

A Real-Time Underwater Acoustic Telemetry Receiver With Edge Computing for Studying Fish Behavior and Environmental Sensing

Yang Yang¹, Member, IEEE, Robbert Elsinghorst², Jayson J. Martinez³, Hongfei Hou⁴, Jun Lu⁵, Member, IEEE, and Zhiqun Daniel Deng⁶

Abstract—Underwater acoustic telemetry has emerged as a powerful tool for practical applications, including resource exploration, environmental monitoring, and aquatic animal tracking. However, current acoustic telemetry systems lack the capability to transmit the collected data continuously in real time, primarily because the acoustic networking bandwidth is limited. Retrieval of the recorded measurements from the deployed receivers usually must be manual, leading to long delays in data retrieval and processing, high operational costs associated with the required manpower, and safety risks for the operators. In addition, there is no efficient way to continuously assess the status of the acoustic telemetry system, including the acoustic transmitters and receivers. Here, we describe the design, implementation, and field validation of a cloud-based, real-time, underwater acoustic telemetry system with edge computing for estimating fish behavior and monitoring environmental parameters. The system incorporates microcontrollers for edge computing and connects to a cloud-based service that further post-processes the transmitted data stream to derive behavior and survival information of tagged animals. The developed system has been demonstrated to have significantly improved performance over the benchmark system because of the integration of edge computing, with a greatly reduced energy consumption of 0.014 W resulting in the energy used by the acoustic modem being reduced by over 300 times. This work opens up new design opportunities for future real-time and multifunctional underwater acoustic systems.

Index Terms—Acoustic telemetry, edge computing, environmental sensing, Internet of Things (IoT), real-time system.

Manuscript received 4 June 2021; revised 12 December 2021 and 19 February 2022; accepted 17 March 2022. Date of publication 8 April 2022; date of current version 7 September 2022. This work was supported by the U.S. Department of Energy (DOE) Office of Technology Transitions and Water Power Technologies Office and the Laboratory Directed Research and Development Program at Pacific Northwest National Laboratory (PNNL). PNNL is a multiprogram national laboratory operated by Battelle for DOE under Contract DE-AC05-76RL01830. (Corresponding authors: Jayson J. Martinez; Zhiqun Daniel Deng.)

Yang Yang was with the Energy and Environment Directorate, Pacific Northwest National Laboratory, Richland, WA 99354 USA. He is now with the Institute of Deep-Sea Science and Engineering, Chinese Academy of Sciences, Sanya, Hainan 572000, China (e-mail: yang.yang0513@gmail.com).

Robbert Elsinghorst, Jayson J. Martinez, Hongfei Hou, and Jun Lu are with the Energy and Environment Directorate, Pacific Northwest National Laboratory, Richland, WA 99354 USA (e-mail: robbert.elsinghorst@pnnl.gov; jayson.martinez@pnnl.gov; hongfei.hou@pnnl.gov; jun.lu@pnnl.gov).

Zhiqun Daniel Deng is with the Energy and Environment Directorate, Pacific Northwest National Laboratory, Richland, WA 99354 USA, and also with the Department of Mechanical Engineering, Virginia Tech, Blacksburg, VA 24061 USA (e-mail: zhiqun.deng@pnnl.gov).

Digital Object Identifier 10.1109/JIOT.2022.3164092

I. INTRODUCTION

UNDERWATER acoustic telemetry finds diverse applications in fields of subsea resource extraction, such as resource exploration [1], tracking the movements of aquatic and marine animals [2], [3], wireless communications for autonomous underwater vehicles (AUVs) [4], and the emerging Internet of Underwater Things (IoUT) [5]–[8]. However, unlike terrestrial wireless communication systems, state-of-the-art acoustic telemetry systems suffer from limited data rates and energy efficiency, because of the characteristics of underwater communication, including bandwidth-limited underwater acoustic channels—generally recognized as the most challenging communication media in use today [9]—caused by high path loss [8], time-varying multipath propagation [10], large and variable propagation delay, node mobility caused by water currents [1], and Doppler spread associated with the complex underwater environment [11]. In addition to the fundamental constraint imposed by the characteristics of underwater sound propagation, there are system constraints that affect the operation of underwater acoustic communications. For instance, acoustic transducers have their own bandwidth limitations. For instance, the upper bit rate for the ATM-903 series (Teledyne Technologies, USA)—a widely used underwater acoustic modem—is 15 360 bits/s [12], far lower than the bit rate of terrestrial wireless communications. For many applications, the mode of operation associated with this bit rate is not suitable as it sacrifices most of the ability for the modem to detect and correct for communication errors and, as a result, significantly lower bit rates must often be used.

As a direct consequence of those limitations, end users cannot communicate interactively with the data source to manage the large streams of complex data; this inevitably leads to high latency and the high operational cost associated with the required manpower, as well as safety risk, for the operators manually recovering data. Furthermore, the harsh environment makes it more difficult to recharge or replace battery-powered acoustic telemetry receivers. Therefore, high energy efficiency is desirable for underwater acoustic telemetry systems [13]. For instance, one of the most important applications of acoustic telemetry in the Pacific Northwest region of the USA has been evaluating the behavior and survival of juvenile salmonids migrating through hydropower

dams in the Federal Columbia River Power System using the Juvenile Salmon Acoustic Telemetry System (JSATS) [14], [15]. The current implementation of the JSATS requires manual recovery of the data stored on secure digital (SD) cards from each of the deployed autonomous acoustic receivers at remote locations. If multiple JSATS autonomous receivers in an array fail, the detection efficiency of tagged animals could be significantly reduced [16]. In addition, once the autonomous receivers are deployed, there is no way to identify whether they are continuing to operate correctly.

The past decade has seen a tremendous research effort to improve underwater acoustic communication to achieve higher data rates and higher energy efficiency. One straightforward approach for higher data rates is to increase the signal bandwidth (or transmission/signaling rate) [17]. But an increase in signaling rate leads to larger intersymbol interference that degrades the quality of the received signal and requires compensation such as channel equalization. Instead, an efficient use of underwater acoustic communication bandwidth has been recognized as the pathway to improving its performance [17]. A software-defined underwater acoustic modem was proposed that has real-time adaptation capabilities to reconfigure its physical layer in real time under varying environmental conditions [18]. A bidirectional decision feedback equalizer was proposed, which exhibits improved signal-to-noise ratio (SNR), especially for time-varying channels [19]. Improved power efficiency was demonstrated using the orthogonal signal-division multiplexing technique over multiple transducers [20]. Novel data transfer schemes and networking protocols were proposed for higher energy efficiency and channel utilization [21]. Multiple-input-multiple-output techniques were explored to increase the data rate over the bandwidth-limited channels [22]. However, the highest data rates that are offered by commercially available acoustic modems could achieve up to a few kilobits per second over distances up to a few kilometers, far below their terrestrial counterpart [5]. Furthermore, the proliferation of IoUT equipment to enable a new generation of underwater monitoring applications, with increasingly complex device structures and large amounts of data, has further intensified the computing resource demands and pushed the horizon of a new computing paradigm.

Edge computing is an emerging networking philosophy in which computational loading is shifted from the centralized cloud or end users to edge sensors where the data are initially collected [23]. Research and practice on this emerging technology have led to a wide spectrum of applications, including live video analytics, agriculture, and smart home and industrial Internet of Things (IoT) devices [24]. Recently, Yang *et al.* [3] have developed the Lab-on-a-Fish: a miniaturized biotelemetry tag that combines edge computing for studying the aquatic animals with improved efficacy. Edge computing offers a new strategy for shorter latency and effective bandwidth usage for underwater acoustic communication [25].

In this work, we developed a cloud-based, real-time underwater acoustic telemetry receiver with edge computing and connectivity to end data users. Edge computing enables the system to provide more effective data analytics with lower

latency. The system incorporates microcontrollers and integrated circuits to create real-time data communication and processing units for edge computing; this is connected to a cloud-based service that further post-processes the transmitted data stream to derive the behavior and survival of tagged aquatic animals. The developed system has been successfully validated in a field environment, with a peer-to-peer acoustic communication range of up to 3.5 km and data communication to the cloud service in near real time. This work opens new design opportunities for future real-time and multifunctional underwater acoustic systems that can be used to estimate the behavior or survival of the tagged animals in near real time.

In summary, our major contributions include the following.

- 1) We identified the challenges in the existing acoustic telemetry solutions, namely, the contradiction between large volumes of raw data to be transmitted and the limited bandwidth of underwater acoustic communication.
- 2) We proposed the integration of edge computing to compress the large volume of raw data to a significantly reduced amount, which dramatically reduces the energy burden and increases the system's lifetime, enabling real-time communications from remotely located acoustic receivers.
- 3) We designed and implemented the proposed system and conducted laboratory and field experiments to demonstrate its feasibility and efficiency.

II. DESIGN AND IMPLEMENTATION

A. System Overview

We developed a cloud-based underwater acoustic telemetry receiver employing edge computing with increased energy efficiency and direct connectivity to end data users to enable real-time fish behavior and environmental monitoring. Fig. 1 is a system-level schematic diagram of the developed system, which consists of the offshore and nearshore/onshore (hereafter referred to as onshore) components.

The offshore components are responsible for receiving the acoustic signal from nearby fish-borne acoustic transmitters [26], performing edge computing, taking environmental measurements using off-the-shelf sensors, and transmitting the data to the onshore components. We designed a miniaturized, fully integrated printed circuit board (PCB). This PCB and the acoustic modem are incorporated into the existing housing of the offshore acoustic receiver. The PCB runs a custom-designed, real-time operating system that acquires the raw data from the acoustic receiver, removes common false positives using multipath and pulse repetition interval (PRI) filtering algorithms [27], obtains environmental sensor measurements [28], and compresses the resulting data into a transmittable packet.

The master onshore components handle five tasks.

- 1) Request and receive data from each acoustically linked offshore component.
- 2) Configure modem settings, such as transmission speed and power level.
- 3) Perform environmental measurements (temperature, barometric pressure, humidity, and wind direction).

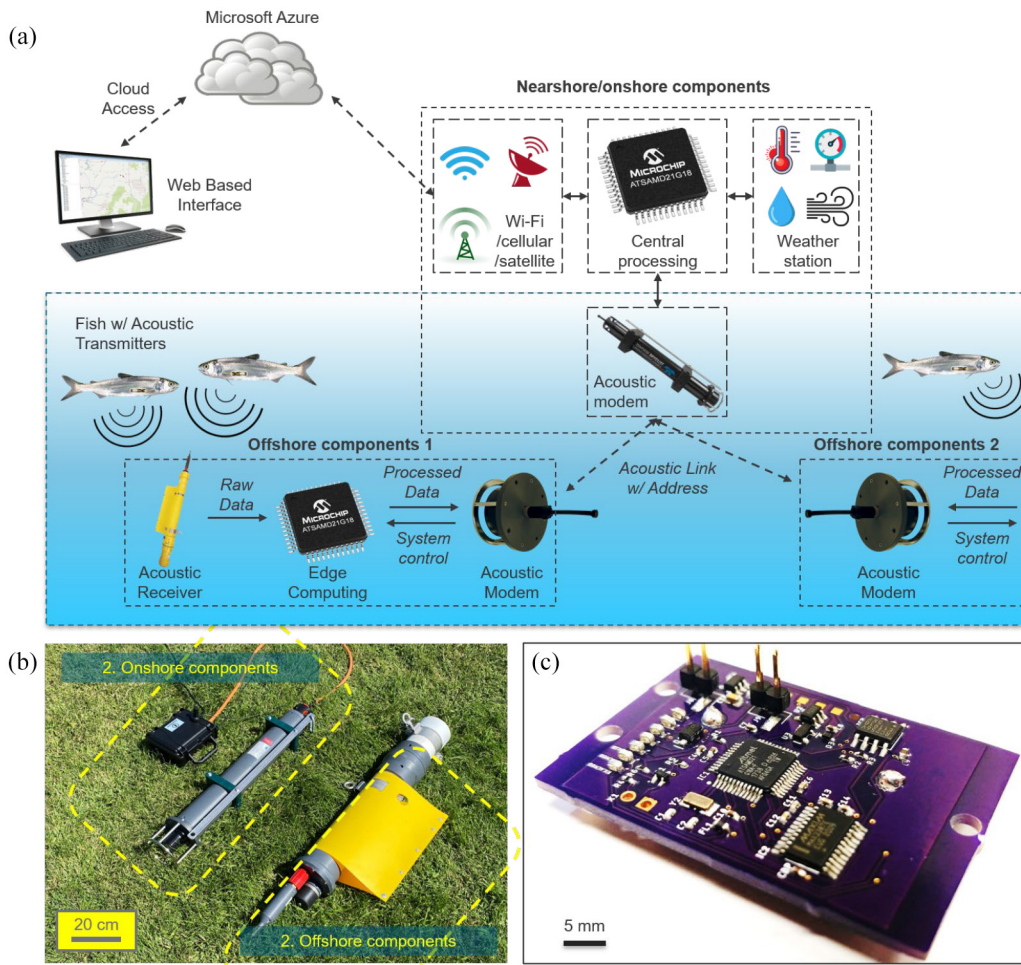


Fig. 1. Real-time underwater acoustic telemetry receiver system with edge computing. (a) Overview of the developed underwater acoustic telemetry system. (b) Offshore and nearshore/onshore prototype. (c) Offshore PCB for edge computing (dimensions: 5 cm × 3.5 cm).

- 4) Notify the slave onshore components to acquire the data from the offshore components they are linked to.
- 5) Send the data to the cloud service (Microsoft Azure Cloud Service) through either cellular or satellite communication, depending on service availability.

The cloud-based system process visualizes the collected real-time data and provides options to view historical data. The embedded cloud services transmit the real-time data from receivers into the database and propagate it to the frontend.

We developed the front end of the cloud-based system using Angular 4, a framework from the Google Angular team and other communities, and the back end using JAVA EE. We chose GlassFish as the application server, and the database we used was PostgreSQL, an open-source relational database. To make the system platform independent, we created RESTful (representational state transfer) Web services for the communications and data transfers between the front-end user interfaces and the back-end database. For each receiver, we created a WebSocket to handle the real-time calculation of collected data for fish behavior. Google Maps was used to display the locations of the deployed systems.

B. System Implementation

The offshore hardware (Fig. 6(a), see the Appendix) consists of: 1) a custom-designed PCB featuring a low-power, 32-bit microcontroller unit (MCU) as the core of the system; 2) an autonomous acoustic receiver; 3) a water leak sensor; and 4) an underwater acoustic telemetry modem for long-distance (up to 3 km in an ideal scenario) wireless communication in an aquatic environment. The autonomous acoustic receiver is equipped with a hydrophone that receives the high-frequency mechanical vibrations from tagged fish and converts them to electrical signals. The integrated digital signal processor (DSP) digitizes the input signal for use by the DSP in its detection and decoding algorithm. The detection algorithm searches for a tag signal, and the decoding algorithm determines what specific tag code is present. When a tag code is validated by the DSP, it passes the detection information on for storage in memory. The acoustic receiver is also equipped with built-in sensors for temperature, tilt angle, pressure (available but not included for this prototype), and battery voltage. The decoded tag detections and the data from the built-in sensors are transferred to the MCU in real time through an RS-232 serial interface. While developing this module, we also integrated a leak

sensor, which sends a warning message to the cloud service, via the onshore system, when water is detected inside the acoustic receiver package. Since this module operates independently of the acoustic receiver, custom sensors can be readily integrated into the system without requiring modification of the hardware or software of the acoustic receiver.

The firmware was implemented as an interrupt-driven system for minimizing the power consumption of the offshore components. Fig. 7(a) (see the Appendix) shows a simplified flowchart of the firmware implementation. If no data are coming from the acoustic receiver at the input serial port of the microcontroller board, the board remains in the sleep mode, maintaining minimal functionalities to conserve power. When real-time data are received at the input serial port, the board wakes up, and the onboard timer is reset. The timer will accumulate if no data are seen at the input, and the microcontroller board will be put back to sleep after a programmed threshold. The firmware checks the incoming data and only starts to process the data if the input format fulfills the criteria. After all the data are processed, the microcontroller transmits the edge computed data to the offshore acoustic modem and resumes the sleep mode. The detailed description of the data processing scheme is as follows. First, the received data are parsed to extract the detected tag code (along with its detection time) and sensor data, including environmental temperature and the battery level of the acoustic receiver. Then, the extracted tag code is checked against a dynamic table of tag codes to search for the table index. If the extracted tag code does not exist in the table and the table is not full, the tag code will be filled into an empty slot of the table and its detection time is stored. Next, the detection time of the extracted tag code in the dynamic array is passed to the multipath filter. If it passes the multipath filter, it will be appended to an array and checked by the PRI filter. Otherwise, this data will be thrown away. For each tag code, it will skip the PRI filter if it has passed the PRI filter during the current hour since only hourly data for each tag code are sent. Further explanation and implementation details of the multipath and PRI filters will be discussed later. Finally, the firmware cleans up the table each hour by removing tag codes that have not been detected over the past hour and their corresponding detection times. The dynamic table ensures the firmware works with a large number of tag codes, with the only limitation being the maximum number of tag codes detected in a single hour. This maximum number of tag codes is determined by the size of the MCU's static random-access memory and can be improved by implementing external memory such as an extended ferroelectric random-access memory.

Fig. 6(b) (see the Appendix) shows the hardware architecture for the onshore components. The core of the onshore components is a low-power microcontroller board running a real-time operating system. The microcontroller board is connected through serial ports to several components: 1) an underwater acoustic telemetry modem; 2) a cellular modem and an Iridium satellite modem for uploading the data to the cloud-based service; and 3) a weather station. The weather station integrates humidity and temperature, barometric pressure, and wind speed/direction sensors. The cellular modem can be replaced with a Wi-Fi modem or Ethernet modem

TABLE I
LIST OF COMPONENTS USED FOR THE HARDWARE IMPLEMENTATION

Description	Part Number	Manufacturer
MCU	ATSAMD21G18	Microchip, USA
autonomous acoustic receiver	SR3001	ATS, USA
offshore underwater acoustic telemetry modem	ATM-903	Teledyne Technologies, USA
microcontroller board	Arduino Zero	Arduino, Italy
onshore underwater acoustic telemetry modem	ATM-915	Teledyne Technologies, USA
cellular modem	Particle Boron LTE Cat M1 + Bluetooth	Particle, USA
Iridium satellite modem	RockBLOCK Mk2	Rock Seven, UK
weather station	Photon Weather Shield	SparkFun, USA
humidity and temperature sensor	Si7021	Silicon Labs, USA
barometric pressure sensor	MPL3115A2	NXP, the Netherlands
wind speed/direction sensor	Weather Meter Kit	SparkFun, USA
Wi-Fi modem	Particle Argon Wi-Fi + Bluetooth	Particle, USA
Ethernet modem	Particle Ethernet FeatherWing	Particle, USA

for less-remote locations where these may be available. The cellular, Wi-Fi/Ethernet, or satellite modems pass the data packet through an intermediate service to our cloud-based service. The acoustic modem operates on a band between 22 and 27 kHz and has a source level of 178 dB. The modem is configured to run at a baud rate of 300 bits/s for enhanced reliability in a low signal-to-noise environment. Details of the components used for both onshore and offshore hardware implementation are shown in Table I.

A finite-state-machine model was adopted for the firmware implementation of the onshore components, which is responsible for performing all control, monitoring, and wireless communication functions. Fig. 7(b) (see the Appendix) shows the simplified flowchart of the firmware implementation. First, the board is initialized with hardware configurations, including serial communication and an onboard real-time clock (RTC). The onboard RTC tracks the current time and wakes the system at the preprogrammed time (usually on an hourly basis). Then, the firmware checks whether the cellular module is awake or not. If not, the software tries to wake it up. Afterward, the board sends a command to the onshore acoustic modem to request data from each of the offshore acoustic modems. Data are requested from each offshore component one at a time, by sending the request to the modem address associated with the specific offshore component, and the onshore component will listen for a predefined duration for the incoming data. If no data are received within that preprogrammed time frame, the board will request data to the acoustic modem again. If no data could be received after a maximum number of retries, the board will stop sending data requests to the modem and proceed to the next step. After

cycling through all the acoustically connected offshore components, the master onshore components notify the slave onshore components on the other side of the riverbank to acquire the data from the offshore components associated with it. This sequential data acquisition process ensures that no conflict would happen (i.e., the master and slave onshore components will not try to communicate with offshore components simultaneously). After collecting the environmental data from the weather station, the board attempts to send the received data to the cloud if valid data have been received or alternately send a warning message if a maximum number of request retries has been reached. The firmware will primarily try to send the data to the cloud through the cellular network. If the cellular network is not available, the data transmission will return failure and trigger satellite communication. This dual-communication mechanism ensures that our system is accessible in even the most remote locations. In most cases, cellular communication will be preferred because it is more cost effective. In places without cellular network coverage, or when the cellular modem fails, satellite communication can be used. Eventually, the board enters the sleep mode to conserve power. The program loops back to the beginning of the finite state machine and the same sequence is repeated at a preprogrammed time.

C. Edge Computing

The complex aquatic environment hampers underwater acoustic communication in several ways, including high path loss, time-varying multipath propagation, large and variable propagation delay, and Doppler spread. Given the low baud rate of acoustic modems (maximum speed of 15 360 bits/s with minimal error detection/correction), it was not feasible to transmit all the raw detection information, even after filtering out false detections. To address these issues, we implemented edge computing at the offshore acoustic receiver, including a sequence of filters to remove false-positive decodes that do not match the expected transmission interval (i.e., PRI filter) and further compression, as shown in Algorithm 1.

A schematic illustration of the filtering steps and their criteria is shown in Fig. 2(a). The original data set received by the acoustic receiver consists of a series of decodes at different times. Decodes are first processed using the multipath filter to remove decodes that originate from reflections of the acoustic transmissions (off the surface, bottom, or any other structures) or from transmission refraction in the water. The real-time implementation of the multipath filter is illustrated in the simplified flowchart shown in Fig. 2(b). Each time a tag message is received, it will be checked against the dynamic lookup table. The program then checks the number of messages for each valid tag code that was received. If the number of messages is equal to zero (either there was no message previously received, or the memory was cleared), the detection time of this message is kept as the time of arrival (TOA) of this tag code. Then, the time difference between this message and the most recently received message of the same tag code is calculated, and the time difference is compared with the multipath filter time window. In the current implementation,

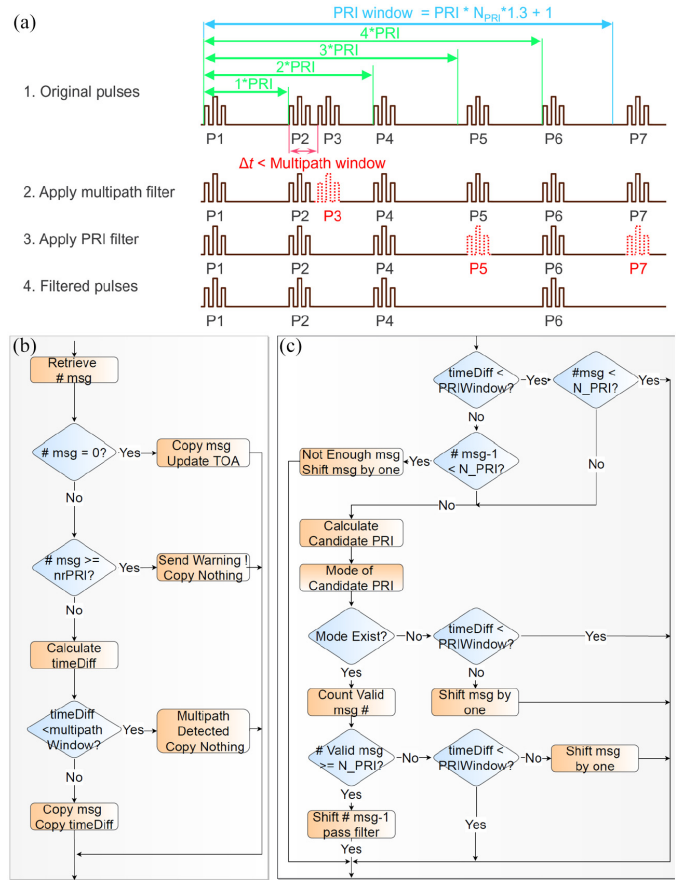


Fig. 2. (a) Schematic of filtering steps implemented for real-time data processing. (b) and (c) Simplified flowcharts showing implementation of the multipath and PRI filters.

a value of 0.5 s (while the tags have a nominal PRI of 3 s) was used as the multipath filter time window. If the difference is smaller than the multipath filter time window, a multipath event is confirmed, and this message is ignored. Otherwise, we increment the number of detected messages for this tag code, and we save this detection time in an array.

We implemented a real-time PRI filter [Fig. 2(c)] to filter out false-positive decodes for each tag code using the expected pattern of transmissions. The time difference of each message between a set of messages and the TOA of the initial message is calculated. This time difference is then sorted into one of three pairings of the number of messages (N_m) and the PRI window. The PRI window is based on a multiple of the tag's nominal PRI value and is defined as the nominal $\text{PRI} \times 1.3 \times 12 + 1$ s [29]. If the time difference is smaller than the PRI window, and if N_m is larger than the minimum number required to pass the PRI filter (N_{PRI}), i.e., $N_m \geq N_{\text{PRI}}$, we proceed to the next step. If the time difference is larger than the PRI window, and if $N_m - 1 \geq N_{\text{PRI}}$ (the last message is dropped out since it is outside the PRI window), we proceed to the next step. If $N_m - 1 < N_{\text{PRI}}$, then the initial message is invalid and discarded and the filter moves on to the next message. After the PRI window condition check, the time difference between the initial message and each subsequent message is calculated, and an estimated PRI for each

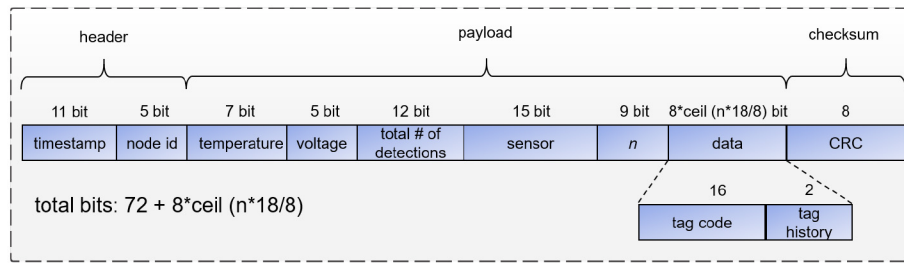


Fig. 3. Implemented data packet structure.

Algorithm 1 Edge Computing

Input: message array

Output: compressed data

```

1: while data processing is not done do
2:   decode message
3:   if tag code is valid then
4:     totalDetections++
5:   end if
6:   process sensor data
7:   extract the tagCode and detection time
8:   if time difference < multiPathWindow then
9:     calculate candidate PRI for each message
10:    truePRI = mode(candidatePRIArray)
11:    if truePRI != 0 then
12:      count number of valid messages
13:      if # of valid messages ≥ numPRI-1 then
14:        passedPRIFilter = true
15:      end if
16:    end if
17:  end if
18: end if
19: end while
20: compress data
21: prepare header, payload, and CRC code
22: send compressed data

```

message is calculated. The mode of the estimated PRIs is computed to obtain the real PRI of these N_m messages. Each estimated PRI is further compared with the real PRI, and only messages with a difference below a threshold (0.006 s in the current implementation) are considered valid. When at least N_{PRI} valid messages from a set of messages match the PRI criteria within a predefined PRI time window, this set of messages passes the PRI filter. Otherwise, the initial message is discarded and the filter moves on to the next message. Note that the required N_{PRI} (typically 4, 5, 6, or 7) depends on the acoustic environment. After the PRI filter is applied, and continuous messages are formed into a detection event. Each detection event contains at least N_{PRI} messages and represents a continuous or nearly continuous sequence of transmissions.

D. Data Compression

We further reduce the data to a minimal data packet (Fig. 3), which is essential to accommodate the limited baud rate and to

prolong the lifetime of a battery-limited underwater acoustic telemetry system [30]. Each data packet structure contains the header information, the payload, and the cyclic redundancy check (CRC). The data packet structure contains the following parts.

- 1) *Timestamp*: 11 bits.
- 2) *Node ID*: 5-bit receiver ID.
- 3) *Temperature*: 7-bit data within a range of 0 °C–25.4 °C to cover the temperature variation in the river over the year.
- 4) *Voltage*: 5-bit receiver battery voltage in a range of 1.9–3.5 V.
- 5) *Total Number of Detections*: 12-bit total number of raw detections (divided by 1000) that have been detected since the node was deployed, which serves as a replacement for the capacity used on the memory card.
- 6) *Sensor*: 16 bits, reserved for the potential environmental sensor.
- 7) *n*: 9 bits to represent the number of the tag code to transmit at this hour.
- 8) *Data*: $8 \times \text{ceil}(n \times 18/8)$ -bit data, which covers all the detected tag codes over the past two hours and their 2-bit tag history. n stands for the total number of tag codes. The first bit of the tag history indicates whether the tag was detected in the earlier of the past two hours, and the second bit indicates whether the tag was detected during the most recent hour.
- 9) *CRC*: 8-bit CRC code to handle error checking. An 8-bit CRC computation was implemented in the hardware (CRC-8-Dallas/Maxim, polynomial representation: $x^8 + x^5 + x^4 + 1$).

This implementation of a data packet does not require pre-knowledge of what tag codes are expected, which provides flexibility to handle a larger number of potential tag codes.

III. SYSTEM VALIDATION

A. Laboratory Validation

The developed cloud-based underwater acoustic telemetry system was validated in a laboratory environment. Fig. 4 shows the results of tag detection, environmental sensing, and the remote receiver's health monitoring using the underwater acoustic telemetry system. The lab testing was conducted in an acoustic water tank filled with freshwater [29]. A receiving hydrophone from the acoustic receiver (SR3017, ATS, USA; electronics nearly identical electronics to ATS SR3001) was submerged in the tank 0.43-m deep and 1 m away from the

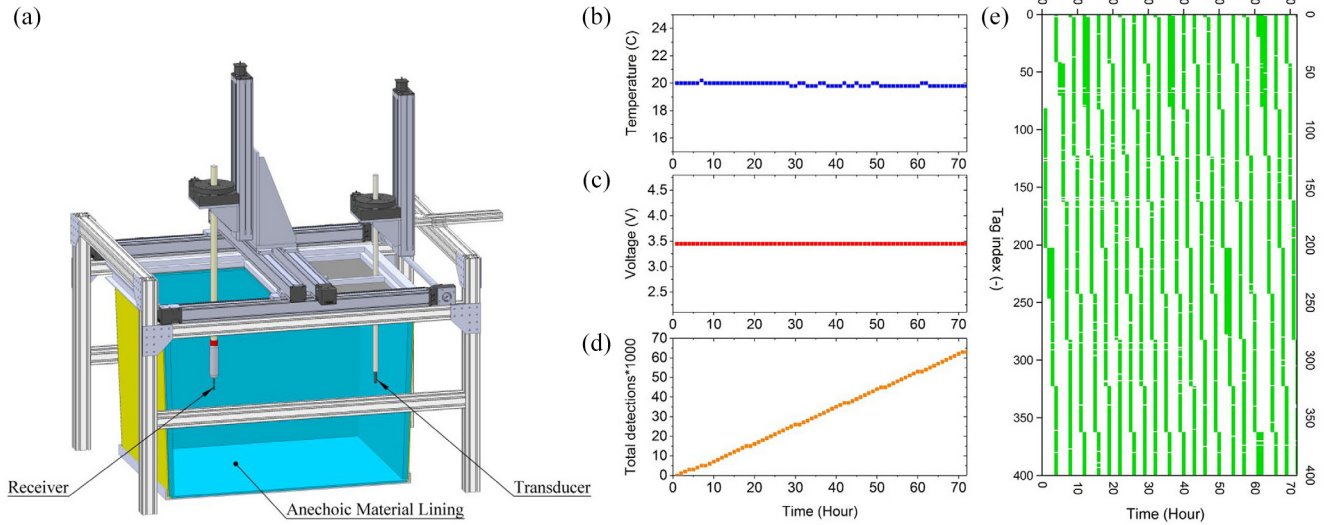


Fig. 4. Laboratory validation for tag detection, environmental sensing, and remote receiver health monitoring using the underwater acoustic telemetry system. (a) Experimental setup of the laboratory validation test. (b) Temperature of the environment. (c) Voltage level of the acoustic receiver. (d) Total number of tag detections. (e) Visualization of tag detections over a period of 72 h.

transducer. The transducer was controlled by a workstation (Precision T7500 workstation, Dell, USA) via a MATLAB-based interface to emit acoustic signals containing different JSATS transmitter tag codes [31]. The tag codes were sent with a 3-s PRI from the MATLAB interface via a data acquisition card (PCI-6111, National Instruments, USA). A total of 400 tag codes were transmitted, and each tag code was transmitted eight times in a row before the system switched to the next tag code. The offshore microcontroller board was connected to the acoustic receiver to obtain the raw detection data. The microcontroller processes this data, forming an hourly summary of tag, receiver, and sensor data and then transmits this summarized data to the onshore acoustic modem serially connected on the other end.

Fig. 4(b) shows the measured temperature of the water tank during 72 h. A stable temperature of around 20 °C was obtained since the lab is a temperature-controlled environment. The unwavering battery voltage of around 3.5 V, as shown in Fig. 4(c), was a result of the acoustic receiver being connected to a lab power supply. About 63 000 detections were captured for the 400 valid tag codes, which complies with the input MATLAB program. Using the developed system, we visualized the detected tag codes in real time. These are shown in Fig. 4(e), where a green box stands for a tag code that was detected and passed the multipath and PRI filters.

B. Field Validation

The developed system was subsequently validated in a field environment. The first field trial took place on August 26, 2020, in Levey Park, WA, USA [Fig. 5(a)]. This field environment was chosen as it is close to the Ice Harbor Hydroelectric Dam. The Ice Harbor Dam is located on the Snake River, approximately 14 km from the mouth of the Columbia River, where there is particular interest in understanding the impacts dams have on the survival and behavior of migrating juvenile

salmonids [32] protected by the Endangered Species Act of 1973.

Both the offshore and onshore components were deployed off the docks at a depth of 1.5 m and approximately 50 m apart. A custom acoustic transmitter, which transmits a list of tag codes to the offshore component at a PRI of 3 s, was deployed off the same dock as the onshore component. The total duration of the data collection and edge computing was about 2 h. The tag code list and sensing data were successfully communicated to the Azure server [Fig. 5(b)], showing that the acoustic telemetry system functioned as expected. Thanks to the developed onboard edge computing algorithms, a total of 458 000 bytes of raw data (which contained 36 valid tag codes after the PRI filter) was compressed to only 200 bytes. To estimate the deployment time of the acoustic modem, the following equations are used:

$$T = \frac{C}{24 * P_{\text{hourly}}} \quad (1)$$

$$P_{\text{hourly}} = \left[\frac{D * P_{\text{transmit}}}{450 * R} + T_{\text{overhead}} * P_{\text{non-transmit}} + \left(1 - \frac{D}{450 * R} - \frac{T_{\text{overhead}}}{3600} \right) * P_{\text{sleep}} \right] / 1 \text{ h} \quad (2)$$

where the constant 24 is derived from 24 h/day, converting hours to days, and the constant 450 is derived from 8 bits/byte and 3600 s/h, converting bits/sec to bytes/hour. Detailed information of the parameters used for the calculations is shown in Table II.

Because of the integration of the edge intelligence, our developed system shows a significant performance improvement over the benchmark system (ATM-903 series, Teledyne Technologies, USA), with a dramatically reduced energy consumption of 0.014 W (in comparison to 4.2478 W), and an estimated lifetime increase by over 300 times (892 days in

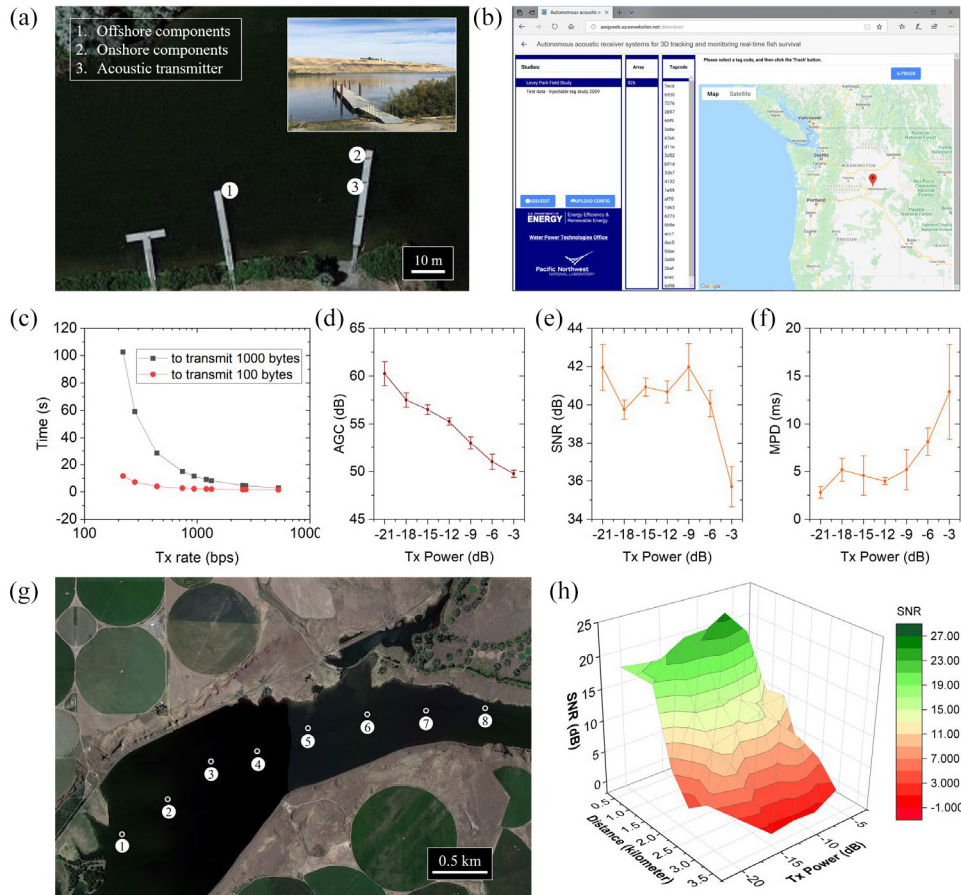


Fig. 5. Field validation of the developed underwater acoustic telemetry system. (a) Experimental setup of the first field trial. The inset photograph shows the river and dock. (b) Cloud graphical user interface showing the detection metrics from the validation test. (c)–(f) Acoustic link-quality parameters of the developed system in the field environment at a distance of 50 m. (g) Experimental setup for the second field trial. (h) Surface plot showing the SNR as a function of Tx Power and receiver–transmitter distance.

TABLE II
PARAMETERS USED TO ESTIMATE THE DEPLOYMENT
TIME OF THE ACOUSTIC MODEM

Symbol	Unit	Value	Definition
T	days	-	the lifetime of the modem
C	watt-hour	300	battery capacity
P_{hourly}	watts	-	hourly energy consumption
D	bytes	-	the amount of data to transmit
D_{raw}	bytes	458,000	raw data
$D_{\text{compressed}}$	bytes	200	compressed data
R	bit/s	2400	transmission bit rate
P_{transmit}	watts	10	power consumption during transmission
$P_{\text{non-transmit}}$	watts	0.6	power consumption when not transmitting
P_{sleep}	watts	0.012	power consumption during system sleep
T_{overhead}	seconds	1.2	overhead time for system wakeup

TABLE III
ESTIMATED PERFORMANCE COMPARISON BETWEEN
THE BENCHMARK AND PROPOSED SYSTEM

	Energy consumption	Lifetime
Benchmark system	4.2478 watt-hour	2.94 day
Developed system	0.0140 watt-hour	892 day

comparison to 2.94 days), as detailed in Table III. Here, the benchmark system refers to what we would expect from the acoustic modem being connected directly to the full raw output of the acoustic receiver, i.e., no edge computing to reduce the amount of data to be transmitted, but the modem operating in an identical configuration.

To confirm the correctness and accuracy of the edge computing, we ran the MATLAB script from the previously designed offline data processing algorithm on the data recovered from the acoustic receiver and compared the result with the results from edge computing. The edge computing showed a perfect agreement with the offline data processing algorithm.

The quality of the acoustic link between the onshore and offshore components was field tested by instructing the onshore component to transmit one preset test message at a specified transmission rate (Tx rate) and power. The resultant transmission time as a function of the Tx rate, automatic gain control (AGC), SNR, and multipath delay (MPD) as a function of transmission power (Tx Power) are shown in Fig. 5(d)–(f), respectively. By using a communication speed of 1000 bits per second (bps), a data packet of 1000 bytes (which

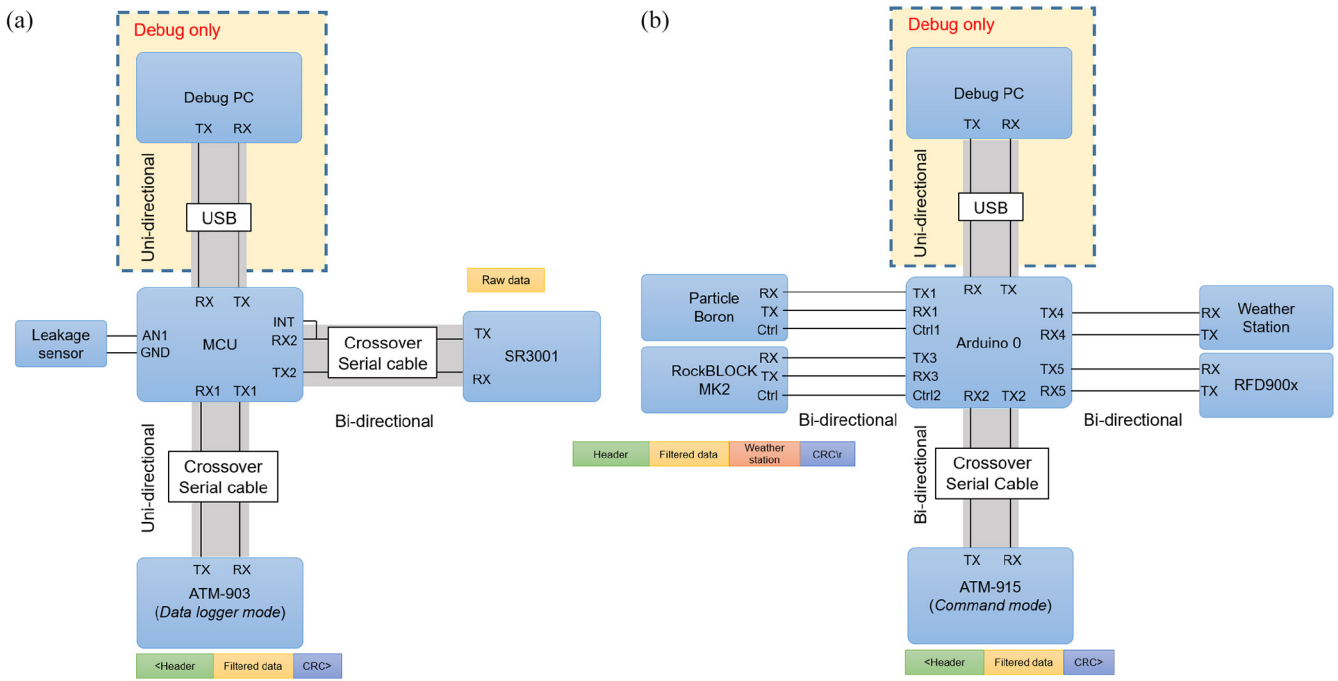


Fig. 6. System implementation. Hardware architecture for the (a) offshore and (b) onshore components.

corresponds to 180 detected tag codes) could be transmitted within 11 s. The MPD shows an increasing trend as the Tx Power increases. The received signals need less AGC as Tx Power is strengthened. Surprisingly, a higher Tx Power has caused a decreased SNR.

To investigate the maximum communication range of the developed system, we carried out a second field test on September 23, 2020, at the same location [Fig. 5(g)]. The offshore component was fixed at a single location throughout the experiment, while the onshore component was temporarily deployed from a boat at each test distance. The offshore component was anchored at the bottom of the river at a depth of about 25 m (location 1 in Fig. 4(g)). The onshore component was lowered into the river from the boat at a depth of about 5 m at locations 2–8 in Fig. 5(g). Eight communication ranges were tested: 1) 0.5 km; 2) 1 km; 3) 1.5 km; 4) 1.5 km; 5) 2 km; 6) 2.5 km; 7) 3 km; and 8) 3.5 km. The system functioned successfully at all eight tested locations, although the communication became observably unreliable at the 3.5 km distance. The SNR shows a clear dependence on the communication range, as displayed in Fig. 5(h). At the communication range of 3.5 km, an average SNR of around zero was obtained.

IV. CONCLUSION

In this study, a real-time acoustic telemetry receiver system with edge computing for studying fish behavior and environmental sensing was designed, implemented, and validated in a controlled field environment. Edge computing was implemented at the acoustic receiver to allow the data to be pre-processed and filtered closer to where it is created. The system integrated a flexible connection to the cloud via cellular, Wi-Fi, or satellite communication, to allow the most cost-effective solution that still allows the system to be deployed in remote

locations. These calculated metrics could be used to elucidate the overall status of fish migration during a given season and allow hydropower operators to observe the effects of operational changes, such as changes to spill patterns, in near real time. The system was implemented with the flexibility to allow additional sensors to be integrated into both the onshore and offshore components for environmental monitoring. The system has been tested and validated in both a laboratory and a controlled field setting. The maximum operational range of the system was confirmed to be 3.5 km in a field environment. In addition to allowing near-real-time detection information, this system will allow remote health monitoring of each of the receivers deployed. This will allow users of the system to detect issues that may significantly reduce the performance of the system. The system described in this contribution is expected to bridge diverse technologies for sensing the ocean and contributing to the development of IoUT and Smart Ocean. Possible future extensions and applications of the system include: 1) a system that combines the onshore and offshore systems for applications where we want the real-time data but can deploy from the surface and 2) a system that implements a customized version of the receiver firmware that outputs slightly less information but close to true real time for applications such as using the system on an autonomous robot.

APPENDIX

See Figs. 6 and 7.

ACKNOWLEDGMENT

The authors would like to thank Advanced Telemetry Systems (ATS) and their staff for their support in modifying the design of their SR3001 autonomous acoustic receiver to

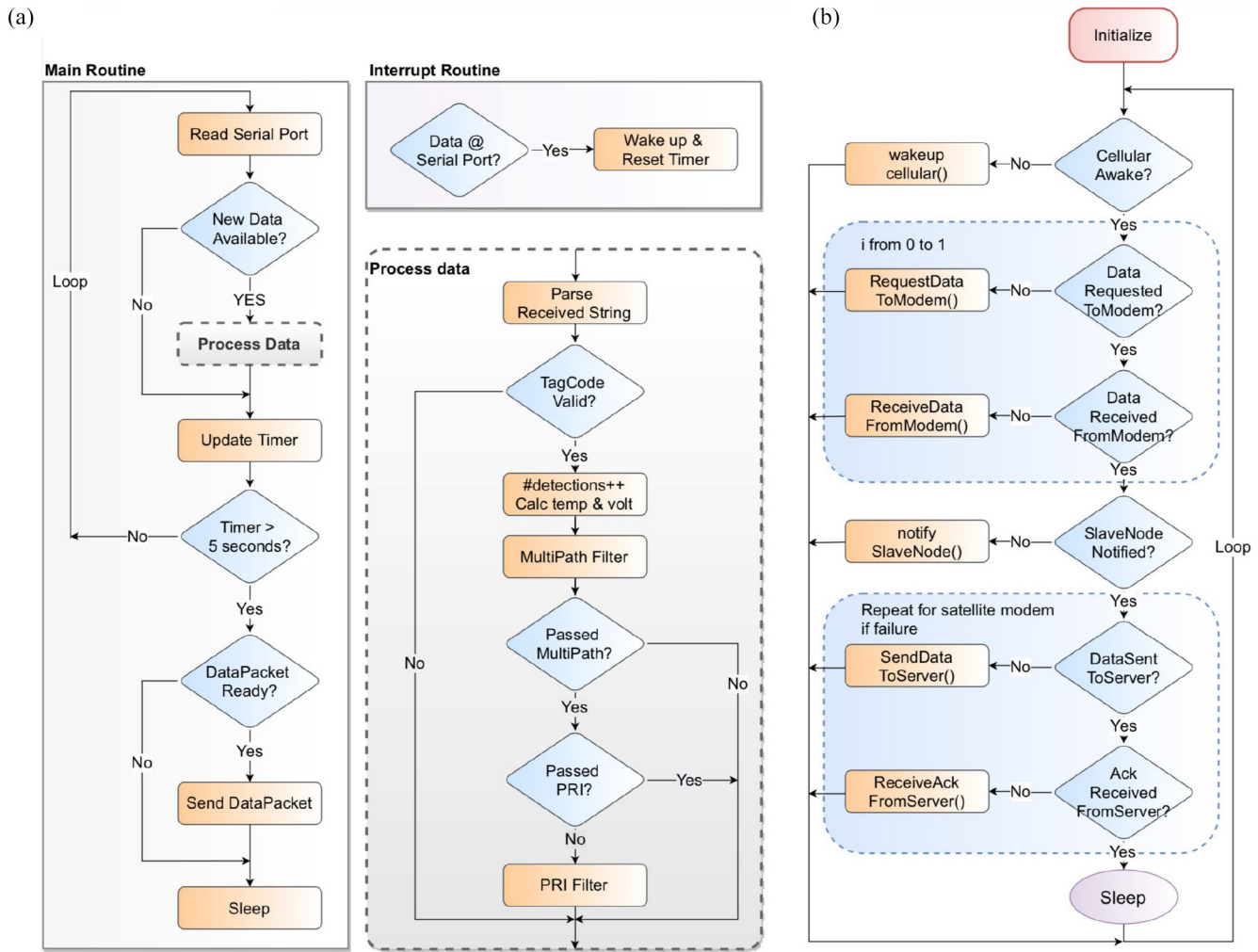


Fig. 7. System implementation. (a) Simplified flowchart of the firmware implementation of the MCU deployed off-the-shore and (b) onshore. “Ack” is “acknowledgment.”

integrate an acoustic modem and assistance in interfacing the custom electronics designed in this study with their hardware.

REFERENCES

- [1] G. Han, C. Zhang, L. Shu, and J. J. P. C. Rodrigues, “Impacts of deployment strategies on localization performance in underwater acoustic sensor networks,” *IEEE Trans. Ind. Electron.*, vol. 62, no. 3, pp. 1725–1733, Mar. 2015.
- [2] N. E. Hussey *et al.*, “Aquatic animal telemetry: A panoramic window into the underwater world,” *Science*, vol. 348, no. 6240, 2015, Art. no. 1255642.
- [3] Y. Yang *et al.*, “Lab-on-a-fish: Wireless, miniaturized, fully integrated, implantable biotelemetric tag for real-time in vivo monitoring of aquatic animals,” *IEEE Internet Things J.*, early access, Nov. 13, 2021, doi: [10.1109/JIOT.2021.3126614](https://doi.org/10.1109/JIOT.2021.3126614).
- [4] A. Song, M. Stojanovic, and M. Chitre, “Editorial underwater acoustic communications: Where we stand and what is next?” *IEEE J. Ocean. Eng.*, vol. 44, no. 1, pp. 1–6, Jan. 2019.
- [5] J. Heidemann, M. Stojanovic, and M. Zorzi, “Underwater sensor networks: Applications, advances and challenges,” *Philos. Trans. Roy. Soc. A, Math. Phys. Eng. Sci.*, vol. 370, no. 1958, pp. 158–175, 2012.
- [6] M. C. Domingo, “An overview of the Internet of Underwater Things,” *J. Netw. Comput. Appl.*, vol. 35, no. 6, pp. 1879–1890, 2012.
- [7] K. Casey, A. Lim, and G. Dozier, “A sensor network architecture for tsunami detection and response,” *Int. J. Distrib. Sens. Netw.*, vol. 4, no. 1, pp. 27–42, 2008.
- [8] T. Qiu, Z. Zhao, T. Zhang, C. Chen, and C. L. P. Chen, “Underwater Internet of Things in smart ocean: System architecture and open issues,” *IEEE Trans. Ind. Informat.*, vol. 16, no. 7, pp. 4297–4307, Jul. 2020.
- [9] M. Stojanovic and J. Preisig, “Underwater acoustic communication channels: Propagation models and statistical characterization,” *IEEE Commun. Mag.*, vol. 47, no. 1, pp. 84–89, Jan. 2009.
- [10] Y. Song, “Underwater acoustic sensor networks with cost efficiency for Internet of Underwater Things,” *IEEE Trans. Ind. Electron.*, vol. 68, no. 2, pp. 1707–1716, Feb. 2021.
- [11] J. Ling, H. He, J. Li, W. Roberts, and P. Stoica, “Covert underwater acoustic communications,” *J. Acoust. Soc. Amer.*, vol. 128, no. 5, pp. 2898–2909, 2010.
- [12] J. Cho, F. Ahmed, E. Shitiri, and H.-S. Cho, “Power control for MACA-based underwater MAC protocol: A Q-learning approach,” in *Proc. IEEE Region 10 Symp. (TENSYP)*, 2021, pp. 1–4.
- [13] Z. Zhou, B. Yao, R. Xing, L. Shu, and S. Bu, “E-CARP: An energy efficient routing protocol for UWSNs in the Internet of Underwater Things,” *IEEE Sensors J.*, vol. 16, no. 11, pp. 4072–4082, Jun. 2016.
- [14] G. A. McMichael *et al.*, “The juvenile salmon acoustic telemetry system: A new tool,” *Fisheries*, vol. 35, no. 1, pp. 9–22, 2010.
- [15] Z. D. Deng *et al.*, “An injectable acoustic transmitter for juvenile salmon,” *Sci. Rep.*, vol. 5, no. 1, pp. 1–6, 2015.
- [16] X. Li, Z. D. Deng, L. T. Rauchenstein, and T. J. Carlson, “Contributed review: Source-localization algorithms and applications using time of arrival and time difference of arrival measurements,” *Rev. Sci. Instrum.*, vol. 87, no. 4, 2016, Art. no. 41502.
- [17] H. C. Song and W. S. Hodgkiss, “Efficient use of bandwidth for underwater acoustic communication,” *J. Acoust. Soc. Amer.*, vol. 134, no. 2, pp. 905–908, 2013.

[18] E. Demirors, G. Sklivanitis, G. E. Santagati, T. Melodia, and S. N. Batalama, "A high-rate software-defined underwater acoustic modem with real-time adaptation capabilities," *IEEE Access*, vol. 6, pp. 18602–18615, 2018.

[19] H. C. Song, "Bidirectional equalization for underwater acoustic communication," *J. Acoust. Soc. Amer.*, vol. 131, no. 4, pp. EL342–EL347, 2012.

[20] T. Ebihara, "Improvement of power efficiency for underwater acoustic communication using orthogonal signal division multiplexing over multiple transducers," *Jpn. J. Appl. Phys.*, vol. 52, no. 7S, p. 7HG04, 2013.

[21] P. Xie, Z. Zhou, Z. Peng, J.-H. Cui, and Z. Shi, "SDRT: A reliable data transport protocol for underwater sensor networks," *Ad Hoc Netw.*, vol. 8, no. 7, pp. 708–722, 2010.

[22] B. Li *et al.*, "MIMO-OFDM for high-rate underwater acoustic communications," *IEEE J. Ocean. Eng.*, vol. 34, no. 4, pp. 634–644, Oct. 2009.

[23] G. Premsankar, M. Di Francesco, and T. Taleb, "Edge computing for the Internet of Things: A case study," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1275–1284, Apr. 2018.

[24] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge intelligence: Paving the last mile of artificial intelligence with edge computing," 2019, *arXiv:1905.10083*.

[25] L. Lin, X. Liao, H. Jin, and P. Li, "Computation offloading toward edge computing," *Proc. IEEE*, vol. 107, no. 8, pp. 1584–1607, Aug. 2019.

[26] H. Li *et al.*, "An energy harvesting underwater acoustic transmitter for aquatic animals," *Sci. Rep.*, vol. 6, Sep. 2016, Art. no. 33804.

[27] Z. D. Deng *et al.*, "Comparing the survival rate of juvenile Chinook salmon migrating through hydropower systems using injectable and surgical acoustic transmitters," *Sci. Rep.*, vol. 7, Feb. 2017, Art. no. 42999.

[28] Y. Yang and Z. D. Deng, "Stretchable sensors for environmental monitoring," *Appl. Phys. Rev.*, vol. 6, no. 1, 2019, Art. no. 11309.

[29] Z. Deng, M. Weiland, T. Carlson, and M. B. Eppard, "Design and instrumentation of a measurement and calibration system for an acoustic telemetry system," *Sensors*, vol. 10, no. 4, pp. 3090–3099, 2010.

[30] H. U. Yildiz, V. C. Gungor, and B. Tavli, "Packet size optimization for lifetime maximization in underwater acoustic sensor networks," *IEEE Trans. Ind. Informat.*, vol. 15, no. 2, pp. 719–729, Feb. 2019.

[31] H. Li, Z. D. Deng, Y. Yuan, and T. J. Carlson, "Design parameters of a miniaturized piezoelectric underwater acoustic transmitter," *Sensors*, vol. 12, no. 7, pp. 9098–9109, 2012.

[32] J. J. Martinez *et al.*, "Hydraulic and biological characterization of a large Kaplan turbine," *Renew. Energy*, vol. 131, pp. 240–249, Feb. 2019.



Jayson J. Martinez received the bachelor's and master's degrees in mechanical engineering from Portland State University, Portland, OR, USA, in 2008 and 2009, respectively.

He has been a Research Engineer with the Energy and Environment Directorate, Pacific Northwest National Laboratory (PNNL), Richland, WA, USA, since December 2009. He has contributed to the research and design of different systems that address engineering and ecological concerns related to hydro, wind, marine, and hydrokinetic energy systems. In addition to research and design, he has been involved extensively with the field deployment of these technologies as well as the data reduction and analysis of the data gathered in the field. He is the Technical Manager of the Bio-Acoustics and Flow Laboratory, PNNL, which is accredited by The American Association for Laboratory Accreditation. He has coauthored 30 published journal publications and 38 technical reports.



Hongfei Hou received the Ph.D. degree in computer science from Washington State University, Pullman, WA, USA, in 2018.

He is a Senior Software Engineer with Pacific Northwest National Laboratory, Richland, WA, USA, and has over 20 years of professional experience that includes software design and software development. His research focuses on designing and implementing cloud-based applications and how to use data mining to monitor the behaviors of these applications.



Yang Yang (Member, IEEE) received the Ph.D. degree in electrical engineering from imec and Ghent University, Ghent, Belgium, in 2017.

From 2017 to 2018, he was a Postdoctoral Fellow with Japan Society for the Promotion of Science, Osaka University, Suita, Japan. From 2018 to 2020, he was a Postdoctoral Research Associate and then a Staff Member with Pacific Northwest National Laboratory, Richland, WA, USA. In 2020, he joined the Institute of Deep-Sea Science and Engineering, Chinese Academy of Sciences, Sanya, China, as an

Associate Professor. He has authored or coauthored over 30 peer-reviewed papers and conference proceedings. His current research interests include flexible and stretchable sensor systems and marine sensors and electronics.



Jun Lu (Member, IEEE) received the Ph.D. degree from the Department of Electrical and Computer Engineering, State University of New York at Binghamton, Binghamton, NY, USA, in 2012.

He is currently a Senior Electrical Engineer with Pacific Northwest National Laboratory, Richland, WA, USA. His research interests include autonomous sensors for environmental assessment, and low-power electronics design for tracking and sensing technologies.



Robbert Elsingerhorst received the bachelor's degree in electrical engineering from Washington State University, Pullman, WA, USA, in 2021.

His undergraduate degree focus was to develop a thorough fundamental understanding in control, communication, and embedded systems. At Pacific Northwest National Laboratory, Richland, WA, USA, he is a Post-Bachelor's Research Associate that provides engineering support to a variety of projects, including the development of marine life tracking systems, marine and avian tags, and water

quality monitoring systems. His current motivation is to continue his education and experience to further improve his ability to contribute to specialized projects.



Zhiqun Daniel Deng received the Ph.D. degree from the University of Illinois at Urbana-Champaign, Champaign, IL, USA, in 2003.

He is a Laboratory Fellow with Pacific Northwest National Laboratory (PNNL), Richland, WA, USA, and an Adjunct Professor of Mechanical Engineering with Virginia Tech, Blacksburg, VA, USA. He directs the Bio-Acoustics and Flow Laboratory, PNNL, an accredited multidisciplinary Research and Development Laboratory, addressing

a broad range of engineering and environmental issues for hydropower, wind, marine, and hydrokinetic energy systems. He has authored or coauthored over 130 journal articles. His research interests include integrated sensor system, energy harvesting, and renewable energy.