Robust TDA-MAC for Practical Underwater Sensor Network Deployment: Lessons from USMART Sea Trials

Nils Morozs*, Paul D. Mitchell, Yuriy Zakharov
Department of Electronic Engineering
University of York
York, YO10 5DD, UK
*nils.morozs@york.ac.uk

Rahul Mourya, Yvan R. Petillot, Tyler Gibney, Mauro Dragone
School of EPS
Heriot-Watt University
Edinburgh, EH14 4AS, UK

Benjamin Sherlock, Jeffrey A. Neasham, Charalampos C. Tsimenidis
School of Engineering
Newcastle University
Newcastle-upon-Tyne, NE1 7RU, UK

Mohammed E. Sayed, Alistair C. McConnell, Simona Aracri, Adam A. Stokes
Institute for Integrated Micro and Nano Systems
School of Engineering, University of Edinburgh
Scottish Microelectronics Centre
Edinburgh, EH9 3FF, UK

ABSTRACT

In this paper, we present the results of deploying the first test prototype of the USMART low cost underwater sensor network in sea trials in Fort William, United Kingdom, on 29 June 2018 and 3 July 2018. We demonstrate the first ever hardware implementation of the Transmit Delay Allocation MAC (TDA-MAC) protocol for data gathering in underwater acoustic sensor networks (UASNs). The results show a successful application of TDA-MAC to remote environmental monitoring, integrating a range of sensor nodes developed by the universities of Heriot-Watt, York, Newcastle and Edinburgh. We focus on the practical challenges and their mitigation strategies related to TDA-MAC to increase its robustness in real-world deployments, compared with theoretical and simulation-based studies. The lessons learned from the sea trials reported in this paper prompted several crucial modifications to TDA-MAC which form a solid foundation for further work on the development of TDA-MAC based UASNs.

KEYWORDS

Medium Access Control, Sea Trials, TDA-MAC, Underwater Acoustic Network, Wireless Sensing

ACM Reference Format:
Nils Morozs*, Paul D. Mitchell, Yuriy Zakharov, Rahul Mourya, Yvan R. Petillot, Tyler Gibney, Mauro Dragone, Benjamin Sherlock, Jeffrey A. Neasham, Charalampos C. Tsimenidis, and Mohammed E. Sayed, Alistair C. McConnell, Simona Aracri, Adam A. Stokes. 2018. Robust TDA-MAC for Practical Underwater Sensor Network Deployment: Lessons from USMART Sea Trials. In *The 13th ACM International Conference on Underwater Networks & Systems (WUWNet'18), December 3–5, 2018, Shenzhen, China.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3291940.3291970

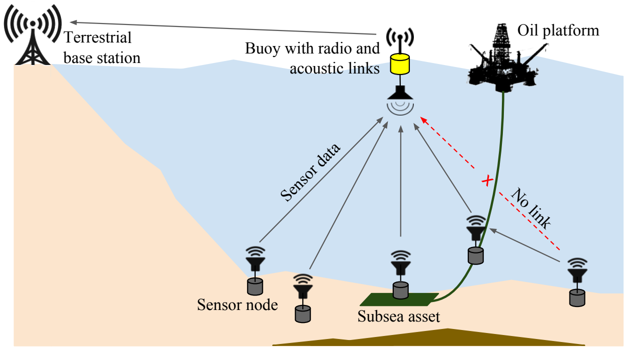


Figure 1: Underwater wireless sensor network deployment with a surface buoy used as the gateway node.

1 INTRODUCTION

The use of wireless sensor networks (WSNs) for remotely monitoring the ocean environment is becoming an increasingly popular research subject due to modern developments in underwater acoustic modem technologies [2, 8]. It is proposed as a solution to a range of underwater sensing tasks, such as water pollution measurements [10], fish tracking [3] and seismic monitoring [14]. The WSN approach to ocean monitoring provides significant advantages over the traditional deployment of data logging sensor nodes from dedicated ships, because WSNs allow flexible long term deployments and eliminate the need to retrieve the sensor nodes from the sea bed to collect the data. For example, Figure 1 shows a typical underwater WSN deployment scenario that is considered in this study. A buoy is used as the surface gateway node to gather the readings from sensor nodes deployed underwater via acoustic communications, and relay these sensor readings to an on-shore base station via a wireless radio link.

In contrast with terrestrial wireless communication systems, underwater radio propagation is severely limited in range due to high

WUWNet'18, December 3–5, 2018, Shenzhen, China

ACM ISBN 978-1-4503-6193-4/18/12.
https://doi.org/10.1145/3291940.3291970

absorption of electromagnetic (EM) waves in seawater, while optical communications suffer from both high absorption and optical scattering [12]. Acoustic waves are the preferred practical medium of communication in the underwater environment; they exhibit significantly better propagation characteristics compared with EM waves. However, acoustic communications are fundamentally limited by the low sound propagation speed - approximately 1500 m/s in water - and low bandwidth with carrier frequencies typically limited to tens of kHz, or lower for long range transmissions [8, 12].

The long propagation delays of acoustic signals present a significant challenge in medium access control (MAC), i.e. coordinating transmissions of multiple acoustic communication nodes potentially spaced kilometres apart. Much of the well-established research on MAC in underwater acoustic networks (UANs) focuses on schedule-based time division multiple access (TDMA) protocols. There, the nodes are scheduled to transmit their packets in particular time slots such that they arrive at the intended receivers without collisions, e.g. [6, 11, 13]. Schedule-based MAC schemes do not involve contention for communication resources, thus removing the need for control signalling, e.g. Request-to-Send (RTS) and Clear-to-Send (CTS), in order to establish collision-free links. Therefore, they are capable of achieving high throughput by efficiently scheduling transmissions in a way that results in a stream of data packets separated by guard intervals at the intended receivers. However, the drawback of the scheduling protocols is their need for clock synchronization across different nodes, which is a challenging task in UANs due to long propagation delays, noisy time-varying multipath channels, and the signaling overhead that is often not negligible compared with terrestrial radio systems [4, 8].

In [15] we designed Transmit Delay Allocation MAC (TDA-MAC), a MAC protocol for scheduled data gathering in UANs which does not require clock synchonization across the sensor nodes. It focuses on UANs that consist of many low cost, low specification sensor nodes, that are currently being investigated in the EPSRC "Smart dust for large scale underwater wireless sensing (USMART)" project [1]. In particular, TDA-MAC has the following key features:

- scalability to large networks (up to hundreds of nodes),
- no requirement for clock synchronization,
- little control signalling overhead,
- low energy consumption,
- low computation requirements.

Between 27 June 2018 and 3 July 2018 we completed a series of initial trials of the USMART sensor network test prototype, which implemented the TDA-MAC protocol for underwater acoustic data gathering, in Loch Linnhe, Fort William, UK. This was facilitated by the EPSRC ORCA Hub (Offshore Robotics for Certification of Assets) [7], and was a collaboration among the universities of Heriot-Watt, York, Newcastle and Edinburgh. As a result of these trials we learned crucial lessons and made several improvements to the TDA-MAC protocol to increase its robustness for future practical deployments. Therefore, the purpose of this paper is two-fold:

(1) To present the results of deploying the initial prototype of the USMART sensor network in sea trials;
(2) To discuss the key issues with TDA-MAC that we encountered in practice, compared with theory and simulations, and to list our improvements that made TDA-MAC more robust in real-world deployments.

The rest of the paper is organized as follows: Section 2 introduces the TDA-MAC protocol and its application to data gathering in UANs; Section 3 gives the implementation details of the USMART sensor network prototype and the sea experiment setup; Section 4 presents the pressure and temperature sensing results from the sea trials; Section 5 discusses the practical considerations and improvements that we made to TDA-MAC based on the sea trials; finally, Section 6 concludes the paper and discusses further work.

2 DATA GATHERING USING TDA-MAC

In [15], we proposed the TDA-MAC protocol for centralized scheduling of data transmissions from sensor nodes connected to the same gateway node, such as in the scenario depicted in Figure 1. Its main advantage over other MAC protocols found in the literature is that it can achieve network throughputs close to the maximum channel capacity for a large number of nodes, without clock synchronization or any other advanced functionality at the sensor nodes. It achieves it by reducing the amount of control signalling at the data gathering stage to a single packet per full set of sensor readings regardless of the network size. Therefore, it shows great potential as a practical solution for efficient data gathering in UANs. Figure 2 gives a simple example that shows the packet flow in TDA-MAC. The gateway node transmits a broadcast data request (REQ) packet that is received by every sensor node at a different time (due to long, variable propagation delays). Each sensor node then waits for a specific (individually assigned) time before transmitting their data packet back to the gateway node.

The gateway node assigns a transmit delay to every sensor node using the following iterative equation:

$$\tau_{\mathrm{tx}}[n] = \tau_{\mathrm{tx}}[n\text{-}1] + T_{\mathrm{dp}}[n\text{-}1] + T_{\mathrm{g}}[n\text{-}1] - 2(\tau_{\mathrm{p}}[n] - \tau_{\mathrm{p}}[n\text{-}1]), \tag{1}$$

where $\tau_{\mathrm{p}}[n]$ is the estimated propagation delay from the gateway node to the n^{th} sensor node, $\tau_{\mathrm{tx}}[n]$ is the transmit delay assigned to the n^{th} sensor node, $\tau_{\mathrm{tx}}[1] = 0$, i.e. the first node starts transmitting its data packet as soon as it receives the REQ packet from the gateway node, $T_{\mathrm{dp}}[n]$ is the duration of the n^{th} node's data packet and $T_{\mathrm{g}}[n]$ is the guard interval after the n^{th} node's data packet reception at the gateway node. The nodes in the $\boldsymbol{\tau}_{\mathrm{tx}}$ =

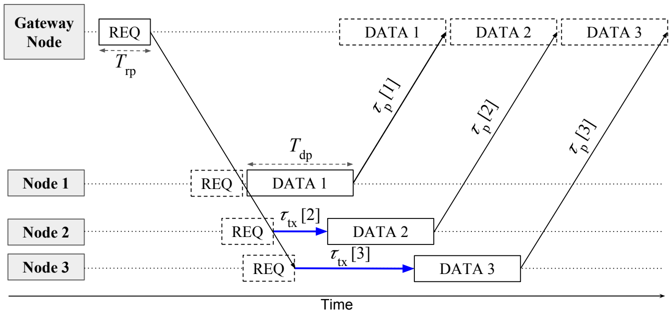


Figure 2: TDA-MAC packet flow example, where a single gateway node gathers data from three sensor nodes [16].

Algorithm 1 TDA-MAC algorithm implementation on a sensor node; TDI - transmit delay instruction, REQ - data request packet

```
if ping packet received from gateway node then
    Transmit ping packet back to gateway node
end if
if TDI packet received from gateway node then
    Store transmit delay allocated to this node
end if
if REQ packet received from gateway node then
    Schedule packet transmission with allocated delay
end if
```

$(\tau_{\mathrm{tx}}[1], \tau_{\mathrm{tx}}[2], ..., \tau_{\mathrm{tx}}[N])$ and $\boldsymbol{\tau}_{\mathrm{p}} = (\tau_{\mathrm{p}}[1], \tau_{\mathrm{p}}[2], ..., \tau_{\mathrm{p}}[N])$ vectors are sorted from the shortest to longest propagation delay from the gateway node. Transmit delays calculated using (1) may be negative. In those cases they are set to zero before continuing to iterate over the rest of the nodes in $\boldsymbol{\tau}_{\mathrm{tx}}$.

The only prerequisite for implementing TDA-MAC is the knowledge of propagation delays between the gateway and sensor nodes, which is measured using a sequence of ping signals during the initial network deployment. Afterwards, during the normal operation of the network, the gateway node can continuously monitor the accuracy of the estimated propagation delays by measuring the error in the timing of the received data packets. For full details of the initialization process and operation of TDA-MAC, see [15].

The guard interval $T_{\mathrm{g}}[n]$ is an important design parameter used to avoid packet collisions due to inaccuracies in propagation delay estimates, slow variations in node positions and the multipath spread. For example, a 100 ms guard interval can tolerate approximately up to 150 m changes in relative node positions before the transmit delays have to be adjusted by sending out updated transmit delay instructions (TDI) to the nodes in question.

One of the advantages of TDA-MAC compared with other schedule-based MAC protocols is the simplicity of implementation at the sensor nodes, which lends itself well to large scale, low cost sensor network deployments. Algorithm 1 shows the functionality required at the sensor nodes to enable TDA-MAC data gathering. It consists of three basic operations, responding to one of the three types of packet received from the gateway node.

3 USMART SENSOR NETWORK PROTOTYPE

In this section we give details of the initial USMART sensor network prototype built for the ORCA Hub sea trials that took place in Fort William, UK between 27 June 2018 and 3 July 2018. The network consisted of different sensor nodes and a gateway node gathering their data using TDA-MAC. Table 1 summarizes the key components used in the experiments discussed in this paper, while the rest of this section describes them in more detail.

Table 1: Summary of the hardware used in the experiments

Module	Description
Acoustic Nanomodems	Miniature, low-cost, low-power acoustic modems developed at Newcastle University, 24-28 kHz frequency range, 168 dB re 1μPa @ 1m source level (Figure 3)
Evologics SONOBOT	Autonomous surface vehicle carrying gateway/sensor nodes (Figure 6)
DS18B20 sensor	Temperature (T) sensor, integrated into sensor node v1 via OneWire interface
MS5837-30BA sensor	Temperature + pressure (T+P) sensor, integrated into sensor node v2 via I^{2}C
Sensor node v1a (SNv1a)	Raspberry Pi 3B+, Nanomodem, DS18B20 T sensor, 5V Li-Ion battery (Figure 4a)
Sensor node v1b (SNv1b)	MicroPython pyBoard v1.1 development board with STM32F405RG microcontroller, Nanomodem, DS18B20 T sensor, 4xAA battery pack (same design as Figure 4a)
Sensor node v2 (SNv2)	Raspberry Pi Zero, Nanomodem, MS5837-30BA T+P sensor, 5V Li-Ion battery, watertight case from BlueRobotics (Figure 4b)
Limpet sensor node (LSN)	Sensor node (PCB) with 9 exteroceptive sensors (temperature, pressure, humidity, optical, distance, sound, magnetic field, accelerometer and gyroscope), Nanomodem, 3.6V Li-Ion Battery, MAX3232 (RS-232 To TTL adapter), watertight case from BlueRobotics (Figure 4c)
Gateway node	Repurposed SNv1a (with Raspberry Pi), running gateway node software via SSH

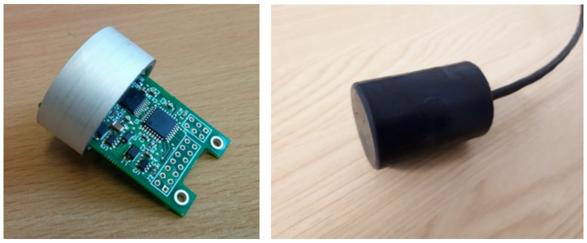
3.1 Acoustic Nanomodems

Figure 3 shows the miniature, low-cost, low-power "Nanomodems" developed at Newcastle University which measure 42 mm diameter by 60 mm long. Operating in the acoustic frequency band 24-28 kHz, the modems utilise a simple but highly robust spread spectrum modulation scheme based on binary orthogonal keying using chirp waveforms (chirp-BOK). This achieves data transmission at 40 bps over ranges up to 2 km transmitting a source level of 168 dB re 1μPa @ 1m (0.5 W), while the receiver power consumption is < 10 mW. The range between devices can be measured using ping packets to a resolution of 10 cm. Interfacing to a PC or other processor platform is via RS232 at 9600 baud. Implementation of a robust spread spectrum receiver with minimal energy and processor cost has been enabled by several innovations in sparse signal processing to reduce the computational load whilst maintaining high processing gain and immunity to the severe multipath distortions commonly found in the underwater channel. These devices were developed to provide a communication and positioning building block for large scale, cost effective WSNs to be constructed subsea.

3.2 Sensor Nodes

Throughout the preparations for the sea trials we developed four types of sensor nodes:

- **Sensor node v1a (SNv1a)**, shown in Figure 4a, comprises a Raspberry Pi 3B+, a 5V (10000 mAh) Li-Ion power supply, a GPS module for positioning, a DS18B20 temperature sensor, contained in a waterproof (IP67) enclosure. A Nanomodem



(a) PCB + transducer

(b) Encapsulated

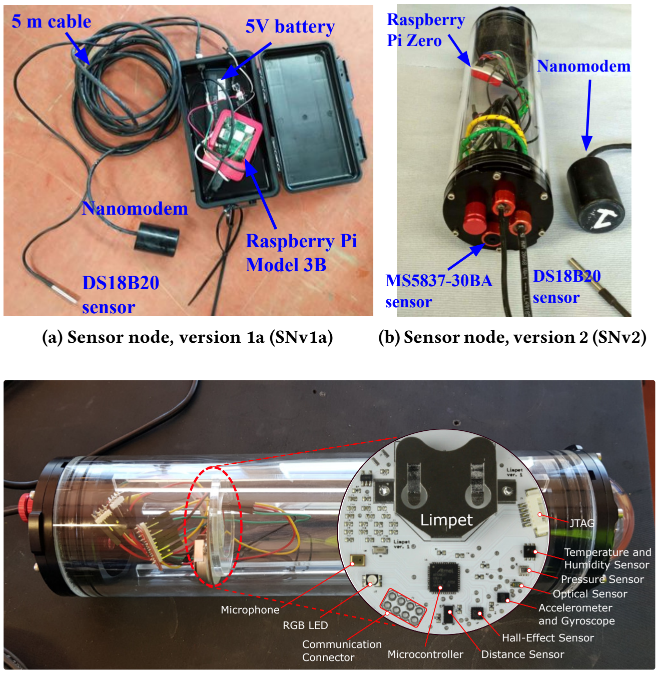
Figure 3: Acoustic "Nanomodems" developed by Newcastle University

is connected to the Raspberry Pi via RS232 and hangs several meters below water surface using the 5 m cable seen in Figure 4a.

- **Sensor node v1b (SNv1b)** follows the same design as SNv1a, except it uses the STM32F405RG microcontroller on the MicroPython pyBoard v1.1 as the computing unit, and is powered by a 4×AA battery pack.
- **Sensor node v2 (SNv2)**, shown in Figure 4b, is a more recent, robust design compared with SNv1a/b. It consists of a Raspberry Pi Zero, a 5V (10000 mAh) Li-Ion power supply, a DS18B20 temperature sensor, MS5837-30BA pressure + temperature sensor, all contained in a watertight (up to 1000 m depth) enclosure by BlueRobotics.
- **Limpet sensor node (LSN)**, shown in Figure 4c, uses the same enclosure as SNv2, but incorporates a novel Limpet sensor platform developed by the University of Edinburgh, described in detail in Subsection 3.3.

3.3 Limpet Node

The Limpet is a multi-sensing platform designed to be low-cost and highly manufacturable, which can be deployed in large collectives for monitoring offshore energy platforms [17]. It is designed to be one part of a heterogeneous collection of field robots (drones, UAVs, mobile legged robots etc.) that together comprise the ORCA Hub system for asset certification and management [7]. The Limpet is equipped with nine exteroceptive sensors: temperature, pressure, humidity, optical, distance, sound, magnetic field, accelerometer and gyroscope. The total cost of the electronic components used in the design of the Limpet is approx. £22. The Limpet consists of a single two-layer printed circuit board (PCB) and a detachable Li-Ion coin cell battery. It is a fully integrated PCB incorporating a low-power microcontroller (ATSAMD21G18A), RGB LED (LTSTN683EGBW), battery holder (BK-877) for a rechargeable Li-ion battery (LIR2477), charging IC (MCP73812T), charger connector, programming port [JTAG] (Molex 532610571), and a communication connector as shown in Figure 4c. The sensors on the Limpet, except for the microphone, are controlled by the microcontroller through the I^{2}C bus. The Limpet is powered with a rechargeable 3.6V 160 mAh Lithium-Ion coin cell battery and has a battery life @160 mAh of 0.87 to 1600 hours.



(a) Sensor node, version 1a (SNv1a)

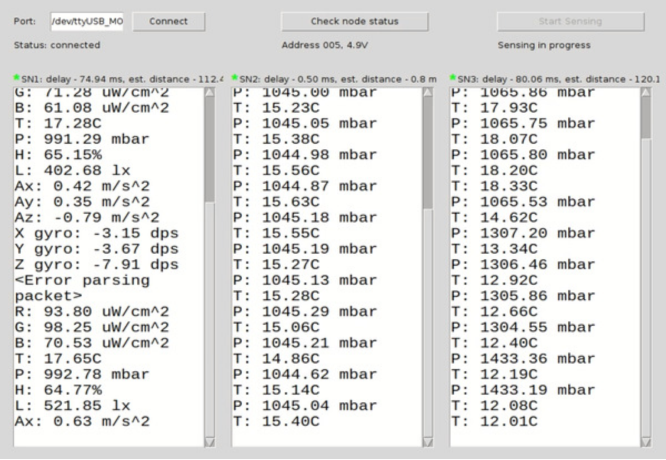
(b) Sensor node, version 2 (SNv2)

(c) Limpet sensor node (LSN)

Figure 4: Sensor nodes used in the sea trials

3.4 Gateway Node

Figure 5 shows a screenshot of the GUI written in Python 2.7 to be used at the gateway node. It controls the acoustic Nanomodem via the serial port interface using a USB-RS232 adapter. The "Start sensing" button initiates the TDA-MAC protocol, which starts by pinging every sensor node sequentially and noting the measured propagation delays. It then proceeds to send the TDI packet to



Port: /dev/ttyUSB_MO

Status: connected

Address 005, 4.9V

Sensing in progress

SN1: delay - 74.94 ms, est. distance - 112.4

```
G: 71.28 uW/cm^2
B: 61.08 uW/cm^2
T: 17.28C
P: 991.29 mbar
H: 65.15%
L: 402.68 lx
Ax: 0.42 m/s^2
Ay: 0.35 m/s^2
Az: -0.79 m/s^2
X gyro: -3.15 dps
Y gyro: -3.67 dps
Z gyro: -7.91 dps
<Error parsing
packet>
R: 93.80 uW/cm^2
G: 98.25 uW/cm^2
B: 70.53 uW/cm^2
T: 17.65C
P: 992.78 mbar
H: 64.77%
L: 521.85 lx
Ax: 0.63 m/s^2
```

SN2: delay - 0.50 ms, est. distance - 0.8 m

```
P: 1045.00 mbar
T: 15.23C
P: 1045.05 mbar
T: 15.38C
P: 1044.98 mbar
T: 15.56C
P: 1044.87 mbar
T: 15.63C
P: 1045.18 mbar
T: 15.55C
P: 1045.19 mbar
T: 15.27C
P: 1045.13 mbar
T: 15.28C
P: 1045.29 mbar
T: 15.06C
P: 1045.21 mbar
T: 14.86C
P: 1044.62 mbar
T: 15.14C
P: 1045.04 mbar
T: 15.40C
```

SN3: delay - 80.06 ms, est. distance - 120.1

```
P: 1065.86 mbar
T: 17.93C
P: 1065.75 mbar
T: 18.07C
P: 1065.80 mbar
T: 18.20C
T: 18.33C
P: 1065.53 mbar
T: 14.62C
P: 1307.20 mbar
T: 13.34C
P: 1306.46 mbar
T: 12.92C
P: 1305.86 mbar
T: 12.66C
P: 1304.55 mbar
T: 12.40C
P: 1433.36 mbar
T: 12.19C
P: 1433.19 mbar
T: 12.08C
T: 12.01C
```

Figure 5: TDA-MAC Python GUI on the gateway node, recorded during an experiment with one Limpet sensor node (LSN) and two pressure & temperature nodes (SNv2)

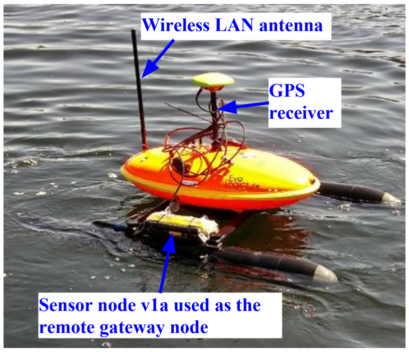


Figure 6: Evologics SONOBOT - autonomous surface vehicle with the gateway node (repurposed SNv1a) attached to it

every sensor node which contains the delay value that the given node has to wait before transmitting data as shown in Figure 2. When it receives a data packet from any sensor node, it displays its contents in the corresponding text panel. For example, in Figure 5 it is gathering data from three sensor nodes - one LSN and two SNv2. The first sensor node is cycling through the different sensor types available on the Limpet, while the other two nodes only have a temperature and pressure sensor available.

This gateway node software was executed both directly on a PC/laptop for initial tests, and remotely on a repurposed Raspberry Pi based sensor node (SNv1a) mounted on the Evologics SONOBOT surface vehicle as shown in Figure 6. Since the SONOBOT vehicle was equipped with a Wi-Fi access point, it allowed us to connect to the Raspberry Pi computer using a Wi-FI repeater on the pier and run the gateway node software via SSH.

4 UASN EXPERIMENT RESULTS

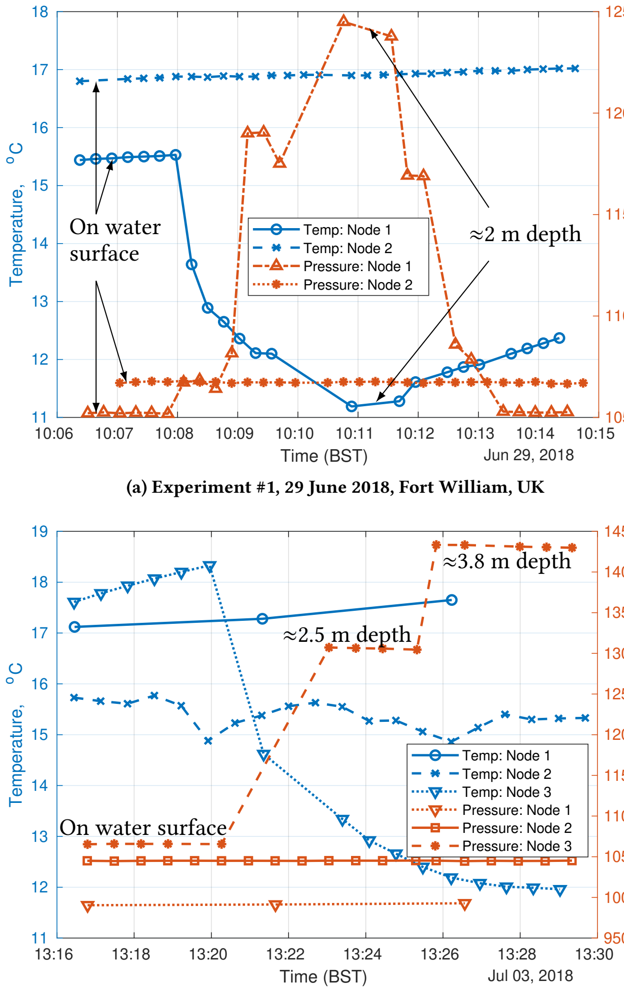
In this section, we discuss the examples of successful deployment of the initial USMART sensor network prototype described in Section 3. Table 2 summarises the setup of the two experiments reported in this section.

Table 2: Description of the reported experiments

Date	Experiment
29/06/18	**#1** Gateway node on SONOBOT ≈100 m away from the pier, SNv2 × 2 deployed from pier approx. 120 and 130 m away from the SONOBOT
03/07/18	**#2** Gateway node on SONOBOT ≈100 m away from pier, SNv2 on SONOBOT, SNv2 and LSN deployed from pier approx. 110 and 120 m away from the SONOBOT

4.1 Temperature and Pressure Sensing Results

Figure 7a shows a plot of the temperature and pressure data recorded live on 29 June 2018 in Experiment #1 from Table 2. The gateway node was deployed on the SONOBOT vehicle with an SSH connection to the laptop on shore. It was gathering data from two sensor nodes (SNv2) by alternating between pressure and temperature readings using a modified broadcast REQ packet structure, which included a sensor type field. This was due to the limited payload length of the current Nanomodem packet structure (7 bytes). In future versions of the USMART sensor nodes, many types of sensor readings will be integrated into a single data packet if needed.



(a) Experiment #1, 29 June 2018, Fort William, UK

(b) Experiment #2, 3 July 2018, Fort William, UK

Figure 7: Live temperature and pressure sensor readings with one of the nodes having adjustable depth, obtained via TDA-MAC in sea experiments described in Table 2

Both Nanomodems were submerged several metres underwater, while Node 2 sensor payload was attached to the pier above water and Node 1 was attached to a pulley for adjustable depth. This enabled us to test the sensors in a live TDA-MAC trial by submerging Node 1 approx. 2 m underwater to observe the changes in pressure and temperature above and below the surface. Despite some packet loss, in particular when Node 1 was submerged deeper and the acoustic Nanomodem was close to sea bottom, most of the sensor data was successfully delivered to the gateway node via TDA-MAC to monitor the changes in pressure and temperature in real-time.

Figure 7b shows the results from Experiment #2 from Table 2, a similar experiment but with three sensor nodes. In this case the majority of the packets were also successfully delivered to the gateway node. The plot includes only 3 pressure and temperature sensor readings from Node 1, because this was the Limpet sensor node which cycled through all the different sensors it integrated in addition to temperature and pressure as described in Subsection 3.3. The live GUI output from this experiment is shown in Figure 5, where the gateway node was receiving many different types of sensor readings from Node 1, while the other two nodes were alternating between pressure and temperature.

4.2 Throughput and Packet Loss Performance

Table 3 summarizes the network performance at the MAC level in the two sea experiments discussed in the previous subsection. 20% of the sensor data packets were lost in Experiment #1, and 10% in Experiment #2. Packet loss occurred due to several factors such as:

- obstructed signal paths and disturbance of the water column due to boat and ship traffic;
- challenging multipath propagation with signal reflections from the harbour infrastructure and the sea bed;
- collisions between data packets at the gateway node, and between data and REQ packets at sensor nodes due to errors in packet scheduling.

However, 10/20% packet loss does not account for the overall loss in network throughput shown in Table 3. The raw bitrate of the Nanomodems is 40 bps as described in Subsection 3.1. Part of this channel capacity is occupied by the header and footer data, e.g. the synchronisation waveform, node address, CRC, etc. This leaves the channel capacity in terms of the data payload at approximately 27 bps, if the maximum packet length of 7 bytes is used.

The sensor network achieved the total throughput of 37% and 24% of the maximum 27 bps channel capacity, which is far lower than the near-optimal channel utilization of TDA-MAC achieved in simulation experiments in [15, 16]. The factors that contributed to the throughput loss in the reported sea experiments, in addition to the packet loss, are the following (most to least significant):

- The guard intervals between the scheduled packet slots were increased to 1 s in Experiment #1 and 3 s in Experiment #2 to accommodate for sporadic errors in packet timing;
- The small number of nodes, compared with simulations, resulted in the increase in proportion of channel airtime used for control signalling, i.e. REQ packet for every 2/3 data transmissions, e.g. instead of 100 in simulations;
- The data payload was 6 bytes which reduced the maximum channel capacity by ≈5%, compared with the 7-byte payload;

The maximum achievable throughput of the TDA-MAC protocol was derived in [15]. For fixed length data packets and guard intervals, it is given by the following expression:

$$\gamma_{\max} = \frac{NT_{\mathrm{dp}}}{T_{\mathrm{rp}} + 2\min_{n}\{\tau_{\mathrm{p}}[n]\} + N(T_{\mathrm{dp}} + T_{\mathrm{g}})}, \tag{2}$$

where $\gamma_{\max}$ is the network throughput normalized by the channel capacity, T_{dp} is the data packet duration, T_{rp} is the REQ packet duration, $\tau_{\mathrm{p}}[n]$ is the propagation delay to the n^{th} node, T_{g} is the guard interval, and N is the number of sensor nodes. It shows that the increase in the guard interval T_{g} and decrease in the number of nodes N have a direct negative impact on the maximum achievable network throughput of TDA-MAC.

Although the high throughput capability of TDA-MAC was not fully exploited in the sea trials due to large guard intervals, these experiments provided valuable data that helped us identify practical considerations of the TDA-MAC protocol that needed to be addressed. In Section 5, we look more closely at the practical issues with TDA-MAC encountered in the sea trials and what actions we took to make the protocol more robust in real-world deployments.

Table 3: Throughput and packet loss in TDA-MAC sea trials

Exp.	Packet loss	Throughput	
#1	20%	10 bps	13 packets/min
#2	10%	6.6 bps	7.9 packets/min

5 PRACTICAL CONSIDERATIONS AND IMPROVEMENTS TO TDA-MAC

In addition to the two sea experiments reported in this paper, we tested our TDA-MAC protocol implementation in a large number of initial and intermediate experiments in the lab environment over-air, in an anechoic water tank at Newcastle University, and in a large water tank and Loch Linnhe at the Underwater Centre in Fort William, UK. During these experiments we discovered several issues which provide additional challenges for TDA-MAC in practical deployments, compared with theory and simulations. Table 4 lists all of the practical considerations that came to light during these trials, strategies to overcome them and modifications we made to the original version of TDA-MAC proposed in [15] and described in Section 2.

First, the network discovery and setup stage requires additional robustness features due to occasional loss of ping packets used for ranging and TDI control packets used to instruct the nodes about their individual transmit delays, the single parameter per sensor node which achieves TDA-MAC scheduling. The first simple modification we made to TDA-MAC was to introduce several attempts at a successful ping exchange with every sensor node to establish a connection and measure the link propagation delay (actions 1, 2 in Table 4). For example, if the gateway node did not receive a ping response from a sensor node, it waits for a timeout period, e.g. 3 seconds, and retransmits the ping packet. If it fails to receive a ping response after 5 successive attempts, it is highly unlikely that it has a usable link with this node; therefore the latter has to be connected via multi-hop links, if multi-hop networking functionality designed in [16] is available.

The original TDA-MAC design involves the gateway node transmitting TDI packets in the "packet train" manner to every sensor node to streamline the network setup stage and make the protocol insensitive to long propagation delays by avoiding bi-directional communication at this stage. However, in practice the loss of a TDI packet at the sensor node results in significant disruption in the subsequent data gathering stage. For example, in one of the initial experiments, one of the sensor nodes failed to receive the

Table 4: Solutions and mitigation strategies for practical issues with TDA-MAC

Practical Issue	Impact	Solution / Mitigation Strategy
Loss of ping packets	Unverified presence of a link; unknown propagation delay	1) 5 attempts at ping exhange per sensor node 2) Nodes that did not respond are either excluded from TDA-MAC data gathering, or connected via other nodes using multi-hop TDA-MAC*
Loss of TDI packets	Wrong transmit times at sensor nodes; packet collisions	3) ACK packet must be sent by sensor node to confirm TDI 4) 5 attempts at sending TDI and receiving the ACK from sensor node 5) Sensor node “activates" the link, i.e. responds to subsequent data requests (REQ), only if it receives a TDI 6) Nodes that did not send ACK are either excluded from TDA-MAC data gathering, or connected via other nodes using multi-hop TDA-MAC*
Loss of REQ packets	Loss of sensor readings	7) Verify packet timing at all nodes to check if REQ-data collisions occur 8) Re-estimate propagation delays between gateway and sensor nodes 9) Re-estimate processing delays, e.g. signal processing & sensing, by timing data packet arrivals 10) Send out updated TDI to one or more sensor nodes, if there was a change 11) Increase guard interval if the above measures failed
Loss of data packets	Loss of sensor readings	12) Perform actions 7-11 to avoid REQ-data and data-data collisions 13) Modify REQ packet structure to target specific nodes, not only broadcast 14) Consistently unresponsive nodes can be reconnected via multi-hop links*
Packet decoding and sensor reading delays	Late transmissions resulting in data packet collisions	15) Enforce a minimum transmit delay for any sensor node in (1), e.g. 1 sec 16) Take precise time of the REQ packet arrival, incorporating/estimating a packet processing delay 17) Measure the sensor reading delay and subtract it from overall Tx delay

* In these trials, the sensor nodes without a link to the gateway node are excluded from the TDA-MAC process. However, the solution to this connectivity issue via multi-hop TDA-MAC is demonstrated in simulation in [16].

TDI packet from the gateway node; as a result, when it received the subsequent REQ packets asking it to send its data, it did not know what its individually assigned transmit delay was, and instead transmitted its data with the default delay of zero, thus causing persistent packet collisions with other sensor nodes. This has prompted further modifications to TDA-MAC (3-6 in Table 4), which crucially incorporate TDI acknowledgement packets from the sensor nodes, and a link activation process whereby only the sensor nodes that received a valid TDI packet are allowed to transmit to the gateway node. These updates are depicted in Figure 8.

Despite establishing a more robust TDA-MAC network discovery and setup stage, as described above, there are still many factors in a practical deployment that result in the errors in packet timing which, if greater than the guard interval between consecutive packet slots, cause collisions. In our TDA-MAC sea trials the following implementation issues gave rise to packet collisions:

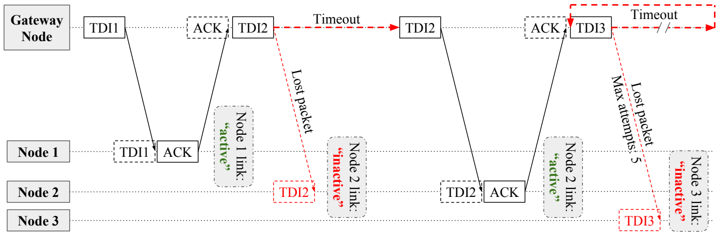


Figure 8: A modified TDA-MAC network setup stage, robust to control packet loss, incorporates TDI ACK packets and link activation triggered by TDI packet reception

- Sensor reading delays: it can take up to 750 ms to get a reading from the DS18B20 sensor [9], whereas the MS5837-30BA sensor returns the readings near-instantly [5].
- Signal processing delays: in many cases, in particular on lower cost nodes, there will be a delay between the end of an acoustic reception and the moment when the packet is decoded and passed to the higher layer.
- Software crashes, e.g. serial/UART port issues, resulted in inaccurate detection of the REQ packet arrival times, which in turn caused scheduling errors of the data packets.
- Different sensor node hardware incurred different software execution time and delay function precision, e.g. the less powerful microcontroller based sensor node (SNv1b) consistently transmitted its readings approx. 1 second late, because its software execution time and the delay function precision of 1 second (as opposed to 1 microsecond on Raspberry Pi) were initially unaccounted for.
- Packet collisions are also possible due to significant changes in node positions and an insufficient guard interval to compensate for them. In these cases, re-estimating the propagation delays and sending updated TDI packets is required.

Algorithm 2 Updated TDA-MAC implementation on a sensor node; blue italic text highlights new features

```
if ping packet received from gateway node then
    Transmit ping packet back to gateway node
end if
if TDI packet received from gateway node then
    Store transmit delay, mark link as “active"
    Send ACK to gateway node
end if
if REQ packet received from gateway node then
    if link is marked “active" and REQ is addressed to this node then
        Note precise time of REQ arrival
        Take sensor readings, and note the sensor reading delay
        Schedule packet transmission with allocated delay
    end if
end if
```

Accurate local timing of the received REQ packets and data transmissions at sensor nodes is crucial in avoiding scheduling errors due to practical factors such as those listed above. Some of these issues were alleviated by further modifications to the TDA-MAC protocol. For example, enforcing a minimum transmit delay, e.g. 1 s instead of zero, allows the sensor reading, signal processing and software execution delays to occur without a negative effect on scheduling. There is also scope for modifying the REQ packet structure to include useful content, e.g. addressing particular nodes rather than always broadcasting, and including corrections to TDIs for some of the nodes. Other issues we discussed above were specific to our implementation, and can be solved by a more precise, lower level implementation of the protocol, compared with the prototype presented in this paper, e.g. using GPIO interface and interrupt routines to accurately detect packet start and end times.

Algorithm 2 summarizes all the key changes to the TDA-MAC implementation at the sensor nodes discussed in this section. It highlights the new features designed to increase the robustness of TDA-MAC in practical deployments in blue italic text, and can be directly compared with the original TDA-MAC protocol implementation in Algorithm 1.

6 CONCLUSIONS AND FURTHER WORK

The sea trials discussed in this paper demonstrated the first ever implementation of the TDA-MAC protocol in practice, and the first test prototype of the USMART low cost underwater sensor network. We showed a successful application of TDA-MAC based underwater acoustic networking to remote environmental monitoring, integrating a range of different sensor nodes developed by the universities of Heriot-Watt, York, Newcastle and Edinburgh.

Furthermore, the experiments we performed during these sea trials have revealed a number of practical challenges that were not addressed during the development of the TDA-MAC protocol in theory and simulations. These practical issues prompted several crucial modifications to the TDA-MAC protocol to make it robust in real-world deployments. We presented the details of the updated network setup stage, and a list of modifications and troubleshooting actions required for successful TDA-MAC data gathering.

The initial USMART network prototype and the lessons learned from the sea trials reported in this paper form a solid foundation for further work on the development of TDA-MAC based UASNs, including larger networks with more sensor nodes, and the development of multi-hop functionality, such as that proposed in [16].

ACKNOWLEDGMENTS

This work was supported by the UK Engineering and Physical Sciences Research Council (EPSRC) through the USMART project (EP/P017975/1). Furthermore, Y.R. Petillot, M. Dragone, A.C. McConnell, S. Aracri and A.A. Stokes acknowledge support from the EPSRC ORCA Hub (EP/R026173/1).

REFERENCES

[1] 2018. EPSRC USMART (EP/P017975/1). https://research.ncl.ac.uk/usmart/
[2] G. Acar and A. E. Adams. 2006. ACMENet: an underwater acoustic sensor network protocol for real-time environmental monitoring in coastal areas. *IEE Proceedings - Radar, Sonar and Navigation* 153, 4 (2006), 365–380.
[3] B.J. Boom, Jiyin He, S. Palazzo, P.X. Huang, C. Beyan, Hsiu-Mei Chou, Fang-Pang Lin, C. Spampinato, and R.B. Fisher. 2014. A research tool for long-term and continuous analysis of fish assemblage in coral-reefs using underwater camera footage. *Ecological Informatics* 23, Supplement C (2014), 83 – 97.
[4] N. Chirdchoo, W.S. Soh, and K.C. Chua. 2008. MU-Sync: A Time Synchronization Protocol for Underwater Mobile Networks. In *Proceedings of the ACM International Workshop on Underwater Networks (WuWNet 2008)*. 8.
[5] TE Connectivity. 2015. Ultra Small Gel Filled Pressure Sensor. MS5837-30BA datasheet.
[6] R. Diamant and L. Lampe. 2011. Spatial Reuse Time-Division Multiple Access for Broadcast Ad Hoc Underwater Acoustic Communication Networks. *IEEE Journal of Oceanic Engineering* 36, 2 (2011), 172–185.
[7] H. Hastie, K. Lohan, M. Chantler, D.A. Robb, S. Ramamoorthy, R. Petrick, S. Vijayakumar, and D. Lane. 2018. The ORCA Hub: Explainable Offshore Robotics through Intelligent Interfaces. In *Proceedings of the ACM/IEEE International Conference on Human Robot Interaction (HRI 2018)*.
[8] J. Heidemann, M. Stojanovic, and M. Zorzi. 2011. Underwater sensor networks: applications, advances and challenges. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 370, 1958 (2011), 158–175.
[9] Maxim Integrated. 2015. Programmable Resolution 1-Wire Digital Thermometer. DS18B20 datasheet.
[10] A. Khan and L. Jenkins. 2008. Undersea wireless sensor network for ocean pollution prevention. In *Proceedings of the International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE 2008)*.
[11] K. Kredo II, P. Djukic, and P. Mohapatra. 2009. STUMP: Exploiting Position Diversity in the Staggered TDMA Underwater MAC Protocol. In *Proceedings of IEEE INFOCOM 2009*.
[12] L. Lanbo, Z. Shengli, and C. Jun-Hong. 2008. Prospects and problems of wireless communication for underwater sensor networks. *Wireless Communications and Mobile Computing* 8, 8 (2008), 977–994.
[13] S. Lmai, M. Chitre, C. Laot, and S. Houcke. 2017. Throughput-efficient super-TDMA MAC transmission schedules in ad hoc linear underwater acoustic networks. *IEEE Journal of Oceanic Engineering* 42, 1 (2017), 156–174.
[14] A. K. Mohapatra, N. Gautam, and R. L. Gibson. 2013. Combined Routing and Node Replacement in Energy-Efficient Underwater Sensor Networks for Seismic Monitoring. *IEEE Journal of Oceanic Engineering* 38, 1 (2013), 80–90.
[15] N. Morozs, P.D. Mitchell, and Y. Zakharov. 2018. TDA-MAC: TDMA Without Clock Synchronization in Underwater Acoustic Networks. *IEEE Access* 6 (2018), 1091–1108.
[16] N. Morozs, P.D. Mitchell, and Y. Zakharov. 2018. Unsynchronized dual-hop scheduling for practical data gathering in underwater sensor networks. In *Proceedings of the IEEE International Conference on Underwater Communications and Networking (UComms 2018)*.
[17] M.E. Sayed, M.P. Nemitz, S. Aracri, A.C. McConnell, R.M. McKenzie, and A.A. Stokes. 2018. The Limpet: A ROS-Enabled Multi-Sensing Platform for the ORCA Hub. *Sensors* 18, 10 (2018), 3487.