembourg
Wachtberg, Germany Aachen, Germany Luxembourg, Luxembourg
{<firstname>.<lastname>}@fkie.fraunhofer.de
{<lastname>}@itsec.rwth-aachen.de {<firstname>.<lastname>}@uni.lu

***Abstract*—Even though underwater communication evolves from a niche topic into an area of significant political interest, researchers without prior experience in practical trials may be reluctant to venture into this branch of research, as a lack of experience with the common pitfalls can lead to a waste of time, energy, and money. Having gone through this experience firsthand, this work sumarizes the lessons learned from one year of lake trials to serve as a guide for other research groups undertaking similar projects. We are specifically targeting this work at newcomers in order to help make this branch of research more accessible to research groups that possess the necessary background knowledge to advance underwater networks but lack experience with practical deployments.**

***Index Terms*—Underwater Acoustic Networks; Practical Trials.**

I. Introduction

Underwater communication has been an ongoing topic of interest in designated research communities for decades, with early research activities arising with the emergence of SONAR technology [1]. However, global geopolitical changes in recent years and phenomena caused by climate change, such as shorter Arctic winters, have been leading to an increased strategic interest in regions on earth that are primarily of maritime nature and only accessible with equipment that can operate above and below the surface. Suddenly, underwater communication and even networks call for a transformation from a niche research topic to a field of high political interest.

However, despite its attractiveness, entering the academic world of underwater communication is fraught with hurdles. Practical sea trials are time-consuming, costly, and error-prone. While simulators adapted to the features of underwater networks, such as MATLAB or the ns2-based DESERT simulator [2], remedy the need for practical tests in the early stages of a new research idea, but they cannot entirely replace practical testing. Within research groups that have been active in this field for decades, the necessary expertise on this matter has been built up and maintained. Newcomers without the experience gained in years of practical trials, however, are often faced with countless pitfalls that can exorbitantly delay the success of deployment, leading to a waste of time, energy, and money, or the strategic decision to not step into this research field. As we have experienced these struggles first-hand, this work is dedicated to providing others with the experience that we have compiled through one season of practical lake trials. It is primarily addressed to newcomers in the underwater research field, but certainly entertaining for anyone with or without experience in practical testing.

II. Related Work

Publications describing challenges and experiences gained from practical outdoor deployments are no rarity. The first of its kind and to this day one of the most influential ones is the work *Murphy loves potatoes* by Langendoen et al. [3] from 2006. Their description of setbacks and challenges during the deployment of a large-scale sensor network for monitoring of a potato field during the growing season gives an honest and representative impression of the difficulties that are faced during the deployment process. This explains why a non-negligible series of works have abetted their respective research field by sharing their meta-experiences instead of simply concentrating on mere results. While in particular the area of smart agriculture has produced a variety of publications focused on their lessons learned and experiences [4]–[9], we have also found works describing operational challenges in smart home monitoring [10], public infrastructure such as water quality testing [11], community wireless networks [12] and IoT sensing [13] as well as extremely hostile environments such as construction sites [14], planetary exploration [15], high-arctic [16] and rainforest areas [17], as well as publications about practical deployment of Wireless Sensor Networks (WSNs) in general [18]–[20].

However, particularly in the field of underwater research, in which practical experiments play a considerable role, the number of works reporting on deployment challenges or providing guidelines for such turns out to be sparse. To the best of our knowledge, only the works by Voigt et al. [21] and Lazzarin et al. [22] provide insights into problems that occurred during practical trials. Yet, these works are not addressed to newcomers in the field. As such, they are missing a detailed collection of pitfalls that will arise and guidance on how they can be avoided with only low-cost equipment and no prior experience with in-situ hardware deployments.

III. Opening a Can of Worms – Attempting a Wormhole Attack

The idea that had been driving a season full of lake tests was to implement a *wormhole attack* in a real-world deployment

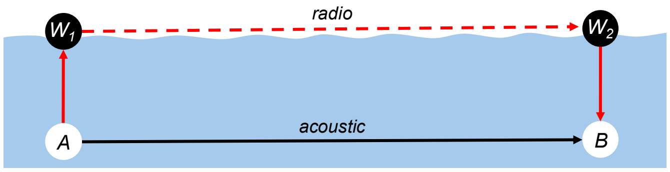


Fig. 1. Wormhole scenario comprising four nodes (A, B, W_1, W_2) with W_1, W_2 having additionally a surface-level wormhole link established, cf. [23].

as a practical proof-of-concept. Details on our results and an extended analysis regarding the parameters enabling the attack in practice can be found in [23]. Such an attack is based on an out-of-band link, i.e., the wormhole, that out-speeds regular communication within the targeted network [24], [25] and was originally introduced in the more general context of Mobile Adhoc Networks (MANETs) [26].

However, in contrast to terrestrial (MANET) communication, water as a propagation medium is not suitable for communication with electromagnetic waves. At best, communication ranges of a few meters can be achieved, which is insufficient for most application scenarios [27]. Thus, underwater communication is dominated by the use of acoustics, which enables communication ranges of kilometers [28]. However, the acoustic channel also introduces some major challenges such as very high latency (due to the relatively low propagation speed of sound), narrow bandwidth, and a high noise-level [28]. Thus, the so-called Underwater Acoustic Networks (UANs) are not only a striking example of resource-constrained, lossy networks, but are also vulnerable to wormhole attacks. Since underwater cables or radio surface channels offer an adversary a channel with low latency, wormholes appear to be straightforward to implement in UANs.

To realize such an attack setting, we have envisioned a simple scenario as depicted in Fig. 1 to deploy in a nearby lake comprising four nodes A, B, W_1, and W_2 all of which are within communication range of each other. Furthermore, we assume W_1 and W_2 have installed a wormhole link, in our scenario a radio surface connection, between each other (dotted red path). Our goal was to log an event in which a message broadcasted by A routed over the wormhole (W_1, W_2) would be received sooner at B than over the direct underwater link between A and B. While the propagation speed of sound is five orders of magnitude lower than that of electromagnetic signals on the surface, additional delays were introduced on the wormhole path due to two additional hops. The challenge was to find a sufficiently large distance between A and B, in which the delays introduced on the wormhole path would be smaller than the propagation delay on the underwater link. As we estimated this distance to be between 150–250 m, we knew right from the beginning that a deployment in a lake rather than in a swimming pool was inevitable.

IV. Deployment Setup

In the following, we describe our basic deployment setup regarding hardware and software, as we commissioned it at the beginning of our trials in April 2025. The progressive improvements of this setup based on the experience gathered during the lake trials are discussed alongside our lessons learned in Sec. V-A.

A. Hardware

Each of our nodes comprises a set of hardware building blocks that are shown in Fig. 2 and described in the following.

1) Single-board Computer: The acquired single-board computer was a RaspberryPi 5. It can be powered over a USB-C port and has 4 USB-A ports for connecting different peripherals. It supports Wi-Fi 802.11ac and Bluetooth.

2) Underwater Modem: While getting the RaspberryPi (RPi) was somewhat trivial, choosing the right combination of modem and hydrophone is anything but straightforward. There are numerous low-cost, academic acoustic modems for research applications, as summarized, e.g., by Campagnaro et al. [29] However, this selection was quickly reduced by an important selection criterion: Our modem had to be *open-source* to support necessary changes, and with adequate documentation for a smooth commissioning. Finally, we found all of that in the ahoi! modem [30] (≈ 800 EUR/unit).

3) Hydrophone: As Renner et al. [30] report the AS-1 hydrophone [31] to be compatible with the modem and the AS-1 also being a prominently used hydrophone in other works as well [32]–[34], we also purchased 4 hydrophones for roughly 400 EUR each. The AS-1 hydrophone has a cable length of 10 m, a BNC connection, and provides omnidirectionality.

While these prices might not appear low-cost at first glance, they are small compared to the costs of commercial and ruggedized underwater equipment.

B. Software & Parameter Configuration

1) Surface Communication Module: Early on, we decided on LoRa to be a suitable surface communication, because it is commonly used in marine applications and able to communicate point-to-point without the need for a base station or access point [35], [36]. We used the LoRa HAT by Waveshare [37] operating in the 868 MHz frequency band.

2) GNNS Receivers: The GPS receivers that we used were part of the internal institute's hardware collection. We used two devices of the model Royaltek RGM-3800, which could be polled over a USB connection to the RPi. Even though the receivers could be powered via USB, they work more reliably

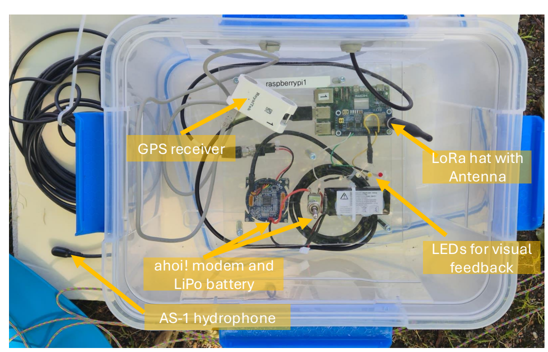


Fig. 2. Hardware setup of node B in splash-proof case in lake trials #7 – #11.

when provided with some in-battery power, such that they would stay turned on, even when the RPi was switched off, resulting in a stable reception without having to re-search for a position fix after reboot of the RPi.

3) Additional Hardware Components: Besides these main components, we also needed some additional hardware, including power sources and cables for a serial connection between RPi, modem, and hydrophone. The power sources we used were LiPo batteries for the modem, which already came with a BEC connector. The RPis were powered with 10,000 mAh 5 V-3 A power banks.

Finally, we also opted to solder a switch to the LiPo batteries to lower the risk of breaking the BEC connectors on the circuit board due to repeatedly connecting and disconnecting the battery to switch off the modem. This also lowered the risk of mixing up positive and negative connectors during switch-on and switch-off. Equally optional but advisable is the decision to screw the hardware components to a plexiglass panel to avoid moving the components during the experiment and thereby tearing the extremely thin cable connecting the modem and the hydrophone.

As the ahoi! modem is open-source, its firmware can be individually configured to the needs of the operator. Furthermore, the sole operation of the modem without configuration of the default firmware can be done via a serial connection to the RPi with a platform-independent Python API. Thus, this modem is very intuitive to use even without prior experience. Similarly, communication via the LoRa HAT can be managed via Python scripts on the RPi. Those scripts can be started via an ssh connection established using, e.g., a smartphone WiFi hotspot at the deployment site, to which the RPis were configured to connect automatically.

The ahoi! message format consists of a preamble, a header of 6 bytes, a payload of arbitrary length and a CRC. For our experiments, we soon realized that the smallest possible payload size of 1 B + 1 B CRC would be advantageous. Also, we needed the data rate of the modem to be as high as possible, which is configuration-dependent and ranges between 250–4500 bps. To exploit the highest data rate, we used the high-speed wide-band configuration with spreading factor S=1. These settings can be configured by changing the respective FLAGS in the CONFIG file of the modem's firmware.

C. Code Procedure

The intended procedure of the wormhole scenario was as follows: In our experiments, node A, acts as the sender, and sends a message with a payload of 1 B every 5 s. Once A sends a message, the hydrophone of W_1 picks it up, the modem processes it, and then sends it over its serial connection to the RPi, which broadcasts the message over its LoRa interface. After receiving a message from W_1 via its LoRa HAT, the message is forwarded by W_2 over its serial connection to the modem, which then transmits it acoustically via its hydrophone. Finally, node B constantly listens on its acoustic channel for messages sent by either W_2 or B. With sufficient distance between A and B (i.e., with the necessary delay), we assumed that the message being forwarded by W_1 and W_2 would arrive earlier at B than the direct message of A, cf. Fig. 1.

V. Lessons learned

In this section, we unfold the progress of the lake trials over the course of half a year in detail. To encourage other research groups with practical experiments, we not only describe the experienced hurdles, but also guide how problems and pitfalls were solved or avoided in the following. We start with a small overview (Sec. V-A) before structuring our findings into common mistakes in field trials (Sec. V-B), lessons learned about underwater deployment setups (Sec. V-C), lessons learned about the ahoi! hardware (Sec. V-D).

A. Chronological Overview of Lake Trials

In total, we needed 11 lake trials over the course of eight months to obtain a well-documented practical Proof-of-Concept (PoC) of the wormhole attack as envisioned (cf. Tab. I). The journey there was anything but straightforward. From April to July, our struggles were primarily driven by the deployment being outside, operating without technical infrastructure. While these initial hurdles felt like wasted time, those pitfalls were mostly easy to overcome as the underlying problem was often quite obvious.

During the second period, starting in July until September, our setup had already gone through some iterations, resulting in a lot more stability. During this time, we accumulated valuable knowledge both about the challenges of underwater communication in practice and also parameter and hardware specifics of the ahoi! modems. Only during the last two of our lake trials, the wormhole attack, as we envisioned it almost half a year earlier, succeeded. During the process, however, we were sometimes so frustrated that we did not even know if there would ever be a *next* lake trial (cf. Fig. V-A).

B. Common Mistakes – Outdoor Experiements

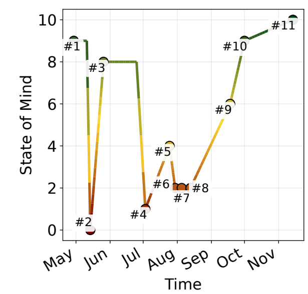
When we started with lake trials in April 2025, our hardware experience was essentially limited to our preliminary laboratory tests. This resulted in an iterative process to stabilize our outdoor test setup, such that the experiences gathered within this first phase are not necessarily underwater-related.

1) No Power, No Progress: Even before the first lake trials, we already ran into some power supply issues: After a productive lab day, one of the modems was forgotten to be switched off, resulting in the deep discharging of one of the LiPo batteries. It needed replacing. While this was something we reliably double-checked from then on, we had bad luck with power problems once again at our second sea trial. Having naively used power banks from our private collection during the first lake trial without any problems to reduce procurement times, we started our second lake trial even more motivated – and this time, more prepared with GPS receivers to track distances and locations of the nodes. Unfortunately, this increased USB periphery at the RPis needed more starting current than before, resulting in the RPis booting up and then

TABLE I
CHRONOLOGICAL OVERVIEW OF 11 CONSECUTIVE LAKE TRIALS IN 2025.

Trial	Date	Changes	Problems	Reason
#1	04/29	First Messages sent and received	–	–
#2	05/14	GPS for distance tracking	Battery Problems	voltage drops by powerbanks
#3	05/26	two nodes deployed on lake	LoRa signal often lost	LoRa antennas too small
#4	07/03	Introduction of HaLow	Fails with Halow	manual activation needed
#5	07/25	Back to LoRa, additional LED feedback, wormhole setting	no wormholing	Wrong bit-spread, AGC, signal clipping
#6	07/30	Long script running Lab-Tests	rain flooding hardware	no casing
#7	08/05	new Casing	Deafing of nodes	Wrong bit-spread, AGC, signal clipping
#8	08/13	Debugging of RPi (incl. CPU/RAM usage)	Deafing of nodes	Wrong bit-spread, AGC, signal clipping
#9	09/18	Parameter change and static RX gain	no signal on short links	signal clipping
#10	10/01	Increasing distance on short links	no GPS signal	GPS Polling Bug
#11	11/14	GPS Polling adapted	–	–

State of mind throughout the 2025 season.



shutting down shortly after, because the power banks were not able to maintain an output voltage above 4.7 V during the booting. After this, we acquired power banks of the model with 10,000 mAh each and 30 W, supporting 5 V-3 A. They were able to reliably power our RPis throughout $\approx$6 h.

However, another workaround would have been to boot up the RPi without USB peripherals and plug in the necessary USB devices only after the RPi has completely booted. This avoids an extreme current peak at startup and helps if the voltage drop is caused by an elevated inrush current, rather than by the current required for continuous operation.

2) Automation Is Key: Our setup included Python scripts on the RPis that could be started and monitored via ssh from the Laptop. However, to be accessible over ssh, the RPis needed to be in range of the hotspot. So, whenever they needed restarting or debugging, they would need to be transported close to the smartphone providing the hotspot. As we were operating in distances of a few hundred meters, this started to become inconvenient very soon, forcing us to utilize cron jobs to automatically start scripts whenever the RPi needed a reboot, which came in handy in situations where it seemed like the device had frozen.

3) Flying Blind Is Not an Option – Creating a Suitable Control Channel or Feedback Loop: By automating Script-start-up, we already avoided having to bring the RPis back in hotspot range as soon as they needed restarting. However, we were still completely blind about their internal status if they were outside WiFi ssh range. This was not desirable as we could spend hours thinking that we were collecting valuable data – and in reality, the scripts hung for the entire time due to some dubious reason that could have been fixed if it were visible. We thus decided that we needed some form of control channel. We first started to use periodic LoRa messages to the shore during the third sea trial. However, this was not ideal as the LoRa connection was often lost because the shore was too far away. As this left us again often blind, we switched to HaLow for the fourth sea trial. While HaLow would have even provided us with the luxurious option to access the RPis per ssh over a range of a few meters, the HaLow modules did not turn out to be very robust. Hardware-wise, their USB cable jacks would fall off with the smallest movement, and they were somewhat buggy, being stuck during the boot process and thus having to be manually activated over the serial connection to the RPi - again requiring them to be in hotspot range. Being extremely frustrated by this (cf. Fig. V-A, lake trial #4), we thought about what the simplest solution for such a feedback loop would look like without adding even more complexity in a setting that did not run too stably. Finally, we decided to use three LEDs on node B as a limited but very simple visual feedback (cf. Fig 2). One LED was intended to indicate, when lighting up, whether the RPi did not receive a LoRa signal for the last 15 s, one that indicated whether it did not receive underwater messages from A for 15 s, and one that indicated a successful wormhole, meaning that both messages were received correctly in the right order. Even though this was not much information, it was sufficient to indicate whether the setup worked and, if not, which of the links caused the problem, if we needed to bring the nodes closer together, and whether the attack was carried out successfully when everything looked to work as planned.

4) Testing, Testing, Testing: Even though Langendoen et al. [3] have already highlighted this issue 20 years ago, the importance of thorough testing cannot be stressed enough. There were bugs that would only occur during lake trials when the hardware was disassembled, wrapped up, restarted, etc., introducing situations that would just not take place in the laboratory. Syntax errors occurred, for instance, because the USB port to the modem or the GPS receiver was renamed by the system after a restart. Then, some bugs would become evident only when the setup was run for a longer time (at least a few hours). E.g., we found out that our scripts were causing RAM and CPU overload after some time due to continuous loops without proper sleep commands.

One time, we created a bug ourselves. When integrating the LEDs for feedback on underwater and LoRa reception, we accidentally wired them to GPIO pins that were already assigned to LoRa reception. This often resulted in almost no LoRa packet reception as soon as the LED was turned on. We could not explain this phenomenon and came up with wild hypotheses, such as the enforcement of LoRa duty cycling by the firmware. This issue cost us some serious nerves until we found out that we were barking up the wrong tree completely.

The last mistake falling into this category, ironically, occurred at the lake trial #10, when we were already pretty used to universal testing. When looking at the wormhole attack data, we found that something with the data polling from the GPS devices must have gone wrong. Instead of correctly polling the latest GPGGA data every second, the script would poll the serial connection line by line. As the GPS device not only provides GPGGA data, but also GPGSA, GPGSV, and GPRMC recordings, this resulted in several lines per second. A backlog of lines was created, and thus the real-time and the GPS time would gradually drift apart until the data polled from the GPS device was almost two hours outdated. Since the devices often do not find a GPS fix inside the lab, the incorrectness of the GPS data was not apparent beforehand.

The only thing that helps against such bugs, and that is to simulate outdoor trials in the lab with as much detail as possible, creating similar friction as would occur during trials.

5) Rainfall – When Electronics Meet the Elements: One mistake that could have cost us dearly was the lack of splash-proof casing, which we were desperately missing on the lake trial #6 on July 30, 2025, when we were caught in a heavy downpour. For some reason, we were sailing pretty smoothly in that regard beforehand, even though the water was always very close. On this day, however, the hardware was drained of water so that we could not get it booting again. Fortunately for us, this damage turned out to be only temporary. After drying in the laboratory overnight, everything started working again the next day, and no heavy corrosion had occurred. Still, we were humbled by that experience and invested in splash-proof boxing (cf. Fig. 2) that kept our hardware nice and dry.

C. Lessons Learned – Underwater Deployment Setups

Running parallel to the attempts to stabilize the hardware setup for outdoor testing over the course of a few hours, a second thread of experiences made during the first trials from April to July related to work above and in water.

1) Lost in Transmission: Working with underwater hardware during several lake trials, we were already expecting that signal quality is extremely dependent on the concrete environment as frequently described in literature [27], [28], [30]. Thus, it required some trial and error to work out regions that were suitable for the testbed. Not only does transmission range alone play a decisive factor. As can be seen in Fig. 3, the signal strength received in different areas within the same lake is quite different. As a rule of thumb, we started to avoid shallow waters, which we assumed to lead to multipath propagation, muddy underground resulting in sound absorption, and boat landings that likely cause a lot of echo, thus resulting in destructive interference. Regarding the bathymetry of a lake, investigating online sources such as diving maps can provide hints as to why certain areas are more or less suitable for testing purposes. To access deep waters in the middle of the lake, we tied down our boxes on floating 1×1 m styrofoam islands. The islands of nodes A, W_1 and W_2, B were each roped together, so that they would remain considerably close together. To keep node A and node W_1 in a static position, we used a buoy in the middle of the lake already moored to the lake ground to which the two islands were attached (cf. Fig. 4). The position of node B and W_2, on the other hand, needed to be dynamic to flexibly adapt the distance between A and B. We thus used a stand-up paddling (SUP) board, on which we attached node B and the island of node W_2, that could be moved around the lake (cf. Fig. 4). While this setup did not provide us with the possibility to access the RPis of B via Laptop on the lake continuously, the LEDs gave us some limited feedback on the deployment, whereas the SUP board provided a low-cost option to access the lake and has a lower noise footprint than a boat.

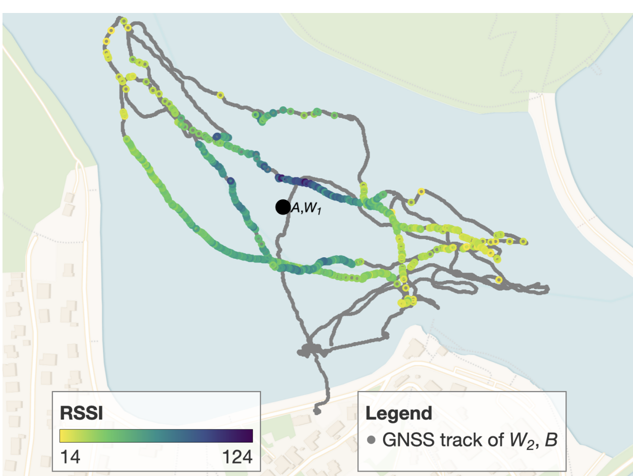


Fig. 3. Reception events and received signal strength indicator (RSSI) of received messages from A by B at different positions of B.

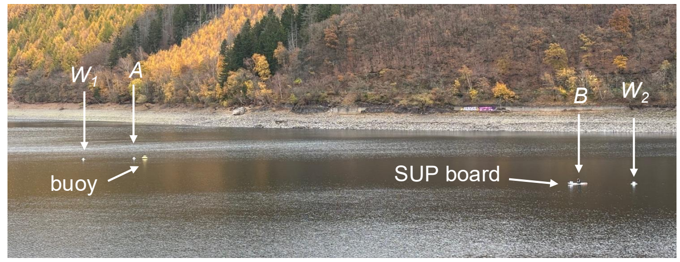


Fig. 4. Deployment of nodes A, W_1 roped together and attached to buoy, movement of nodes W_2, B on the lake via a paddleboard. Details can be found in Tab. II in the Appendix.

2) Small Antennas, Big Problems: Working in and above water not only has an influence on the underwater links, but also on the LoRa connectivity. Concretely, we experienced a much smaller communication range than was reported in the data sheets (≈ 5 km). We trace this back to the small antennas sitting right above the water surface, causing a high amount of reflection. Using slightly taller antennas, we noticed that this effect was somewhat softened. However, LoRa range still remained smaller than reported (≈ 250–300 m vs. 5 km).

3) Bugs That Only Swim: In general, debugging turned out to be quite cumbersome. In case something was not working (e.g., the modems do not receive anything during the lake trial), the lack of equipment at the lake made in-depth debugging impossible. On the other hand, sound signals are extremely sensitive to the transmission environment. Thus, it was difficult, if not impossible, to reproduce phenomena observed at the lake in the laboratory later on. E.g., one issue that we noted when working with the modems in the laboratory was that they could *hear* each other over air, when the cable was not completely submerged. This led to high reception rates in the lab, but was, of course, non-representative for trials.

One implementation detail that proved useful in this context was keep-alive messages on the LoRa link, in case node W_1 would not receive acoustic messages from A for a pre-defined amount of time. Since our setup consisted of three links connected in series, the loss of a message on one of the links also meant that no messages were received on the subsequent links. Via keep-alive messages between W_1 and W_2 that were also forwarded acoustically to B upon reception, we were able to localize which of the links were affected by connectivity issues and which ones were not working simply because of their preceding connections.

4) Hello Bureaucracy, Dear Friend: During the winter, lakes are often granted a resting time, during which on-water activities are prohibited. Only because of that, we changed the territory for the last sea trial on November 14. At the same time, we also applied for a special permit to deploy during the winter season in our original lake, which involved the local nature conservation authority, the local water authority, and the operating company of the lake. Fortunately, this permission was granted, as research activities are typically excluded from off-season regulations.

D. Lessons Learned – ahoi! Modem

As the lake trials progressed, the setup worked relatively stably. Still, the summer trials at the end of July and August were characterized by extremely low reception on all three underwater links. Everything appeared to work well until the hydrophones were submerged underwater. After that, the sound signal was lost and was only regained briefly after reboot of RPi and the modem before it was lost again. Having excluded the possibility of any external factors like crashed software scripts or sound-absorbing phenomena, we started to dig deeper into the parameters and firmware of the modem. Note that the final modem configuration parameters for our setup can be found in Tab. II in the Appendix.

1) Fast and Furious vs. Safe and Sound: The envisioned attack setting led us to fully utilize the available data rate with the settings presented in Sec. IV-B. At the same time, our setting required us to communicate at ranges that were on the upper end of the reported ahoi!'s transmission. Choosing a bit-spread parameter of S=1 (no bit redundancy) thus led to a very high data rate, but also extremely low robustness. This caused a total packet loss on the long underwater link, which also confirms the measurements in [30]. Unfortunately, in the beginning, the number of parameters and features of the hardware was quite overwhelming, and only during the continuous work with the modems did we slowly start to understand the purpose and effect of each of the settings. Revising the parameter settings set in the beginning, such as the bit-spread parameter to S=3, then flipped the switch.

2) Too Much, Too Little, Just Right – Gain Balancing: Analyzing the data from our unsuccessful lake trials, we observed a phenomenon which was akin to the modems going *deaf* after the hydrophones were submerged in deep water. Even though we have no structural testing results confirming this hypothesis, we assume this was partly caused by the automatic gain control (AGC) of the modems' firmware, which is turned on by default. This is also supported by the results achieved as soon as we changed both the bit-spread parameter and the RX gain control, resulting in a stable packet reception on the long link at the lake trial #9 (cf. Tab. I).

3) Too Loud, Too Close: As we managed to get the more error-prone long underwater link to work, while the short links were still a problem, we assumed that the distance on the short link might have been *too short*. As we found out by investigating the ahoi! website, series $< v7.1$ have a hardware-bug that disables the adaptability of the TX gain. While this is advantageous on long links, we assumed that this can lead to extreme signal clipping in short distances of 1–2 m. This could also not be remedied by the use of extremely low RX gains. The straightforward and pragmatic way out was thus to increase the length of the short links from 1 m to 3–5 m. Our hypothesis turned out to be correct as this led to a series of successful wormhole events on the lake trial #10 (cf. Tab. I).

4) The Beat of Serial Communication: Even though the wormhole attack, as envisioned, succeeded after all, there is still room for improvement when working with the modems in a time-restricted fashion. Analyzing the measured data, we found out later that the serial connection between the modem and RPi works in a cyclic fashion of ≈ 16 ms, leading to a discretization of reception timestamps at the RPi of node B. In future deployments, we would thus either log reception timestamps on the modem and transmit them in the packet footer or work with the packet flag that is output on one of the IO pins and enables toggling upon pre-defined events. Both options would enable a much more precise time-logging of message reception events.

VI. Conclusion

With maritime surveillance becoming increasingly important due to the expansion of offshore infrastructure, geopolitical shifts, and the chase for resources in regions that are becoming more accessible due to climate change, underwater communication transforms from a niche academic field to a broader research area. Trends like the emergence of more and more low-cost underwater hardware for research purposes foster this development. However, the lack of experience for practical underwater testing, resulting in increased resources, might discourage novice groups from joining this research area. To facilitate entry into this area and to make collective expertise more accessible, this work aims to provide detailed descriptions of problems and working solutions for a series of lake trials over the course of half a year that are representative of the initial challenges faced due to the lack of expertise with underwater setups. In that, we hope to provide support to anticipate and circumvent initial hurdles smoothly.

References

[1] R. Urick, *Principles of Underwater Sound*. 3rd ed. McGraw-Hill, 1983.

[2] F. Campagnaro *et al.*, "The DESERT underwater framework v2: Improved capabilities and extension tools," in *Proc. of the 3rd IEEE Underwater Communications and Networking Conference (UComms)*, Lerici, Italy, 2016, pp. 1–5.

[3] K. Langendoen, A. Baggio, and O. Visser, "Murphy Loves Potatoes: Experiences from a Pilot Sensor Network Deployment in Precision Agriculture," in *Proc. of the 20th IEEE International Parallel & Distributed Processing Symposium (IPDPS)*, Rhodes, Greece, 2006, pp. 1–8.

[4] J. Bauer and N. Aschenbruck, "Design and Implementation of an Agricultural Monitoring System for Smart Farming," in *Proc. of the IEEE Internet of Things Vertical and Topical Summit for Agriculture – Tuscany (IOT Tuscany)*, Monteriggioni (Siena), Italy, 2018, pp. 1–6.

[5] M. O. Ojo *et al.*, "Practical Experiences of a Smart Livestock Location Monitoring System Leveraging GNSS, LoRaWAN and Cloud Services," *Sensors*, vol. 22, no. 1, 2022.

[6] W. Hu *et al.*, "Outdoor Sensornet Design and Deployment: Experiences from a Sugar Farm," *IEEE Pervasive Computing*, vol. 11, no. 2, pp. 82–91, 2012.

[7] P. Kumar and S. Reddy, "Lessons Learned From the Deployment of Test-Bed for Precision Agriculture," in *Proc. of the Int. Conf. on Sustainable Computing in Science, Technology and Management (SUSCOM)*, Jaipur, India, 2019, pp. 134–142.

[8] M. M. McIntosh *et al.*, "Deployment of a LoRa-WAN near-real-time precision ranching system on extensive desert rangelands: What we have learned*," *Applied Animal Science*, vol. 39, no. 5, pp. 349–361, 2023.

[9] R. Hartung *et al.*, "On the Experiences with Testbeds and Applications in Precision Farming," in *Proc. of the 1st ACM Int. Workshop on the Engineering of Reliable, Robust, and Secure Embedded Wireless Sensing Systems (FAILSAFE)*, Delft, Netherlands, 2017, pp. 54–61.

[10] A. Elsts *et al.*, "Internet of Things for smart homes: Lessons learned from the SPHERE case study," in *Proc. of the Global Internet of Things Summit (GIoTS)*, Geneva, Switzerland, 2017, pp. 1–6.

[11] T. L. Dinh *et al.*, "Design and Deployment of a Remote Robust Sensor Network: Experiences from an Outdoor Water Quality Monitoring Network," in *Proc. of the 32nd IEEE Conf. on Local Computer Networks (LCN)*, Dublin, Ireland, 2007, pp. 799–806.

[12] B. Braem, C. Blondia, and S. Latré, "Experiences from building an outdoor testbed for community wireless networks," in *Proc. of the IFIP/IEEE Int. Symposium on Integrated Network Management (IM)*, Ottawa, ON, Canada, 06 2015, pp. 898–901.

[13] X. Fafoutis *et al.*, "Experiences and Lessons Learned From Making IoT Sensing Platforms for Large-Scale Deployments," *IEEE Access*, vol. 6, pp. 3140–3148, 2018.

[14] T. Fernandez-Steeger *et al.*, ""And Then, the Weekend Started": Story of a WSN Deployment on a Construction Site," *Journal of Sensor and Actuator Networks*, vol. 2, no. 1, pp. 156–171, 2013.

[15] D. Sanz *et al.*, "Wireless sensor networks for planetary exploration: Experimental assessment of communication and deployment," *Advances in Space Research*, vol. 52, no. 6, pp. 1029–1046, 2013.

[16] M. Chang and P. Bonnet, "Monitoring in a High-Arctic Environment: Some Lessons from MANA," *IEEE Pervasive Computing*, vol. 9, no. 4, pp. 16–23, 2010.

[17] T. Wark *et al.*, "Springbrook: Challenges in developing a long-term, rainforest wireless sensor network," in *Proc. of the Int. Conf. on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, Sydney, NSW, Australia, 2008, pp. 599–604.

[18] T. Laukkarinen *et al.*, "Pilot studies of wireless sensor networks: Practical experiences," in *Proc. of the Conf. on Design & Architectures for Signal & Image Processing (DASIP)*, Tampere, Finland, 2011, pp. 1–8.

[19] M. Brzozowski and P. Langendoerfer, "Bug-Tolerant Sensor Networks: Experiences from Real-World Applications," in *Proc. of the Int. Conf. on Ad Hoc Networks (ADHOCNETS)*, San Remo, Italy, 2015, pp. 251–262.

[20] E. Gaura *et al.*, *Wireless Sensor Networks: Deployments and Design Frameworks*, 2010, vol. 1, ch. Learning from Deployment Experience, pp. 15–50.

[21] T. Voigt *et al.*, "Sensor Networking in Aquatic Environments – Experiences and New Challenges," in *Proc. of the 32nd IEEE Conf. on Local Computer Networks (LCN)*, Dublin, Ireland, 2007, pp. 793–798.

[22] J. Lazzarin *et al.*, "Underwater Acoustic Data Transmission in the Presence of Challenging Multipath Conditions and Shadow Zones: Sea Trial Analysis and Lessons Learned," *Electronics*, vol. 15, no. 2, 2026.

[23] L. Lux *et al.*, "A deep dive into wormhole attacks in underwater acoustic communication: From theory to practice," in *Proc. of the 19th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*, Saarbrücken, Germany, 2026, p. 256–267.

[24] J. Kong *et al.*, "Low-cost attacks against packet delivery, localization and time synchronization services in under-water sensor networks," in *Proc. of the 4th ACM Workshop on Wireless Security (WiSe)*, 2005, pp. 87–96.

[25] R. Zhang and Y. Zhang, "Wormhole-Resilient Secure Neighbor Discovery in Underwater Acoustic Networks," in *Proc. of the 29th Conf. on Information Communications (INFOCOM)*, San Diego, CA, USA, 2010, pp. 2633–2641.

[26] A. Nadeem and M. P. Howarth, "A Survey of MANET Intrusion Detection & Prevention Approaches for Network Layer Attacks," *IEEE Comm. Surveys & Tutorials*, vol. 15, no. 4, pp. 2027–2045, 2013.

[27] A. Pal *et al.*, "Communication for Underwater Sensor Networks: A Comprehensive Summary," *ACM Transactions on Sensor Networks (TOSN)*, vol. 19, no. 1, Dec. 2022.

[28] A. Boukerche and P. Sun, "Design of Algorithms and Protocols for Underwater Acoustic Wireless Sensor Networks," *ACM Computing Surveys*, vol. 53, no. 6, pp. 1–34, Nov. 2021.

[29] F. Campagnaro, F. Steinmetz, and B.-C. Renner, "Survey on Low-Cost Underwater Sensor Networks: From Niche Applications to Everyday Use," *Journal of Marine Science and Engineering*, vol. 11, no. 1, 2023.

[30] B.-C. Renner, J. Heitmann, and F. Steinmetz, "ahoi: Inexpensive, Low-power Communication and Localization for Underwater Sensor Networks and μAUVs," *ACM Transactions on Sensor Networks (TOSN)*, vol. 16, no. 2, Jan. 2020.

[31] Aquarian Hydrophones, "AS-1 Hydrophone," last accessed: May 2026. [Online]. Available: https://www.aquarianaudio.com/as-1-hydrophone.html

[32] F. Campagnaro *et al.*, "A Software-Defined Underwater Acoustic Modem for Everyone: Design and Evaluation," *IEEE Internet of Things Magazine*, vol. 6, no. 1, pp. 102–107, 2023.

[33] A. Montanari *et al.*, "Experimenting Various JANUS Frequency Bands with the Subsea Software-Defined Acoustic Modem," in *Proc. of the 7th Underwater Communications and Networking Conference (UComms)*, Sestri Levante, Italy, 2024, pp. 1–5.

[34] D. Cosimo *et al.*, "Comparative Analysis of Throughput Maximization Strategies in Underwater Acoustic Networks: Results from At-Sea Experiments," in *Proc. of the IEEE Int. Workshop on Metrology for the Sea; Learning to Measure Sea Health Parameters (MetroSea)*, Portorož, Slovenia, 2024, pp. 52–57.

[35] G. Schneider, M. Goetz, and I. Nissen, "Design and Implementation of a Gateway Buoy for the Underwater-IoT," in *Proc. of the 48th IEEE Conf. on Local Computer Networks (LCN)*, Daytona Beach, FL, USA, 2023.

[36] D. Magrin *et al.*, "Collaboration of LoRaWAN and Underwater Acoustic Communications in Sensor Data Collection Applications," in *Proc. of Global Oceans 2020: Singapore – U.S. Gulf Coast*, Biloxi, MS, USA, 2020.

[37] Waveshare, "SX1262 868M," last accessed: May 2026. [Online]. Available: https://www.waveshare.com/sx1262-868m-lora-hat.htm

Appendix

TABLE II
SETUP AND FINAL MODEM CONFIGURATION.

Setup	Value	Parameter Configuration*	Value
Outside temperature	15 °C	Hardware version	`v7.0`
Water depth lake	2–18 m	Transmitter	`TXB`
Depth hydrophones	2–5 m	Hardware platform	`stm`
Distance A, B	15–270 m	Symbol duration T_s	1.28 ms
Distance A, W_1	5–10 m	Bits per symbol C	6 bits
Distance W_2, B	5–10 m	Spreading factor S	3
Dist. A/W_1, shore	100–150 m	Automatic gain control (AGC)	`OFF`
		Transmit Gain at A, M_2	18
Packet length	1 B	Receive gain at M_1	0
Guard times between packets	5 s	Receive Gain at B	18
		Preamble Length P_s, P_r	16, 11
		Status flags A, R, E	0

*All parameter variables are named according to [30].