Article:
Morozs, Nils, Sherlock, B., Henson, Benjamin et al. (2022) Data Gathering in UWA Sensor Networks: Practical Considerations and Lessons from Sea Trials. Journal of Marine Science and Engineering. 1268. ISSN: 2077-1312

https://doi.org/10.3390/jmse10091268

Journal of *Marine Science and Engineering*

MDPI

Article

Data Gathering in UWA Sensor Networks: Practical Considerations and Lessons from Sea Trials

Nils Morozs [1,*], Benjamin Sherlock [2], Benjamin T. Henson [1], Jeffrey A. Neasham [2] and Paul D. Mitchell [1] and Yuriy Zakharov [1]

[1] School of Physics, Engineering and Technology, University of York, York YO10 5DD, UK
[2] School of Engineering, Newcastle University, Newcastle-upon-Tyne NE1 7RU, UK
* Correspondence: nils.morozs@york.ac.uk

Abstract: Underwater acoustic (UWA) network protocol design is a challenging task due to several factors, such as slow propagation of acoustic waves, low frequency bandwidth and high bit error and frame error rates often encountered in real UWA environments. In this paper, we consider the design of a robust and scalable data gathering protocol for UWA sensor networks (UASNs), focusing on practical considerations and lessons learnt from multiple lake and sea trials. A cross-layer protocol is presented that integrates a network discovery process, intelligent routing, scheduling via Transmit Delay Allocation MAC (TDA-MAC) and multi-node Automatic Repeat Request (ARQ), to facilitate reliable data gathering in practical UASN deployments. Furthermore, this paper presents the details of a novel experimental testbed and underwater sensor node prototype that were used for the trials reported in this study. Based on the results of the trials, important conclusions are drawn on the protocol features required to achieve reliable networked communication in realistic UWA environments. The insights gained from the trials are valuable both for further development of the proposed data gathering protocol, and for the wider UWA networking research community concerned with developing practical solutions for real-world UASN deployments.

Keywords: network protocol; sea trials; TDA-MAC; underwater acoustic network

Citation: Morozs, N.; Sherlock, B.; Henson, B.T.; Neasham, J.A.; Mitchell, P.D.; Zakharov, Y. Data Gathering in UWA Sensor Networks: Practical Considerations and Lessons from Sea Trials. *J. Mar. Sci. Eng.* **2022**, *10*, 1268. https://doi.org/10.3390/jmse10091268

Academic Editor: Alessandro Ridolfi

Received: 28 July 2022
Accepted: 2 September 2022
Published: 8 September 2022

1. Introduction

The use of large scale wireless sensor networks (WSNs) for remotely monitoring the ocean environment is becoming increasingly feasible, owing to the recent developments in underwater acoustic (UWA) modem capabilities [1–4]. Underwater WSN deployments will have a wide range of applications, e.g., water quality monitoring [5], seismic monitoring [6], marine animal tracking [7], off-shore asset monitoring [8], coastal defense [9], etc. 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 to obtain the data. Figure 1 shows an example of the type of an UWA sensor network (UASN) deployment considered in this study. A surface gateway with radio and acoustic communication capabilities is deployed to provide a data sink for the underwater sensor nodes equipped with acoustic modems, e.g., carrying out real-time monitoring of off-shore energy infrastructure, and relaying these sensor readings to an on-shore base station via a wireless radio link.

Compared with terrestrial systems, underwater radio frequency (RF) propagation is severely limited in range due to high absorption in seawater, while optical communications suffer from both high absorption and scattering from solid particles suspended in water [10]. Acoustic waves are the preferred practical medium of communication underwater, since they exhibit significantly better propagation characteristics compared with EM waves and can more readily support both long and short range communication. However, the UWA

communication medium [10,11] presents a number of fundamental challenges for UASN development: extremely slow propagation (sound speed is typically between 1450–1550 m/s), low available bandwidth (typically in the order of several kHz), large multipath delay spread and significant Doppler effect. These challenging channel characteristics necessitate the design of protocols dedicated specifically to UASNs [12,13]. In particular, designing efficient low level network protocols, such as Medium Access Control (MAC) and routing, is essential for successful development and deployment of UASNs at large scale.

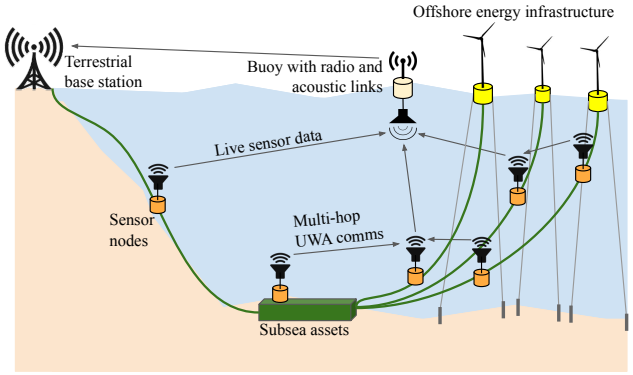


Figure 1. Example of an UWA sensor network deployment for monitoring offshore energy infrastructure.

1.1. Medium Access Control

Many existing MAC protocols designed for UASNs are based on the idea of *contention* for communication resources: the nodes attempt to access a shared channel dynamically, on demand, based on a particular set of rules [14]. These channel access rules are typically based on one or a combination of the following three principles: (1) *Random access*—the nodes are allowed to transmit at random times, whenever they have a packet (the ALOHA principle [15]); (2) *Channel reservation*—before transmitting a packet, a node must reserve channel access via dedicated control signals, typically involving a Request-to-Send (RTS), Clear-to-Send (CTS) exchange between the sender and the receiver [16,17]; (3) *Carrier sensing*—a node must "listen" for the presence of other transmissions on the shared channel and only transmit a packet if the channel is idle [18]. Simple contention-based protocols such as ALOHA, ALOHA-CS or ALOHA-CA [15] can perform well in some UASN scenarios, in particular with low traffic load requirements, but generally the long propagation delays of acoustic signals render them inefficient in high throughput UASN applications due to the increased number of collisions, limited carrier sensing accuracy and higher latency in managing retransmissions. Channel reservation based protocols waste a large part of channel capacity while the nodes are waiting for control signals, e.g., RTS/CTS, acknowledgements (ACKs), to propagate through the water to establish a communication link, e.g., [19,20]. Carrier sensing techniques are also far less efficient in UASNs than in terrestrial radio networks, due to significant delays in detecting transmissions from other nodes [11].

A different class of MAC protocols is based on the principle of Time Division Multiple Access (TDMA), where the nodes are scheduled to transmit their data packets in particular time slots such that the packets arrive at the intended receivers without collisions, e.g., [21–23]. Schedule-based MAC schemes do not involve contention for communication resources, thus removing the need for control signalling in order to establish collision-free links. Therefore, they are capable of achieving high throughput by scheduling the transmissions in a way that results in a stream of data packets separated by guard intervals at the intended receivers. However, in order to mitigate the effect of long propagation delays and achieve high channel utilization and throughput, the solutions in the literature are typically tailored to a specific network geometry, e.g., Super-TDMA for linear ad hoc networks [23] or for regular grid networks [24], where the propagation delays between adjacent nodes are equal to the packet duration, which allows for efficient spatial reuse of time slots in a TDMA schedule. Another established example is the Staggered TDMA Underwater MAC

Protocol (STUMP) [25], which offsets the TDMA frame timing at every node based on knowledge of their propagation delays and, thus, achieves high channel utilization. The key disadvantage of such scheduling protocols is that they are topology-dependent and as such are only suitable for quasi-static networks, in contrast with contention-based protocols that are often topology-agnostic and suitable for dynamic network topologies. However, in most UASN deployments the sensor nodes are typically anchored at particular geographic locations and are only subject to small-scale local mobility caused by water currents, and therefore can be considered quasi-static. The network protocol stack presented in this paper is based on a similar idea of exploiting the knowledge of the network topology for efficient packet scheduling, with a particular focus on designing a network discovery protocol that can facilitate this in practice.

1.2. Routing

Routing is another essential function of a network protocol stack that is required to provide end-to-end connectivity from the sensor nodes to the data sink (e.g., a surface buoy), via multi-hop links if necessary. In many practical UASN deployments, the gateway node will not have direct acoustic connections with every sensor node, e.g., due to channel fading, obstruction, impulsive noise, etc. Therefore, to provide robust network connectivity, support for multi-hop networking is essential. Another advantage of multi-hop networking is the reduction in transmit power and potentially significant energy savings, compared with transmitting at higher power direct to the gateway. There is a large body of literature on routing protocols for WSNs, including many protocols proposed specifically for UASNs [26,27]. Routing protocols range from *static*—with each node storing a routing table indicating the next hop address for every potential packet destination [28], to *dynamic*—where the nodes opportunistically select one or multiple routes for a given packet, with the ability to adapt to time-varying channel conditions, e.g., directional flooding-based routing [29] or hop-by-hop routing [30]. Many routing protocols require the knowledge of the geographical locations of the nodes, e.g., focused beam routing [31], location-aware source routing [32] etc. This requires node *localization*—a significantly more challenging task underwater compared with terrestrial localization, due to the absence of GPS. To alleviate this issue, there are multiple routing protocols in the UASN literature that are based on the nodes' depth [33,34], since local depth readings are relatively simple to obtain via external pressure sensors installed on each node. However, these protocols are designed for UASN deployments where the sea depth is a significant dimension in the overall 3D topology of the network, e.g., in deep ocean, where the depth is comparable with the horizontal size of the network. In this paper, we focus on designing a routing protocol that is generally applicable to quasi-static UASN deployments, and does not assume the knowledge of the nodes' locations. Instead, the network topology information used by the routing protocol is obtained via a dedicated network discovery protocol, designed to support the MAC and routing protocols proposed in this paper.

1.3. Network Discovery

There are many solutions for network discovery developed for terrestrial radio networks. As communication resources are much less constrained than in UASNs, most available solutions rely on regular dedicated beacons to provide the nodes with the requisite information about the network, e.g., [35–37]. However, transferring these solutions to the UWA domain would incur excessive control signalling overheads due to the severe bandwidth and propagation delay constraints of UASNs. There is relatively little research done on network discovery protocols in the UWA domain. Most proposed solutions require a separate network discovery stage, where the nodes broadcast messages to discover their neighbours and adapt to topology changes [38–41]. For example, the DQA-MAC protocol [41] requires a dedicated network setup stage involving the measurement of propagation delays and the assignment of a new packet schedule. Similarly, in the DIVE protocol [40] if a "Hello" packet is received from a new node, the other nodes restart the

neighbour discovery process to update the network topology. The advantage of separating the network discovery and data communication stages of a UASN's operation is in removing the interference and contention between data packets and network discovery probes. However, the disadvantage of this approach is the disruption caused to the normal network operation by the need of periodic network (re)discovery. One of the key contributions of this paper is to report our findings from the sea trials on the duration of network discovery cycles in a practical UASN deployment, and how frequently they need to be repeated to maintain good connectivity within the network.

1.4. Contributions of This Paper

The work presented in this paper focuses on the *cross-layer design* of a network protocol stack for UASNs, including network discovery, routing, MAC, multi-node ARQ and sleep mode management. It is designed to provide a complete networking solution for long-term ocean monitoring using large networks of low cost, low energy sensor nodes, such as those developed in the EPSRC "Smart dust for large scale underwater wireless sensing (USMART)" project [42]. The proposed MAC and routing protocols are based on our previous work on Transmit Delay Allocation MAC (TDA-MAC) [43,44] and dual-hop routing [45], but adapted to focus on energy efficiency and network longevity, instead of maximising the network throughput. New features proposed in this paper include multi-node ARQ, sleep mode management, relay load balancing, and a network discovery protocol that builds a node connectivity tree using a series of ping and test packets without any a priori knowledge of the network topology.

The purpose of this paper is to present a practical solution for a network protocol stack to enable long-term UASN deployments, discuss our findings from the lake and sea trials on its robustness and scalability in a realistic UWA communication environment, and identify potential areas of improvement for the future.

1.5. Paper Structure

The rest of the paper is structured as follows: Section 2 describes the details of our proposed network protocol stack for data gathering in UASNs; Section 3 gives the implementation details of the USMART network prototype and the lake experiment setup, and discusses our findings from initial lake trials; Section 4 presents the novel UWA sensor node hardware and the outcomes of a long-term USMART network deployment in the North Sea; finally, Section 5 concludes the paper and discusses further work.

2. Network Protocol Design

In this section we start by setting out the key requirements of UASN deployments considered in this study, taking a specific example of the USMART network, and then provide a detailed description of our proposed protocol stack for such UASN deployments.

2.1. USMART Network Requirements

The key requirement of the USMART network [42] is its ability to support long-term deployments using low-cost battery powered sensor devices. Therefore, energy efficiency is a priority. To increase the long-term energy efficiency and maximize the lifetime of the network, it will need to operate at a very low duty cycle, e.g., with the sensor nodes waking up to send a reading once an hour or so and then going back to a low power sleep mode until the next transmission. Since the network will operate at such a low duty cycle ($\sim$1 s Tx time per hour), it is also crucial to facilitate reliable delivery of packets within any given data gathering cycle, e.g., with retransmission attempts if necessary, before the nodes go to sleep and can only be reached in the next data gathering cycle (e.g., after an hour).

The topology of the USMART network is depicted in Figure 1. The job of the underwater sensor nodes is to deliver their data to a central gateway node deployed on the sea surface and equipped with both UWA and overwater radio capabilities, which then relays this sensor data to a server or master node on shore (via radio). The underwater nodes

are connected to the surface node either via a direct acoustic link (*single-hop*), or using another sensor node as a relay (a *dual-hop* link), if a single-hop link is not available. Our previous study in [45] showed that extending the network connectivity from single-hop to dual-hop provides a dramatic improvement in the power budget of the underwater sensor nodes (a factor of 40 for an example UASN deployed in a 6×6 km area), i.e., a $40\times$ reduction in acoustic transmit power required to connect all sensor nodes to the gateway node. In principle, it is possible to extend our approach to more hops between the surface and sensor nodes; however, this would result in a significant increase in control signalling and network discovery/setup duration and a significant decrease in the proportion of time some sensor nodes can spend in sleep mode, thus depleting their battery energy more quickly. Therefore, the trade-off point between network robustness, throughput and long-term energy efficiency chosen in our protocol stack design is to limit the UASN connectivity to dual-hop.

2.2. Network Discovery Protocol

Figure 2 gives an overview of the network discovery protocol, a key part of the overall network protocol stack, that is used to discover acoustic links, measure their propagation delays (required for the MAC layer), and estimate the link quality (required to establish the dual-hop routing matrix).

It is infeasible to obtain the full network topology information during the network discovery phase of a UASN deployment, as it would involve an exhaustive $\mathcal{O}(N^2)$ search over all possible acoustic links (N nodes in the network, each sequentially pinging and testing links with the other $N-1$ nodes, thus resulting in $\mathcal{O}(N^2)$ complexity). This would take an excessive amount of time and energy, especially for large-scale networks. To alleviate the effects of $\mathcal{O}(N^2)$ complexity on the duration of the network discovery phase, the approach proposed in this paper is based on a *greedy* algorithm, where the connectivity between each sensor node and the gateway is discovered incrementally, and as soon a *sufficiently good* link to a particular node is found, the algorithm stops searching for other links to this node, even though there might be better links available (if an exhaustive search was carried out). The criterion to stop searching for better links is based on a *link quality threshold* (LQT), which is a tunable protocol parameter that can be adapted to any given UASN deployment. The LQT is defined as an integer with possible values in the range $LQT \in \{0, 1, \ldots, N_{\text{tests}}\}$, where N_{tests} is the number of times each link is tested during the network discovery protocol. For example, $LQT = 5$ will instruct the nodes to keep searching for acoustic links until they find ones that achieved the perfect score (5 out of 5) in the link tests, whereas $LQT = 3$ would relax this constraint by allowing links with 3/5 success rate in the link tests, thus potentially resulting in faster network discovery. Nevertheless, the worst case performance of the proposed LQT-based greedy approach is still bounded by $\mathcal{O}(N^2)$, since the acoustic links are being discovered and tested in a sequential, collision-free manner.

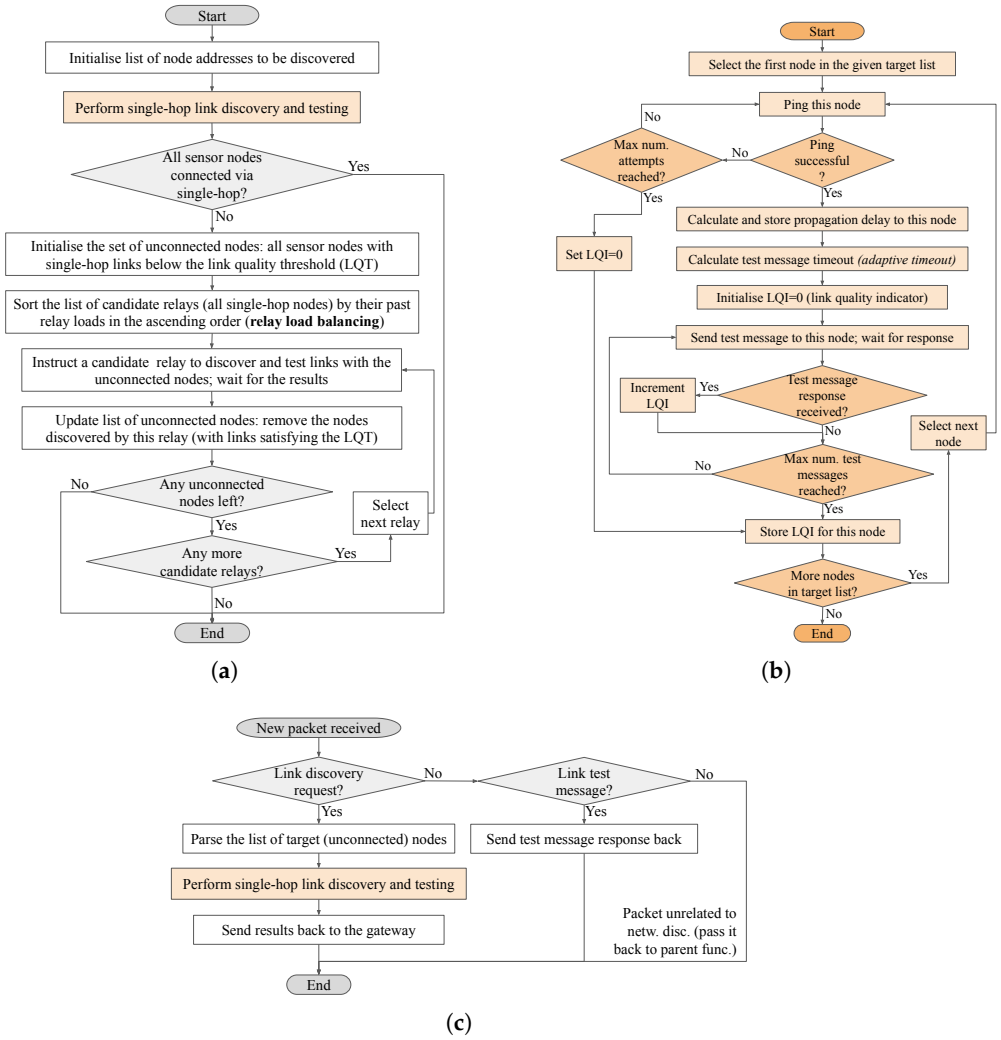


Figure 2. Overview of the network discovery protocol. (**a**) Overall gateway protocol flow; (**b**) Single-hop link discovery and testing function, employed by the gateway and relays; (**c**) Overall sensor node protocol flow.

The process of discovering and testing the links is shown in Figure 2b. When the gateway or a candidate relay needs to discover a link to a particular sensor node (referenced by its address which is known a priori), it starts by sending a ping to see if there is a direct acoustic link with it, and also to measure the propagation delay on this link (based on the recorded roundtrip time of the ping exchange). If it fails to receive a ping response in N_{pings} attempts (e.g., we used $N_{\text{pings}} = 3$ in the trials), the link to this node is deemed non-existent, and the LQI for this link is set to zero: $Q[i, j] = 0$, where $\boldsymbol{Q}$ is an $N \times N$ link quality matrix, i is the index of the master node initiating the ping exchange, and j is the index of the target sensor node. If the ping exchange is successful and there is a direct acoustic link between the two nodes; the propagation delay is measured and stored by the master node. In our previous work in [43,44] the link discovery procedure ended at this point, i.e., successful ping $\rightarrow$ a link was found $\rightarrow$ move on to discovering other links. While this approach was fast and energy-efficient, its key disadvantage was the lack of any information on the link quality (i.e., a successful ping exchange does not mean this will be a reliable link for subsequent data communication). The new approach, shown in Figure 2b, follows the ping exchange with N_{tests} test message exchanges, with the master node i counting the number of times it receives a response from the target node j and storing this number as the LQI for this link $Q[i, j]$. The format and length of these test message transmissions is similar to the normal data transmissions; therefore, they provide a more representative estimate of the

link quality than short ping transmissions. An important parameter for these test message exchanges is the timeout period, i.e., how long does the master node wait for a response before giving up and moving on (to the next test transmission or to the next node). This timeout period is calculated using the propagation delay estimate for the given target node obtained via a preceding ping exchange:

$$\tau_{\text{timeout}}^{\text{test}}[n] = 2\Big(\tau_{\text{p}}[n] + \tau_{\text{tp}} + \tau_{\text{g}}^{\text{nd}}\Big), \tag{1}$$

where τ_{tp} is the test message packet duration, $\tau_{\text{p}}[n]$ is the propagation delay estimate to the target node n, and $\tau_{\text{g}}^{\text{nd}}$ is a network discovery guard interval, e.g., 200 ms, to account for small errors in the $\tau_{\text{p}}[n]$ estimate, signal processing delays, Rx-Tx turnaround time, etc.

Once the master node finishes testing a link to the target node n, it moves on to the next node $(n+1)$ in the list, or finishes the routine if there are no more nodes in the target list.

Figure 2a shows how the single-hop link discovery and testing routine described above fits within the overall network discovery protocol at the gateway node. It starts with a list of all sensor node addresses that are known a priori, i.e., assuming static ID assignment. It then carries out the single-hop link discovery and testing routine, iterating over all sensor nodes in the network, which returns two vectors:

- $\boldsymbol{\tau_{\text{p}}}$—the propagation delay estimate to every sensor node (with default values of zero for any nodes that did not return a ping). This vector is used for TDA-MAC scheduling, as described in Section 2.4.
- $\boldsymbol{q}$—a vector containing the link quality score (between 0 and N_{tests}) for every sensor node.

The values from this vector are extracted into the overall link quality matrix $\boldsymbol{Q}$ stored at the gateway as follows:

$$\forall j \in \{1..N_{\text{sn}}\}, Q[1, j+1] = q[j], \tag{2}$$

since $\boldsymbol{Q}$ includes the gateway (at index 1) in addition to N_{sn} sensor nodes (indexed $\{2..N\}$).

After completing the single-hop link discovery and testing procedure, the gateway assesses the quality of the discovered links against the LQT as follows:

$$M_{\text{single-hop}} = \big\{ j \,\big|\, j \in \{2..N\}, Q[1, j] \geq LQT \big\}, \tag{3}$$

i.e., the indices of all nodes whose direct links with the gateway satisfy the LQT test criterion are added to the single-hop node set $M_{\text{single-hop}}$. The gateway also initializes a binary routing matrix $\boldsymbol{R}$ as follows.

First, $\boldsymbol{R}$ is initialized as an $N \times N$ matrix of zeros:

$$\forall\, i, j \in [1, N],\ R[i, j] = 0, \tag{4}$$

where elements of the routing matrix are defined as: $R[i, j] = 1$ if node j's routing destination is node i, i.e., node i is j's master node (gateway or relay); and $R[i, j] = 0$ if nodes i and j do not communicate directly.

Then, all direct connections between the gateway and sensor nodes are recorded as follows:

$$\forall j \in M_{\text{single-hop}},\ R[1, j] = 1. \tag{5}$$

If $M_{\text{single-hop}}$ contains all sensor nodes, i.e., $M_{\text{single-hop}} = \{2..N\}$, the network discovery protocol finishes here, resulting in the single-hop network topology (all sensor nodes are connected directly to the gateway with sufficiently good links). However, if $M_{\text{single-hop}}$ does not include all sensor nodes, dual-hop links need to be discovered that satisfy the $Q[1, i] \geq LQT, Q[i, j] \geq LQT$ criteria, i.e., the link between the gateway and single-hop node i (acting as relay) and the link between the relay i and the target node j satisfy the LQT.

To discover dual-hop links for the sensor nodes excluded from $M_{\text{single-hop}}$, the gateway first constructs the set of "unconnected nodes", defined as nodes that do not yet have a link to the gateway that satisfies the LQT test criterion:

$$M_{\text{unconnected}} = \{j \mid j \in \{2..N\}, Q[1,j] < LQT\}, \tag{6}$$

i.e.,

$$M_{\text{unconnected}} = \{2..N\} \setminus M_{\text{single-hop}}. \tag{7}$$

It then creates a vector of candidate relays $\boldsymbol{m}_{\textbf{relays}}$ by sorting all single-hop nodes ($M_{\text{single-hop}}$) in the ascending order of their total relay load from previous network cycles $\boldsymbol{l}_{\textbf{relay}}$. This is done to balance the relaying load across the network and achieve long-term energy fairness, as discussed in Section 2.9.

Afterwards, iterating over the nodes in $\boldsymbol{m}_{\textbf{relays}}$, the gateway instructs a candidate relay to perform the single-hop link discovery and testing procedure from Figure 2b only for the nodes in $M_{\text{unconnected}}$. After it receives the link discovery results, those target nodes that were discovered by the relay i with $Q[i,j] \geq LQT$ are added to the routing matrix as its leaf nodes:

$$\forall j \in M_{\text{unconnected}},\ R[i,j] = \begin{cases} 1, & Q[i,j] \geq LQT \\ 0, & Q[i,j] < LQT \end{cases} \tag{8}$$

Then, $M_{\text{unconnected}}$ is updated as follows:

$$M_{\text{unconnected}} \leftarrow \{j \mid j \in M_{\text{unconnected}},\ Q[1,i] \geq LQT \ \wedge\ Q[i,j] \geq LQT\}. \tag{9}$$

i.e., the target nodes that became connected to this relay are taken out of $M_{\text{unconnected}}$.

If $M_{\text{unconnected}}$ is now an empty set, the network discovery protocol is completed with all nodes connected via single- or dual-hop links satisfying the LQT. If there are more nodes left in $M_{\text{unconnected}}$, the gateway repeats the process with the next candidate relay in $\boldsymbol{m}_{\textbf{relays}}$ and so on, until $M_{\text{unconnected}}$ is empty or until all candidate relays have been assessed.

At the end, after iterating over all candidate relays, if some nodes still do not satisfy the LQT, the best possible links (if any) are selected for them:

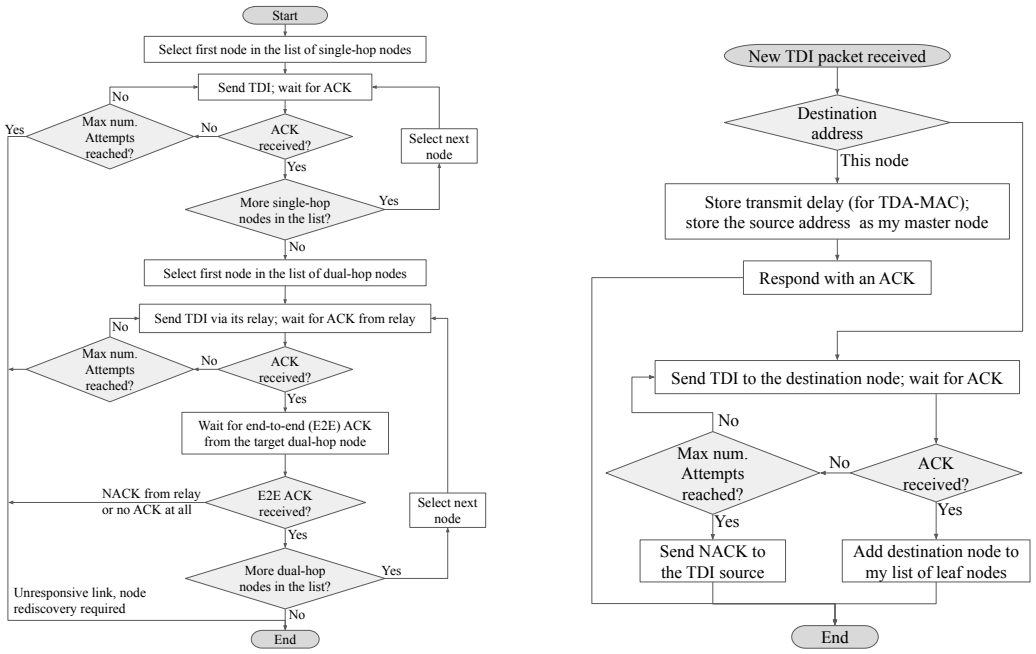
$$\forall j \in M_{\text{unconnected}},\ m_j = \begin{cases} 1, & Q[1,j] = \max\limits_{i \in \{1..N\}} \{Q[i,j]\} \\ \arg\max\limits_{i \in \{2..N\}} \{Q[i,j]\}, & Q[1,j] < \max\limits_{i \in \{1..N\}} \{Q[i,j]\} \\ \varnothing, & \max\limits_{i \in \{1..N\}} \{Q[i,j]\} = 0, \end{cases} \tag{10}$$

where m_j is the master node chosen for node j. If the quality of the single-hop link to node j matches the best quality of all possible relay options, i.e., if $Q[1,j] = \max_{i \in \{1..N\}} \{Q[i,j]\}$, then the single-hop route (direct to gateway) is chosen; otherwise, the best relay option based on the link quality. If there are several options for a relay with equal link quality, a relay with the smallest previous relay load $l_{\text{relay}}[i]$ is chosen. If there are no links to node j with $Q[i,j] > 0$, the node is unreachable (*node outage*) and is excluded from the subsequent data gathering cycles (until the next network (re)discovery cycle).

2.3. Network Setup

After the network discovery phase is completed, the gateway node calculates the TDA-MAC transmit delays for each individual sensor node (based on the propagation delays measured during network discovery), as described in Section 2.4. Then, these are distributed to all sensor nodes by the gateway in the form of Transmit Delay Instruction (TDI) packets. This process is depicted in Figure 3a. First, the gateway sends TDI packets to all single-hop sensor nodes, waiting for an ACK from each node (and retransmitting the TDI if required) before proceeding to the next node. It then iterates through every dual-hop node, and sends a TDI to it via its assigned relay (according to the routing matrix $\boldsymbol{R}$ constructed during the network discovery phase). In this case, it waits for an ACK both

from the relay (to handle any packet loss on the gateway-relay link) and an end-to-end (E2E) ACK from the target dual-hop node. If any sensor nodes fail to acknowledge their TDI, i.e., their link became unresponsive, node rediscovery is required (repeating the network discovery process but only for the unresponsive node(s)).



(**a**) Gateway protocol flow (**b**) Sensor node protocol flow

Figure 3. Overview of the network setup procedure. (**a**) Gateway protocol flow; (**b**) Sensor node protocol flow.

Figure 3b shows the sensor-side protocol flow for the network setup procedure. As for the data gathering and network discovery stages, the sensor nodes' default state is passive listening for commands from their master node. When a new TDI packet is received, if this sensor node is the end destination for it, it stores the transmit delay contained in this TDI, and also stores the source address of the TDI packet as their master node. This achieves the setup of both the MAC layer and the routing protocol with one control packet. However, if the TDI destination address is another sensor node (but the link layer address is this node), then this node tries to deliver the TDI to the end destination and waits for an ACK from it. If it delivers the TDI successfully (with retransmissions if required), the TDI destination node then becomes this node's leaf node (i.e., this node becomes the relay for it). However, if the link to the destination node is unresponsive, and there is no ACK received from it, this node sends a negative ACK (NACK) to its master node to let it trigger node rediscovery (i.e., try to find a different relay for the destination node).

2.4. Data Gathering Protocol

Figure 4 shows four flowcharts that describe the overall operation of the data gathering protocol proposed in this paper.

Figure 4a summarizes the logic performed by the gateway node. It is largely based on the Sequential Dual-Hop TDA-MAC protocol proposed by us in [45], with a modified sequence for transferring data from a relay to the gateway node, and, crucially, with an added retransmission capability via multi-node ARQ, integrated into the TDA-MAC protocol without the need for any additional control transmissions, as described in Section 2.6. At the start of a data gathering cycle, the gateway first instructs all sensor nodes connected directly to it (single-hop nodes) to send their data packets (the transmission times are managed via TDA-MAC explained in Section 2.4). It does so using the process shown in the flowchart in Figure 4c, described in more detail later in this subsection. Then, if the network

topology includes any dual-hop sensor nodes, the gateway loops through every relay node, instructs it to gather the data from its leaf nodes (i.e., sensor nodes connected to this relay) and wait for the relay to report this data back to the gateway. If the relay node fails to initiate a data transfer handshake after a timeout period, the gateway sends another request to this relay, until the data transfer handshake is successful or the maximum number of attempts (e.g., three) is reached.

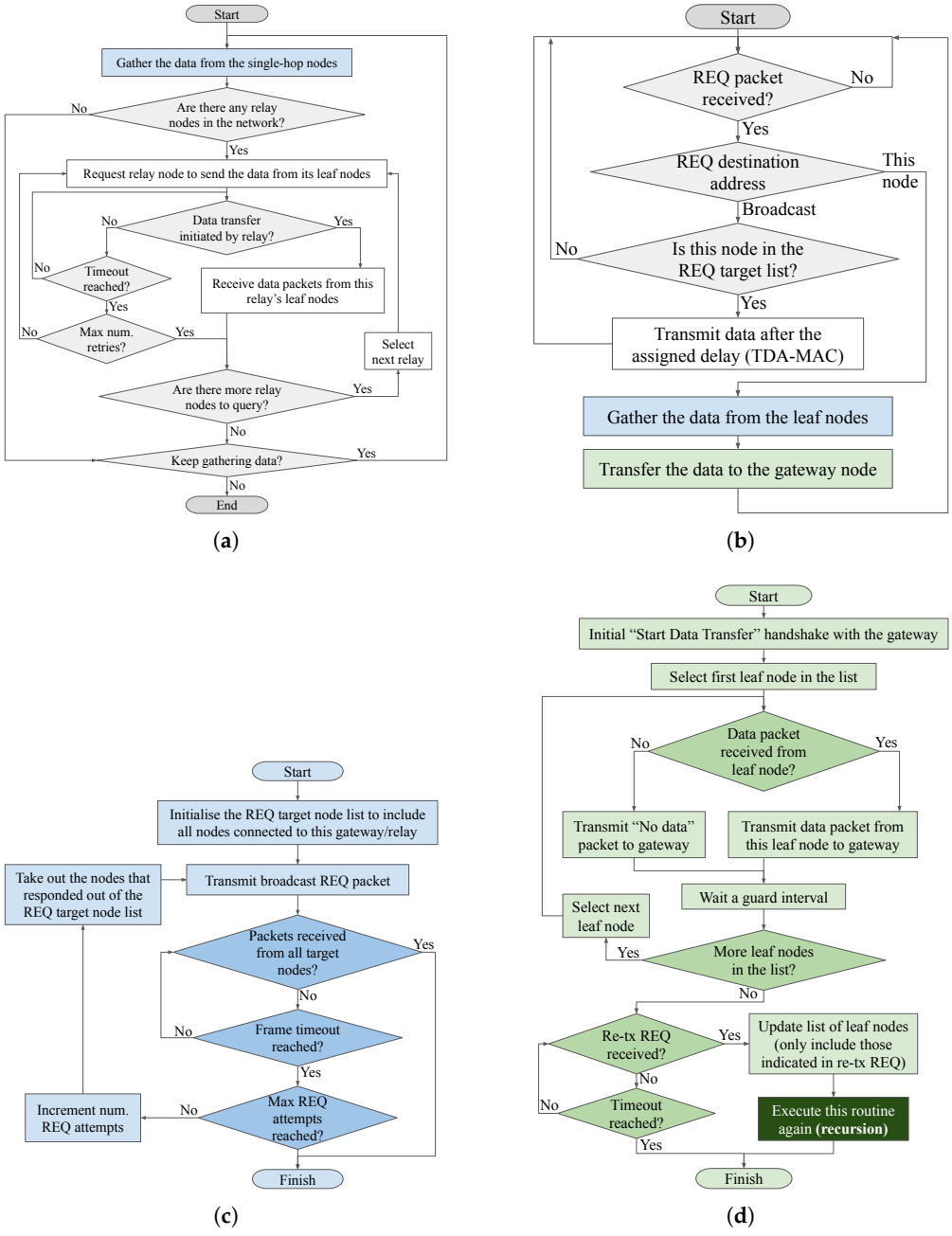


Figure 4. Flowcharts of the data gathering protocol at the gateway and sensor nodes. (**a**) Overall gateway protocol flow; (**b**) Overall sensor node protocol flow; (**c**) Data gathering function invoked by the gateway and relays (based on TDA-MAC); (**d**) Data transfer from a relay to the gateway.

Figure 4b describes the logic of the data gathering protocol at the sensor node side. The default state of a sensor node is passive listening—waiting for a command from its master node (gateway or relay). If a sensor node (SN) receives a broadcast "request for data" (REQ) packet containing this SN's address as one of the target nodes, it transmits its

data packet to the REQ source (gateway or relay) after a preassigned transmit delay (as per the TDA-MAC protocol described in the next subsection). However, if the given SN is a relay, it may also receive a unicast REQ packet from the gateway. This instructs the SN to switch to the *relay mode*: gather the data from its leaf nodes (the function described in Figure 4c), and afterwards report it back to the gateway (using the function described in Figure 4d).

Figure 4c describes the data gathering function that is invoked by the gateway or relay node to gather a set of data packets from the sensor nodes directly connected to it (via single-hop). It is based on the TDA-MAC protocol developed by us in [43] and described in detail in the next subsection. The new addition to this process is the retransmission functionality via multi-node ARQ (MARQ), depicted in Figure 4c and described in Section 2.6.

Finally, Figure 4d describes the process used by a relay node to forward a set of data packets from its leaf nodes to the gateway node. After a relay node finishes gathering the data from its leaf nodes (the process shown in Figure 4c), its job is to deliver these packets to the gateway. It starts with an initial "Start Data Transfer" (SDT) handshake—a predefined short packet transmitted by the relay and acknowledged by the gateway. If the handshake fails, i.e., the relay transmits the SDT packet but does not receive an ACK from the gateway, it tries again until it is successful or until the maximum number of handshake attempts is reached (in which case the node reverts to the passive listening state and waits for future commands from the gateway). After the SDT handshake is completed, the relay transmits a "train of data packets" separated by a guard interval (e.g., 100 ms). If a data packet from any leaf node is missing from the preceding data gathering stage, it transmits a predefined "no packet" instead. After transmitting the data packets, the relay waits for a timeout period (e.g., 10 s) to give the gateway an opportunity to send a retransmission request (Re-Tx REQ) in case it failed to receive any of the data packets sent by the relay. In this case, the Re-Tx REQ will specify the addresses of those nodes whose data packets were lost, so the relay can retransmit only those missing packets using the same procedure as above.

Medium Access Control

The data gathering process shown in Figure 4c involves transmissions from a number of sensor nodes to a common gateway/relay, all using the same acoustic frequency band. The transmissions must be managed using an appropriate Medium Access Control (MAC) protocol to avoid collisions and packet loss at the receiver due to interference from other nodes. The MAC layer of the protocol stack presented in this paper is based on TDA-MAC [43–45]—a protocol for centralized scheduling of data transmissions in a UASN without clock synchronization at the underwater sensor nodes.

Figure 5 shows an illustrative example of the packet flow in TDA-MAC. The master (gateway or relay) node transmits a broadcast REQ packet (of duration τ_{rp}) that is received by every target sensor node at a different time (due to large differences in propagation delays of acoustic links). Each sensor node n then waits for a specific (individually assigned) amount of time $\tau_{tx}[n]$ before transmitting their data packet (of duration $\tau_{dp}[n]$) back to the gateway node, such that the data from all sensor nodes is received in an efficient "packet train" pattern as shown in Figure 5.

Before the data gathering stage, the gateway node calculates a transmit delay for every individual sensor node using the following iterative equation:

$$\tau_{tx}[n] = \tau_{tx}[n\text{-}1] + \tau_{dp}[n\text{-}1] + \tau_g - 2(\tau_p[n] - \tau_p[n\text{-}1]), \quad (11)$$

where $\tau_p[n]$ is the estimated propagation delay from the gateway node to the n^{th} sensor node, $\tau_{tx}[n]$ is the transmit delay assigned to the n^{th} sensor node, $\tau_{tx}[1] = \tau_{min}$, i.e., the first node starts transmitting its data packet as soon as it receives the REQ packet from the gateway node including only a short τ_{min} delay to allow for any device specific signal/data processing delays, $\tau_{dp}[n]$ is the duration of the n^{th} node's data packet and τ_g is the guard interval between scheduled packet receptions. The nodes in the

$\boldsymbol{\tau}_{\mathrm{tx}} = (\tau_{\mathrm{tx}}[1], \tau_{\mathrm{tx}}[2], \ldots, \tau_{\mathrm{tx}}[N_{\mathrm{sn}}])$ and $\boldsymbol{\tau}_{\mathrm{p}} = (\tau_{\mathrm{p}}[1], \tau_{\mathrm{p}}[2], \ldots, \tau_{\mathrm{p}}[N_{\mathrm{sn}}])$ vectors are sorted from the shortest to the longest propagation delay from the gateway node, N_{sn} is the total number of sensor nodes. In some cases, transmit delays calculated using (11) may be negative. Then they are set to τ_{min} before continuing to iterate over the rest of the nodes in $\boldsymbol{\tau}_{\mathrm{tx}}$, i.e., we place the following constraint on $\tau_{\mathrm{tx}}[n]$:

$$\forall n \in [1, N_{\mathrm{sn}}],\ \tau_{\mathrm{tx}}[n] \geq \tau_{\mathrm{min}} \tag{12}$$

These transmit delay values are distributed by the gateway node to all sensor nodes during a network setup stage described in Section 2.3. The key prerequisite for implementing TDA-MAC is the knowledge of propagation delays between the gateway/relay nodes and every target sensor node, i.e., the $\boldsymbol{\tau}_{\mathrm{p}}$ vectors for direct gateway-sensor links and for every relay branch of the network. Those are measured during a dedicated *network discovery* protocol proposed in Section 2.2.

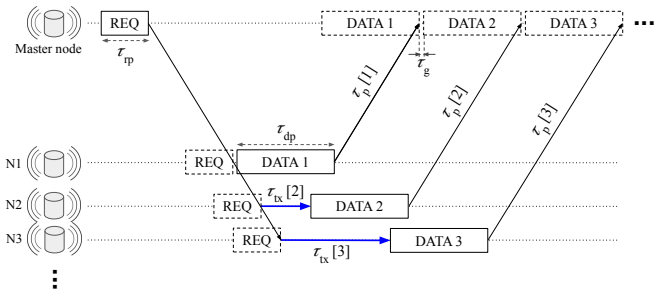
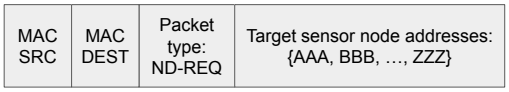


Figure 5. TDA-MAC packet flow [45]. The master node (gateway or relay) sends a broadcast "REQ" packet; when each sensor node receives it, it delays its packet transmission by an allocated transmit delay, which results in an efficient "packet train" reception pattern at the master node.

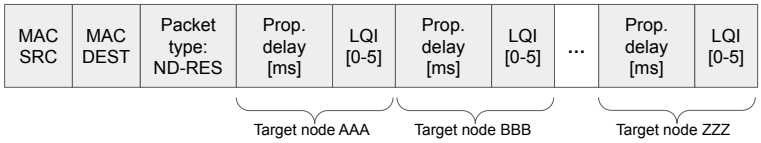
The guard interval (τ_{g}) is an important design parameter that is used to separate consecutive data packets received at the gateway/relay node. It should be long enough to avoid packet collisions due to inaccuracies in propagation delay estimates, slow variations in node positions and the multipath channel delay spread. However, extending the guard interval also increases the amount of idle time on the channel and, thus, reduces the throughput and increases the latency. Therefore, the length of the guard interval should be specifically chosen for a given network deployment and UWA environment, taking into account the key performance metrics, network size, the link lengths, the expected node drift, the reflectivity of the seabed, etc. For example, in the sea deployment reported in Section 4 network reliability and energy efficiency were the priority metrics, whereas high throughput and low latency were not important. Therefore, a long guard interval of 1 s was chosen, in order to reduce the likelihood of interference from possible reverberations of acoustic signals.

2.5. Packet Structure

The structure of the key packets involved in the network discovery protocol is shown in Figure 6. If the gateway node needs to find dual-hop links for a given list of target sensor nodes, it sends a "node discovery request" to a potential relay containing the target node addresses. This prompts the receiving node to carry out the link testing procedure described in Section 2.2. After it finishes, it sends the results back to the gateway in the the format shown in Figure 6b, containing the measured propagation delay and the LQI value for every target node specified in the list.



(**a**) Node discovery request



(**b**) Node discovery results

Figure 6. Network discovery protocol packet structure.

Figure 7 shows the TDI packet structure used in the network setup phase. It includes the network destination address (NET DEST) to distinguish between TDI sent directly from the gateway to the target node, and those intended to set up dual-hop links, i.e., addressed to a node outside of the gateway's range. The address fields of the TDI packet are used by the sensor nodes to set up their routing, by storing the addresses of their master node and leaf node(s) (if any), whereas the "transmit delay" and "subframe duration" fields are used to set up the TDA-MAC parameters at each node for the data gathering stage, with the latter only intended for relay nodes (telling them how long their TDA-MAC subframe is).

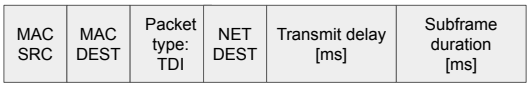


Figure 7. TDI packet structure.

As shown in Figure 5, the TDA-MAC protocol uses a broadcast data request (REQ) packet transmitted by each master node (gateway and relays) to provide all of its leaf nodes (sensor nodes connected to this gateway/relay) with a local time reference for scheduling their data transmissions in a an efficient collision-free pattern.

In the interests of minimising the overall protocol stack overheads and reducing the number of control transmissions required to operate the network, several other key network protocol functions (spanning multiple layers of the protocol stack) can be integrated into each REQ transmission. Figure 8 shows the REQ packet structure proposed in this paper. It includes the following fields:

- *MAC SRC & DEST*—the fields to identify the source and destination address for a given transmission. As explained in Section 2.4, there are two types of REQ - broadcast to leaf nodes, and unicast to a relay; the *MAC DEST* field is used to distinguish between those two types of REQ.
- *Packet type*—a header field used to identify this packet as a REQ.
- *Data type*—a control field identifying the type of data payload to be requested from the senor nodes (if multiple options exist). An example of this is to request sensor readings by default, but occasionally request localization data. As such, this is an (optional) application layer function integrated into the protocol stack at a negligible cost.
- *Time till next frame (TTNF) & Sleep flag*—these two fields are used to schedule the sleep modes at the sensor nodes. This process is described in Section 2.9.
- *REQ index & Target sensor node addresses*—these fields are used to provide multi-node ARQ functionality, as described in Section 2.6.

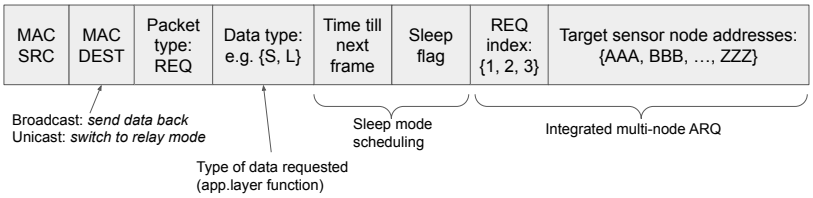


Figure 8. REQ packet structure: integrating multiple network protocol functions from different layers in a single control packet.

It is possible to extend this approach to integrate more network control plane functions within the REQ packets; however, this will depend on the acoustic modem specification, in particular, on the maximum packet size (e.g., 64 bytes for the NMv3 modems used in our trials).

Finally, Figure 9 shows the structure of the data packets, whose purpose is to deliver a given payload of raw data bytes to the gateway node, either directly or via a relay node specified by the MAC DEST field.

Figure 9. Data packet structure.

2.6. Integrated Multi-Node ARQ

The data gathering protocol (Figure 4c) employed by the gateway and relay nodes incorporates a multi-node ARQ function to efficiently handle packet loss from one or more sensor nodes. It works as follows:

1. The master node (gateway/relay) starts by sending a REQ packet targeting all of its leaf nodes (i.e., including the address of each of those nodes in the dedicated REQ packet field shown in Figure 8). It also sets the *REQ index* field to 1.
2. Those sensor nodes that successfully receive this REQ check if their address is included in the target node list. If it is, they send back a data packet using the TDA-MAC delayed transmission principle, as shown in Figure 4b and explained in Section 2.4.
3. If the data packets from all target sensor nodes were successfully received at the master node, the process is completed and there is no need for retransmissions.
4. However, if the data packets from some nodes are missing, the master node transmits another REQ, this time only including those nodes in the target list and incrementing the *REQ index* field.
5. This process repeats until the data packets from all nodes are received, or if the maximum number of attempts has been reached. The maximum number of attempts is the key ARQ parameter providing a trade-off between the network reliability and additional latency and energy consumption required to handle unresponsive nodes. In the experiments reported in this paper, the maximum number of packet transmission attempts is three.

A key parameter for this ARQ process is the timeout value, i.e., how long should the master node wait to receive data packets from all target nodes, before deciding to trigger a retransmission request via the multi-node ARQ process described above. Since the TDA-MAC protocol uses centralized scheduling with a defined timeframe for each packet reception, the value for $\tau_{\text{timeout}}^{\text{REQ}}$ (for a given gateway/relay node) is determined as follows:

$$\tau_{\text{timeout}}^{\text{REQ}} = \max_{n}\{2\,\tau_{\text{p}}[n] + \tau_{\text{tx}}[n] + \tau_{\text{dp}}[n]\} + \tau_{\text{rp}} + \tau_{\text{g}}, \tag{13}$$

i.e., the duration of a full TDA-MAC frame depicted in Figure 5 plus a guard interval τ_{g}; τ_{rp} is the duration of the REQ packet.

2.7. Routing

The routing strategies for TDA-MAC based UASNs were studied in [46]. It showed that the best strategy in terms of throughput is to minimize the number of relays in the network. In this way, the network utilizes the "many-to-one" transmission scheduling of TDA-MAC and minimizes the number of branches in the network topology that need to be resolved sequentially while data gathering. Although throughput is not a key metric in the UASN applications considered in this study (as discussed in Section 2.1), it is still advantageous to choose a high throughput routing strategy in order to shorten the data gathering cycles and thus increase the proportion of time the sensor nodes spend in the sleep state. Figure 10 shows an example of this routing strategy, where 7 out of 20 sensor nodes require a dual-hop connection to the gateway, and they are connected via two relays (the minimum possible number of relays in this case).

In this paper, the static "fewest relays" routing approach described above is extended to also take into account the link quality. Here, instead of using a binary connectivity model for relay selection (i.e., if a link between two nodes exists, it can be chosen for a dual-hop route), the quality of the link must meet a predefined threshold in order to be eligible for selection. There are many different ways to quantify the link quality, e.g., received signal strength, Signal-to-Noise Ratio (SNR), channel multipath structure, etc.; however, the ability to access such PHY layer metrics will vary for different acoustic modems. The protocol stack in this paper uses a modem-agnostic empirical link quality metric—a score out of 5—which is obtained via the network discovery protocol proposed in Section 2.2, which also incrementally sets up a static routing matrix as part of the network discovery process.

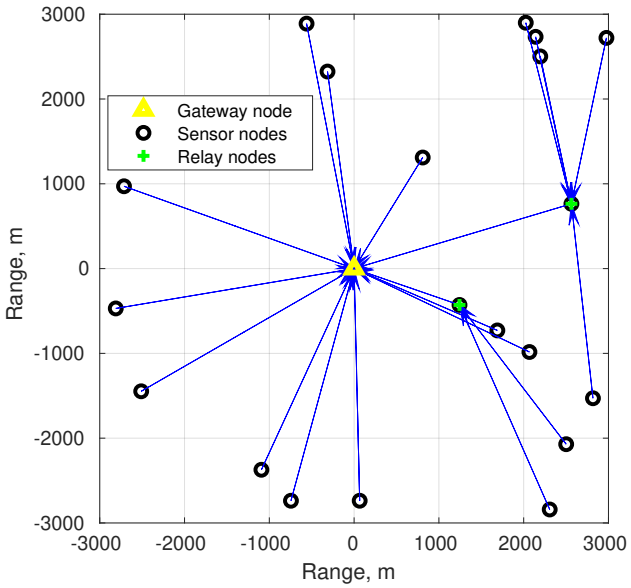


Figure 10. Example of the dual-hop routing strategy applied to a network of 20 sensor nodes at approx. 500 m depth with 3 km communication range. The sensor nodes outside the gateway node's coverage send their data via other sensor nodes. The gateway node is responsible for gathering data packets from all directly connected sensor nodes, and the relay nodes are responsible for gathering data within their respective network branches [45].

The challenge with applying the "fewest relays" routing strategy in a real UASN deployment is to obtain the prerequisite network topology information in order to establish the routing pattern such as that shown in Figure 10. The gateway node would require the knowledge of a full $N \times N$ network link quality matrix (where N is the number of nodes, including the gateway). However, this would involve an exhaustive search over N^2 links, that would take a long time in a UWA environment and would not be scalable for large networks. Furthermore, finding an optimal "fewest relays" routing solution is an NP-Hard

problem (as demonstrated in [45]), so an approximation algorithm is required to find a good, but possibly suboptimal, routing pattern in any given large scale UASN deployment.

Therefore, a solution is required that avoids the need for an exhaustive N^2 link search to derive a global routing strategy (finding a solution to an NP-Hard problem). Addressing this challenge and designing a practical protocol stack that supports such dual-hop routing is a major focus of this paper. In particular, this is addressed by designing a novel network discovery protocol described in Section 2.2.

2.8. Relay Load Balancing and Network Adaptability

A disadvantage of the "fewest relays" routing strategy described above is the significant variability in the energy consumption among the sensor nodes. Nodes that are elected to serve as relays are likely to have multiple sensor nodes connected to them, thus, resulting in many more acoustic transmissions (forwarding other nodes' packets as well as sending their own data) and a higher energy consumption. Since network lifetime is a key metric of the UASNs considered in this study, a means to achieve *energy fairness* over the long-term duration of the network deployment is required.

To achieve energy fairness we propose a centralized relay load balancing scheme, where the gateway maintains a vector $\boldsymbol{l_{relay}}$ to keep track of the long-term relay load of each sensor node (where every element $l_{\text{relay}}[n]$ is defined as an integer tracking the number of transmissions that the corresponding n^{th} sensor node made as a relay). During a periodic network discovery and setup phase (e.g., twice a day), the gateway updates the network routes such that $\boldsymbol{l_{relay}}$ is equalized across all sensor nodes in the long run. This is done as part of the network discovery protocol explained in the next subsection.

Another reason for periodic network rediscovery is to adapt to changes in the UWA communication environment, where some links that were previously reliable may become weak or disappear altogether, in which case the gateway has the capability to discover new reliable routes and use those instead. The network rediscovery process is discussed in more detail in the next subsection.

2.9. Energy Efficiency via Sleep Modes

A key consideration for designing a protocol stack based on TDA-MAC scheduling and static routing is to allow the sensor nodes to enter deep sleep modes (with the receiver, sensing modules, and the main CPU all powered off), while they are not expected to transmit or listen for any packets. Letting the sensor nodes spend most of their time in such sleep states is crucial to achieve long network lifetime on small battery-powered underwater devices.

These sleep modes are managed by two control fields included in the REQ packets shown in Figure 8:

- *Time till next frame (TTNF)*—includes, as the name suggests, the amount of time until the node receiving the REQ packet can expect the next frame to begin, i.e., if it delivered its data packet to the master node and did not receive a retransmission REQ (via multi-node ARQ), it can go to sleep and wake up just when the TTNF elapses (or a few seconds before to be safe). For those sensor nodes that act as relays, they can only go to sleep after they have delivered their own packet as well the packets of all their leaf nodes.
- *Sleep flag*—a binary field, indicating whether the sensor nodes are allowed to go to sleep in this data gathering frame, e.g., if the gateway node wants to initiate a network rediscovery phase it may want to instruct the sensor nodes to stay awake.

The value of the TTNF is updated throughout a data gathering frame by the gateway and relay nodes, every time they send a REQ packet to their leaf nodes. For the first REQ sent by the gateway, TTNF is equal to the data gathering interval (e.g., if data gathering takes place once an hour, then TTNF is 1 h). Then, with every REQ transmitted afterwards, e.g., either for retransmissions or for unicast REQs sent to the relays, TTNF is equal to the initial value (e.g., 1 h) minus the time elapsed since the initial REQ transmission. Similarly

for relay nodes, they run a local timer to measure the time elapsed between receiving the initial REQ from the gateway and sending their own REQ(s) to their leaf nodes, and set TTNF to the difference between the initial value and the time elapsed since then.

3. Lake Trials

In this section we report the outcomes of deploying a prototype USMART network running the protocol stack proposed in this paper in lake trials that took place at the University of York, on the Heslington East campus lake on 27 May 2021.

3.1. Experiment Setup

Figure 11 shows the experimental setup for the Heslington East lake trials. A total of seven sensor nodes + one gateway equipped with acoustic modems were deployed near the shore on the opposite sides of the lake at ∼0.5 m depth. The gateway node consisted of a laptop on shore connected to an acoustic modem deployed in ∼0.5 m deep water via a 10 m cable. The lake has a silty bottom, ∼1 m depth where the sensor nodes are deployed, ≈3–4 m depth in the middle, and a high amount of vegetation (due to late Spring). This provided challenging UWA propagation conditions, with the connectivity among the network nodes significantly restricted by the shallow depth, low reflectivity of the lake bottom, and dense vegetation blocking many potential acoustic links. These conditions have restricted the range of possible topologies that could be established by the network discovery protocol, as shown later in Figure 12; however, this is representative of real large-scale deployments, where many nodes, especially those at the edge of the network, may have a limited choice of routes to communicate with the gateway node.

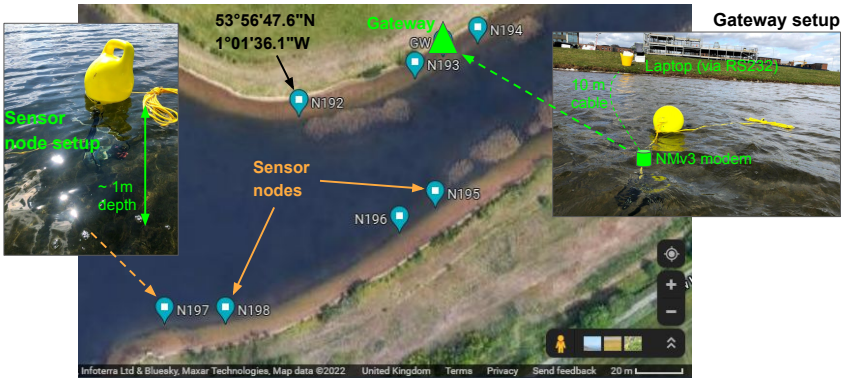
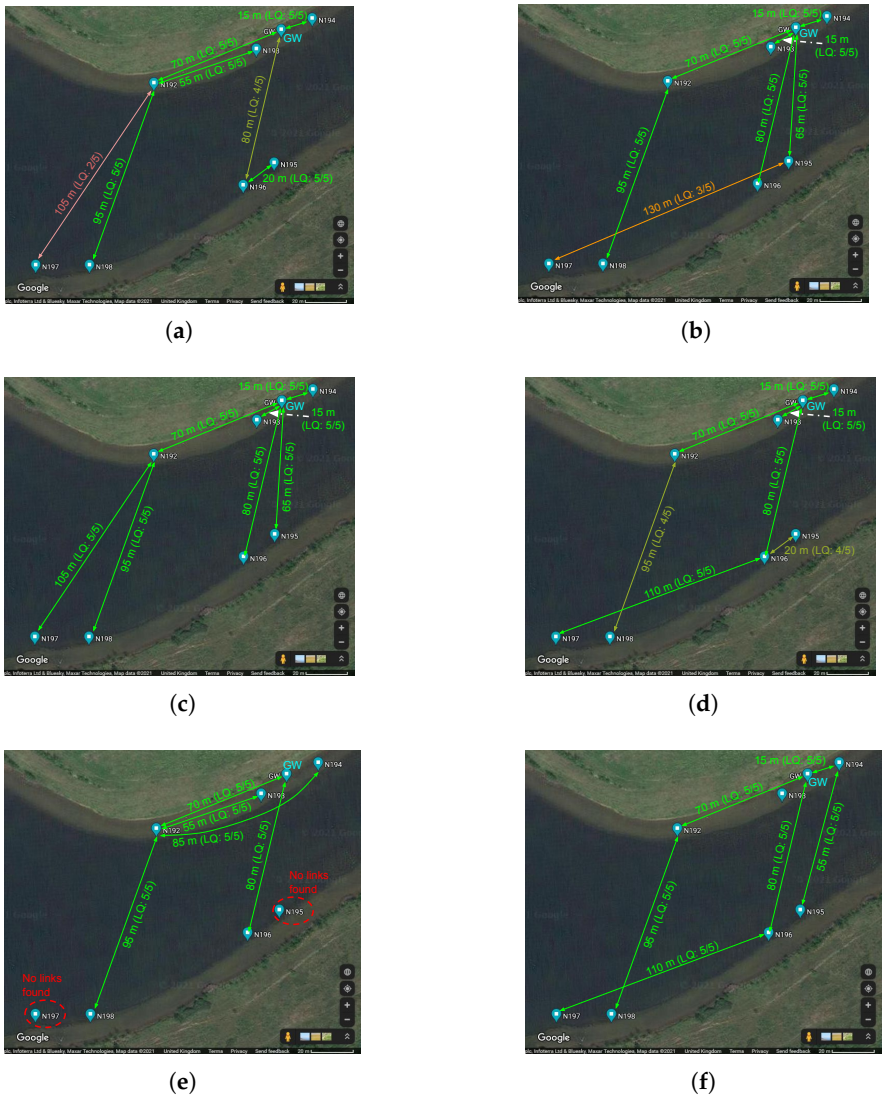


Figure 11. Heslington East lake experiment setup. The nodes were deployed near the shore at shallow depths (1 m depth, with the modem at 0.5 m) on different sides of the lake, with a large amount of vegetation in the middle, resulting in challenging UWA propagation conditions.

The nodes were deployed in the morning (around 10 a.m.) and remained in position throughout the day, while a number of consecutive experiments was conducted, which consisted of a network discovery and setup cycle followed by 50 data gathering cycles, after which a new network discovery phase started, and so on. The idea was to obtain experimental data on the network performance during the network discovery and data gathering phases. The interval between data gathering cycles was set to the minimum possible value, i.e., a new data gathering cycle began as soon as the previous finished. This was done in order to maximize the amount of experimental data on the reliability of the data gathering protocol. The energy efficiency and network longevity via the use of sleep modes was outside of the scope of these trials; those were tested during the long-term sea deployment discussed in Section 4.



(**a**) (**b**)

(**c**) (**d**)

(**e**) (**f**)

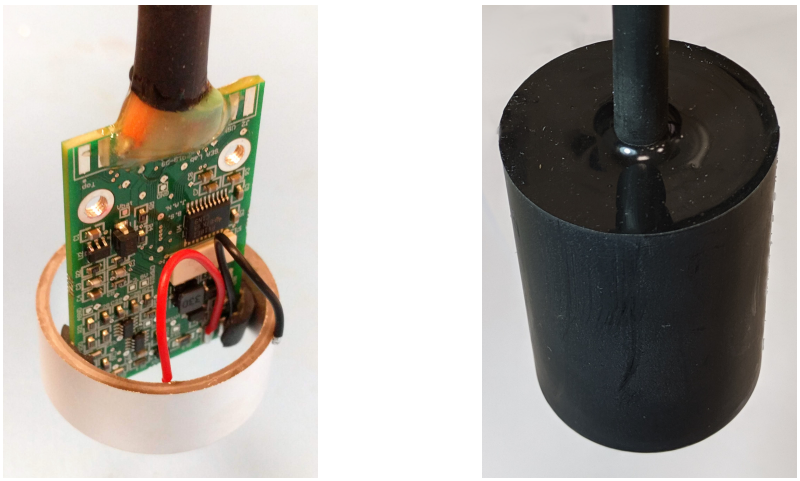
Figure 12. Self-organized network topologies from the Heslington East lake trials, established via the proposed network discovery protocol. (**a**) Topology 1; (**b**) Topology 2; (**c**) Topology 3; (**d**) Topology 4; (**e**) Topology 5 (node outage for N195, N197); (**f**) Topology 6.

3.2. Experiment Hardware

The hardware for the lake trials was developed and built as part of the USMART project [42]; it consisted of 8 NMv3 acoustic modems developed and produced at Newcastle University, seven underwater sensor nodes designed and built at University of York, and one laptop connected to an NMv3 modem acting as the gateway node. This hardware is described below.

3.2.1. NMv3 Acoustic Modems

The NMv3 miniaturized, low-cost, and low-power acoustic modem [47] operates in the 24 kHz to 32 kHz band with receive power consumption of 12.5 mW, transmit power of <1 W, and effective data rates of up to 470 bit/s. Data packets can contain 2–64 user bytes, and can be transmitted as unicast to a specific addressed modem, or as broadcast received by all modems in range. Shorter control packets can be used to ping modems to determine time of flight and hence estimate ranges. The modem assembly consists of a single PCB and transducer as shown in Figure 13a. The potted modem shown in Figure 13b measures 60 mm long by 40 mm diameter.



(**a**) NMv3 Assembly (**b**) NMv3 Potted

Figure 13. NMv3 acoustic modem showing (**a**) the PCB and crystal assembly and (**b**) the final potted modem.

3.2.2. Underwater Sensor Nodes

Figure 14 shows a photo of the underwater sensor node built at University of York for these trials. It comprises:

- BlueRobotics 3“ acrylic enclosure with aluminium endcaps and holes for an external connection with the NMv3 modem, a temperature+pressure sensor and a vent.
- Raspberry Pi Zero W mounted on a custom motherboard (designed at York) and acting as the main CPU of the node.
- 3.7V 10.35Ah Lithium-Ion battery.
- QI wireless charging module including the Microchip MCP73871 charge circuit.
- MAXIM MAX17055 fuel gauge for monitoring the energy consumption, connected to the Raspberry Pi via I^2C.
- NMv3 acoustic modem connected via a 5 m cable and a watertight penetrator to the UART module of the Raspberry Pi.
- BlueRobotics Bar30 external temperature and pressure sensor module (encapsulating the TE MS5837-30BA sensor) connected via I^2C.
- Bosch BNO055 absolute orientation sensor (internal) connected via I^2C.
- ROHM BH1730FVC ambient light sensor (internal) connected via I^2C.
- Bosch BME280 humidity, pressure and temperature sensor (internal) connected via I^2C.

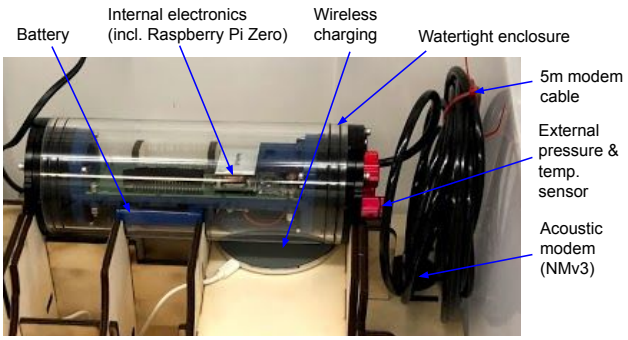


Figure 14. UWA sensor nodes used in the Heslington East lake experiments.

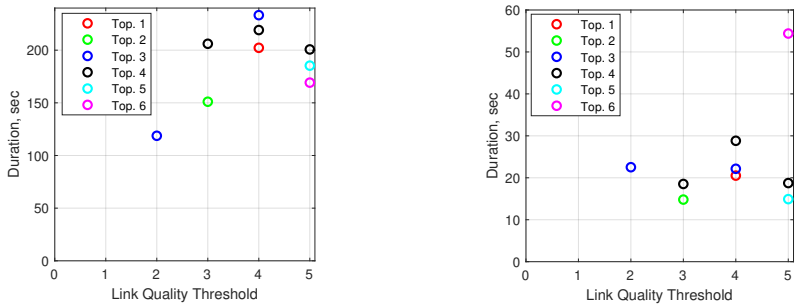
The protocol software was implemented in Python 3, both on the Raspberry Pi based sensor nodes and on the laptop gateway node.

3.3. Network Discovery Performance

First, the performance of the network discovery protocol is analyzed in this section. This is a crucial phase in the overall operation of the network, where a dedicated network discovery procedure is performed in order to establish robust routes from the sensor nodes to the gateway and to initialize the TDA-MAC protocol used in the subsequent data gathering phase.

Figure 12 shows the network topologies that were discovered by our protocol during the Heslington East lake trials on 27 May 2021. It shows the significant variability in connectivity and link quality observed within several hours of this deployment (between 11 A.M. and 4 P.M. GMT). In Topology 1 (Figure 12a), there were no single-hop links with N195 and N193 (located only 15 m away from the gateway) at the link quality threshold $LQT = 4$, whereas in Topology 3 (Figure 12c) the gateway found "perfect" links with those nodes, i.e., with the link quality $Q[1, j] = 5$ (out of 5). There, five sensor nodes were connected directly to the gateway, and two furthest nodes (N197, N198) found "perfect" 5/5 links via N192 as their relay. Topology 3 proved to be the most efficient and reliable topology out of all experiments in these trials. Nevertheless, later in the trials, one of the network discovery cycles (Topology 5, Figure 12e) returned no links to N195 and N197 at all, single-hop or dual-hop. This demonstrates the highly unstable channel conditions encountered in these trials. Nevertheless, our proposed protocol facilitated network adaptability by finding new routes and reconnecting nodes during a periodically repeated network discovery phase.

A total of nine network discovery and setup cycles took place in the Heslington East lake trials, during which the network self-organized into one of the six topologies shown in Figure 12. Figure 15a shows the duration for each of those network discovery cycles. There was no significant correlation observed between the network discovery phase and the LQT, although relaxing the criterion for link selection (lowering the LQT) in principle should result in faster network discovery *all other things being equal*. However, the effect of the variability in the channel conditions from one discovery cycle to the next appears to have far outweighed the impact of varying the LQT on their duration. Overall, the network discovery process took 2–4 min, including the link quality estimation using five test exchanges per link, for a network of seven sensor nodes, which is relatively long, and would not scale well for larger networks. This was due to the challenging propagation conditions, where many acoustic links were absent or too poor, resulting in many long timeouts while nodes were waiting for a ping or test message response from a target node. Nevertheless, these experiments highlight the importance of designing an efficient network discovery protocol, which is a part of the network operation often neglected in UASN protocol research. This issue is addressed in our revised network protocol stack for the North Sea deployment, by streamlining the network discovery process to include a "partial rediscovery" option (see Section 4.1). Furthermore, a key direction of our further work is to design a novel network discovery protocol that is completed in approximately constant time, regardless of the channel conditions.



(**a**) Network discovery (**b**) Network setup

Figure 15. The duration of the (**a**) network discovery and (**b**) network setup stages in the Heslington East lake trials (Top. *n*—topology *n* from Figure 12).

Figure 15b shows that the network setup procedure (distributing the TDI packets and setting up routes) is significantly faster than network discovery (typically between 15–30 s in our experiments); therefore, the focus must be on improving the speed of the latter.

3.4. Data Gathering Performance

Table 1 shows the end-to-end (E2E) packet delivery rates for every node across six different experiments—all of which included 50 data gathering cycles each, using the guard interval of $\tau_g = 300$ ms for TDA-MAC scheduling. The other three experiments (from the previous subsection) focused on the network discovery performance and did not include sufficiently long data gathering phases to provide valid packet delivery rate data.

Table 1. End-to-end packet delivery rates from six consecutive experiments on the Heslington East lake (blue italic text indicates dual-hop connections).

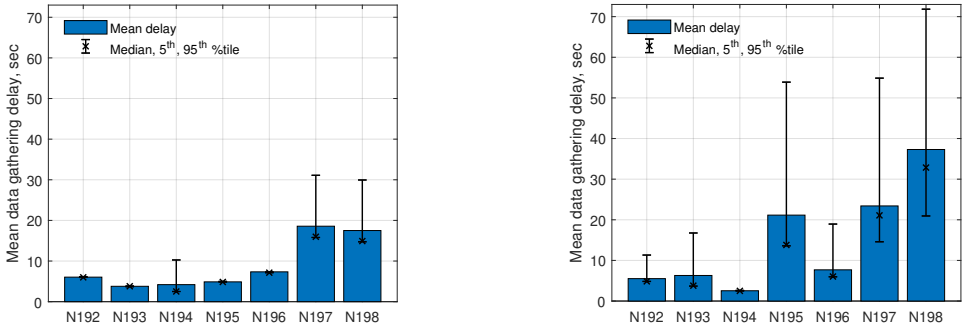
Exp. #	Topology	N192	N193	N194	N195	N196	N197	N198
1	2	0.98	0.98	1.0	0.92	1.0	*0.46*	*0.96*
2	3	1.0	1.0	1.0	1.0	1.0	*0.98*	*1.0*
3	3	1.0	0.96	1.0	0.62	1.0	*0.96*	*0.96*
4	4	1.0	1.0	1.0	*0.94*	0.98	*0.86*	*0.84*
5	4	1.0	0.96	1.0	*0.84*	0.98	*0.86*	*0.92*
6	4	1.0	0.92	1.0	*0.66*	0.86	*0.66*	*0.88*

Four out of seven sensor nodes (N192, N193, N194, N196) always had a direct (single-hop) link with the gateway, and exhibited good performance, achieving E2E packet delivery rates close to 100%, with the exception of Experiment #6, where the channel conditions visibly deteriorated, and N193 & N196 achieving E2E packet delivery rates of 0.92 & 0.86, respectively, which are relatively low considering that the link layer protocol included three transmission attempts to deliver a packet across a given acoustic link. The nodes located furthest from the gateway (N197, N198) were always connected via dual-hop links, whereas N195 switched between single-hop and dual-hop connectivity depending on the channel conditions during a given network discovery cycle.

Overall, the network achieved relatively good E2E packet delivery performance, in most cases 100% or close to it. However, there are also several examples of poor performance for a particular node or pair of nodes, that were investigated further. In Experiment #1, N197 achieved only 0.46 packet delivery rate. It was connected to the gateway via N195 acting as the relay. Inspecting the raw N197-N195 and N195-gateway raw link packet success rates (PSR) revealed that both of those links were poor, with PSR of 0.56 and 0.55, respectively. Both the sensor-relay and the relay-gateway link were therefore unreliable, producing a compound negative effect on the E2E packet success rate. A key issue here was a mismatch between the link performance during the network discovery stage (N195-gateway achieved a perfect LQ score of 5/5), and the link performance during the data gathering stage. In another example (Experiment #6), N195 and N197 both achieved a relatively poor E2E packet delivery rate of 0.66. Inspecting the raw PSR rate, the links from those nodes to their relay (N196) were robust (PSR of 0.92 and 0.96, respectively); however, the link from the relay (N196) to the gateway proved to be the bottleneck, achieving a PSR of 0.63. This example illustrates the importance of selecting robust gateway-relay links, as a poor link can result in performance deterioration for multiple other nodes (the leaf nodes of this relay), despite the sensor-relay links being stable.

Figure 16 shows the E2E delay performance in two example scenarios:

- *Experiment* #2—stable connectivity, good performance.
- *Experiment #6*—poor connectivity for N195 and N197 caused by the relay-gateway (N196-gateway) bottleneck link (as discussed above).



(**a**) Experiment #2 (good performance) (**b**) Experiment #6 (poor performance)

Figure 16. End-to-end packet delivery delays at the data gathering stage. Examples of (**a**) good and (**b**) poor performance.

Figure 16a shows an example of good data gathering performance (Experiment #2, Topology 3), where the single-hop nodes (N192-196) experienced low delays, successfully delivering packets (usually on the first attempt) in the ascending order of their propagation delays—N193, N194, N195, N192, N196—as scheduled by TDA-MAC. The error bar for N194 shows that there was variability for the delivery delay from N194; this is due to a somewhat less reliable channel between N194 and the gateway which sometimes required a retransmission from N194. Nevertheless, the E2E packet delivery rate from N194 was 100% (as shown in Table 1). N197 and N198 experienced significantly longer delays; firstly, because they were connected via dual-hop and, secondly, because our proposed protocol queries the relays to gather the data from their leaf nodes after the gateway gathers the data from the single-hop nodes, i.e., the single-hop nodes are prioritized in the schedule. This is done in order to reduce the overall average E2E delay and increase the proportion of time the single-hop nodes can spend sleeping.

Figure 16b shows an example of poor data gathering performance, where unstable channel conditions resulted in many retransmission attempts (longer delays, higher variability), and especially poor performance for the dual-hop nodes (N195, N197, N198). While N195 and N197 experienced poor connectivity due to the bottleneck relay-gateway (N196-gateway) link, N198 also suffered from it because its transmissions were scheduled (via N192 as relay) after the unreliable N196's branch of the network; therefore, it had to wait for the retransmission attempts and long timeouts due to link outage in that part of the network, before having an opportunity to transmit its data. This demonstrates the scope for further work on dual-hop TDA-MAC scheduling to decrease its sensitivity to link outage and provide an increased fairness to all dual-hop nodes.

4. North Sea Deployment

This section presents the outcomes of a long-term USMART network deployment in the North Sea from 11 October to 26 November 2021.

4.1. Partial Network Discovery Protocol

The results of the Heslington East lake trials discussed in the previous section revealed an issue with the network discovery protocol proposed in this paper: challenging channel conditions and poor node connectivity can significantly extend the duration of network discovery due to timeouts and retries on unresponsive links.

To adapt to changes in channel conditions and node connectivity over the course of a long-term deployment, the network discovery process has to be repeated periodically. At the start of a deployment, when no a priori information about the network topology is available, a full network discovery cycle, as described in Section 2.2, is required. Afterwards, during the network *rediscovery* cycles, prior information about the network can be exploited to streamline this process. To this end, we modified the protocol to include a *partial*

discovery option, aimed at reducing the number of pings and test transmissions required for network rediscovery.

The partial network discovery process reduces the number of link tests using the following three criteria:

1. The stable single-hop and dual-hop links that performed well in the previous data gathering stage are kept (no rediscovery for those nodes). These links are determined by comparing their *packet success rate (PSR)* against a threshold $PSR_{\text{min}}^{\text{good}}$: if $PSR[i,j] \geq PSR_{\text{min}}^{\text{good}}$, the link between nodes i and j is kept for the next data gathering stage, and no rediscovery is required.
2. The links that performed particularly poorly during the last data gathering stage are omitted from the partial network discovery process, as they are likely to continue performing poorly in future. The gateway and relays will skip testing those links, thus saving time and energy. These links are determined by comparing their PSR against a threshold $PSR_{\text{max}}^{\text{poor}}$: if $PSR[i,j] \leq PSR_{\text{max}}^{\text{poor}}$, the link between nodes i and j is skipped in the partial network discovery process.
3. The links that did not produce a successful ping exchange in the previous network discovery cycle are also omitted from the current network discovery cycle, thus saving a potentially significant amount of time, avoiding unresponsive link tests with long timeouts.

The threshold values used in the sea deployment described in this section are: $PSR_{\text{min}}^{\text{good}} = 0.9$, $PSR_{\text{max}}^{\text{poor}} = 0.2$. These values were chosen as a reasonable representation of "good" and "very poor" acoustic communication links; however, the protocol will work with any $PSR_{\text{min}}^{\text{good}}$ and $PSR_{\text{max}}^{\text{poor}}$ values between 0 and 1, providing a full scale of options between doing *no rediscovery* (keeping all links as they are) and doing a full (or almost full) network rediscovery every time.

4.2. Deployment Setup

An 11-node USMART network was deployed ∼3 km off the coast of Blyth in the North Sea in a topology depicted in Figure 17, where the gateway node was located at the edge of the network to provide a more challenging dual-hop connectivity scenario, compared with an optimal gateway placement near the middle which may result in a simple single-hop topology. The sensor nodes were deployed 10 m above the sea bed in water depths of 20 m to 30 m, whereas the gateway was deployed at the surface with the modem at around 7 m depth.

The data gathering interval initially was set to 1 h, with a network discovery cycle scheduled every 6 h, in a recurring pattern of one full network discovery followed by three partial discoveries. It was decided to start the deployment with a high frequency of network discovery cycles in order to capture the variability of the UWA channel conditions between day and night, and also to provide a larger volume of experimental data on network discovery. However, in a practical long-term deployment the ratio between network discovery and data gathering cycles needs to be minimized (primarily to conserve energy) by making the network discovery cycles as infrequent as possible while maintaining reliable performance in the data gathering stages. The sensor nodes and the gateway were deployed on 11 October 2021; with several full system tests and smaller experiments taking place until 2 November, before the main long-term data gathering experiment started. Therefore, the protocol performance data analyzed in this section are from the period of 2–26 November.

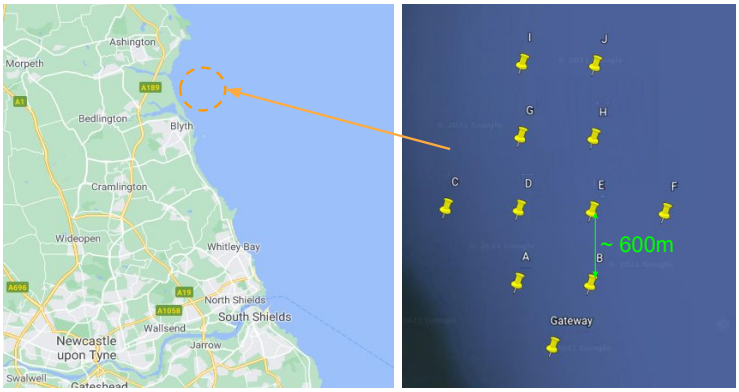


Figure 17. Long-term North Sea deployment setup. Ten sensor nodes and one gateway were deployed off the coast of Blyth in the North of England 10 m above the sea bed in 20 m to 30 m water depths. The gateway node was located on the edge of the network to provide longer end-to-end dual-hop links with the sensor nodes.

4.3. USMART Network Hardware

A number of bespoke USMART sensor nodes were designed and built for the purpose of the field trials as shown in Figure 18. Aimed at offshore experiments the nodes needed to withstand water depths of tens of meters in a range of sea and weather conditions. Off-the-shelf diver canisters (depth rated to 180 m) were used to house the electronics, with the modem cable potted into place using epoxy and polyurethane to seal against water ingress. For extended duration deployments, low-power sleep states and fine grained control of power to subsystems was key in both the selection of the MicroPython processor board, as well as the design of the bespoke expansion PCB for power management and sensors. The sensor node consists of:

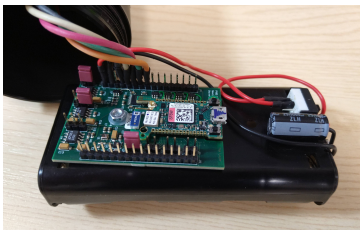
- Diver Canister;
- NMv3 Acoustic Modem [47];
- Expansion PCB - Power Regulators, Switches, Sensors (e-compass, temperature, pressure, humidity);
- MicroPython Pyboard D-series (PYBD) with STM32F767 and WiFi/BT SF6W;
- Four Alkaline C Cells.

The power consumption of the sensor node was measured in various configurable states as described here. MicroPython PYBD and Expansion Board with 5 V supply:– Light Sleep (with RTC and HW interrupts) 1.46 mA (7.3 mW); Active (Running in loop) 79 mA (395 mW). NMv3 Power Consumption [47]:– Off 0 W; Listening 12.5 mW; Transmitting 1.5 W.

When the sensor node was listening for incoming acoustic transmissions from the gateway, the NMv3 was in listening mode and the MicroPython PYBD was in light sleep mode, able to be brought quickly into the active state when the flag line from the NMv3 indicated an incoming packet. A combined power consumption of 7.3 mW + 12.5 mW = 19.8 mW. When the sensor mode was sleeping between transmission windows, the NMv3 was powered off leaving only the MicroPython PYBD in light sleep mode (7.3 mW).

With four C alkaline cells and a total energy capacity of 40 W h the sensor node could spend up to 2020 h (84 days) just listening, or 5479 h (228 days) in light sleep with no modem powered. Time when the MicroPython PYBD is active, and time when the modem is transmitting all take time and energy from these upper limits.

(a) Open Canister



(b) Electronics

(c) Assembled Sensor Node

Figure 18. USMART Sensor Node. (**a**) Opened canister showing the electronics and battery holder. (**b**) Close up of the electronics mounted on the battery holder. Showing the custom power management and sensors PCB with the MicroPython PYBD connected. (**c**) Assembled node with 0.5 m cable to the acoustic modem.

Extensive testing took place prior to sea trials with the sensor nodes distributed around the anechoic test tank as shown in Figure 19. This enabled the network protocols and software to be run over an extended duration to identify any potential problems ahead of deployment at sea. Placing one of the modems outside the tank and acoustically coupling it to the fibreglass wall with some water provided a much weaker acoustic channel between this node and the nearest node within the tank itself. This allowed a multi-hop topology to be forced during testing.

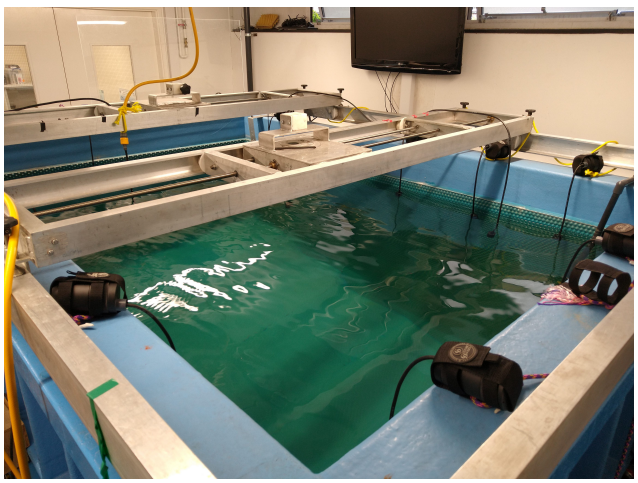


Figure 19. Sensor Nodes positioned around the edge of the anechoic test tank in SEA Lab at Newcastle University. Extensive development and testing of network protocols over long duration runs in the tank environment allowed issues to be resolved prior to sea trials.

Deployment at sea required mooring configurations as shown in Figure 20. The gateway node placed at the surface had a WiFi antenna mounted on the wooden pole with the acoustic modem some 7 m below the sea surface. Weights attached to the rope, to which the modem was also fastened, kept it pulled downwards to counter the tidal flow. The sensor nodes deployed on the sea bed used local floats to keep the modem pointing

upwards, with a separate rope from the anchor and buoy at the surface to act as markers for retrieval. By separating the local float from the surface buoy the effect of tidal surge on the vertical orientation of the sensor node modem was greatly reduced.

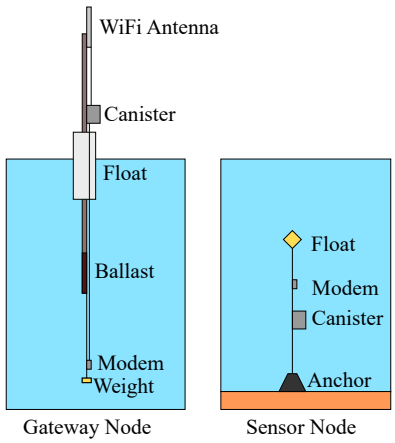
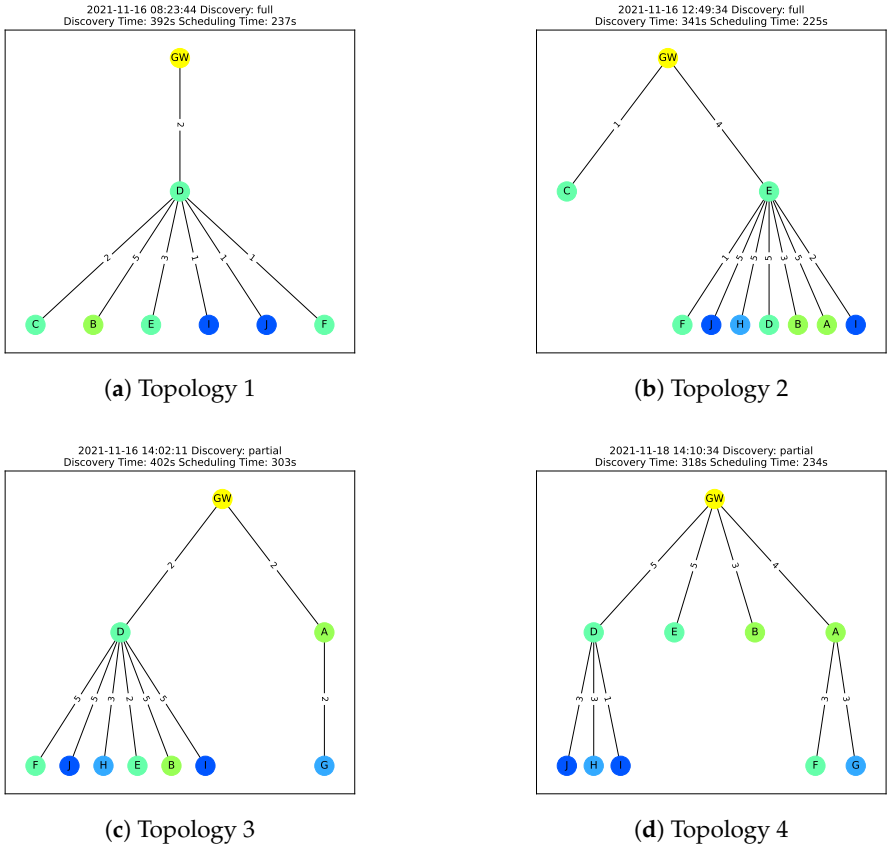


Figure 20. The moorings for the gateway node and the sensor node showing the use of weights and floats to maintain modem orientation.

4.4. Network Discovery Performance

Figure 21 shows several examples of the network topologies established by the network discovery protocol in the North Sea deployment. They show that most sensor nodes were connected to the gateway via dual-hop links, with few direct links between the gateway and sensor nodes. This is a challenging scenario for the network discovery protocol, since establishing direct links with the gateway is significantly faster than searching for dual-hop links via relays, and since the probability of encountering unresponsive links increases with the number of target dual-hop nodes, which further extends the duration of the network discovery.

Figure 22 shows the cumulative distribution functions (CDFs) of the network discovery and setup phase duration throughput the sea deployment. Firstly, there was no significant benefit observed in employing the partial network discovery protocol, compared with the original full version. This is because of the overall poor connectivity of the nodes (discussed in the next Section), especially the gateway, which precluded the network from exploiting stable links based on the data gathering performance, as a result having to search for new links most of the time (regardless of whether this was a full or partial network discovery cycle). Secondly, the average duration of a network discovery cycle was 5 min 51 s, and in some cases exceeding 10 min. This highlighted the major drawback of the contention-free, sequential network discovery protocol proposed in this paper: the duration of a network discovery cycle is significantly affected by the UWA environment, in particular, by the node connectivity and quality of the links. An alternative solution for the network discovery process with lower complexity than $\mathcal{O}(N^2)$, which scales with the number of nodes and is not affected by the channel conditions, is the key direction of further work. For example, an opportunistic network discovery protocol based on a fixed length time window for exchanging pings and broadcasting beacons for channel estimation is a promising approach, particularly because of its $\mathcal{O}(1)$ complexity, which would make it scalable to large numbers of nodes.



(**a**) Topology 1 (**b**) Topology 2

(**c**) Topology 3 (**d**) Topology 4

Figure 21. Examples of self-organized network topologies in the North Sea deployment. Each edge displays the Link Quality [integer from 1 to 5] as measured during the discovery process.

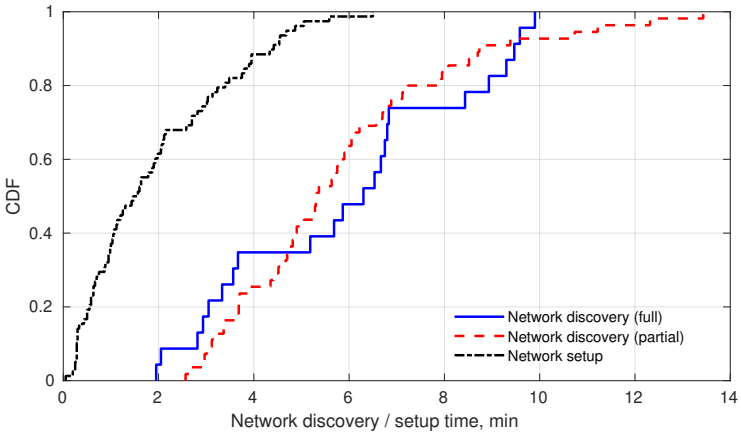
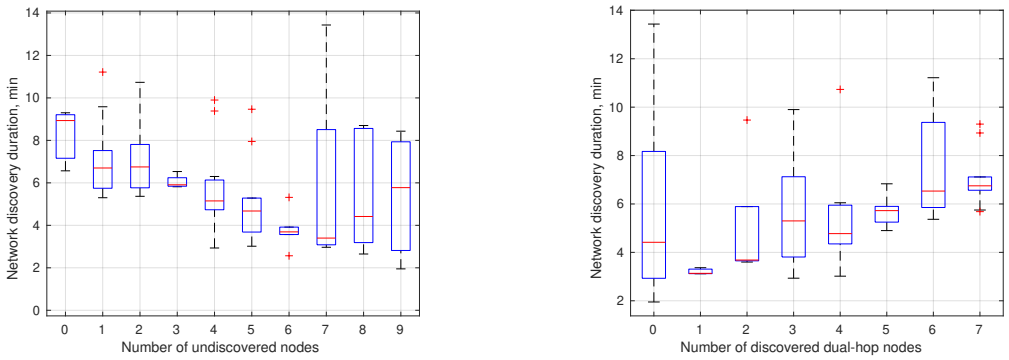


Figure 22. Challenging channel conditions and poor connectivity in the North Sea deployment resulted in long network discovery phases (6 min on average). The average network setup time was approx. 2 min, significantly longer than in Heslington East lake trials.

Figure 22 also shows that the network setup process (i.e., TDI distribution to set up the MAC and routing protocol for data gathering) is significantly shorter than network discovery, but still took ∼2 min on average—much longer than in the Heslington East lake trials. It was also severely affected by the poor channel conditions, high packet loss and unresponsive links. Streamlining the network setup stage is another direction of further work; though not as critical as the development of a new network discovery protocol.

Figure 23 shows how the duration of the network discovery cycles is affected by the number of undiscovered sensor nodes (nodes for which no single- or dual-hop links

were found), and by the number of discovered dual-hop nodes. Interestingly, the network discovery duration statistically decreases with the number of undiscovered nodes (from 0 to 6), showing that larger network topologies took longer to discover, despite having fewer unresponsive links and ping timeouts. This is because the quality of the links obtained via the link testing process described in Section 2.2 was often below the link quality threshold ($LQT = 4$ for most of the deployment), thus requiring further dual-hop network discovery and link testing. The variability in the network discovery duration is significantly larger in cases where most nodes were undiscovered, i.e., the majority of the network discovery process was spent waiting for ping and test message timeouts. Figure 23b shows that the duration of the network discovery process statistically increases with the number of dual-hop nodes, which is an expected outcome—finding good dual-hop links is significantly more time-consuming than single-hop links, especially if there are several relay candidates to iterate through.



(**a**) (**b**)

Figure 23. The duration of the network discovery cycles in the North Sea deployment is loosely correlated with the number of dual-hop nodes in the discovered topology, and with the number of undiscovered nodes (i.e., nodes that the gateway failed to connect with). (**a**) Network discovery duration vs the number of undiscovered nodes; (**b**) Network discovery duration vs. the number of discovered dual-hop nodes.

A potential vulnerability of the proposed network discovery protocol, which has possibly manifested in the sea deployment, is the "cascade effect" of lost ACKs for network discovery requests. In the worst case scenario, a relay node receives the network discovery request from the gateway, transmits back an ACK to the gateway (which gets lost), and starts the link testing procedure (exchanging pings and test messages with the target nodes). Meanwhile, the gateway does not receive the ACK, tries to resend the request to the relay several more times, fails (because the relay's modem is busy), and then asks the next relay to perform network discovery. This could result in two (or possibly more) network discovery subroutines launched by multiple relays in parallel which would interfere with each other. Increasing the protocol robustness against ACK losses is an issue often neglected in the UAN literature, but it is a key direction of further work as it can result in significant disruption to the protocol flow, as described in an example above.

4.5. Network Connectivity Statistics

Figure 24 summarizes the challenging network connectivity encountered in the North Sea deployment. It shows that in over 60% of cases no more than two sensor nodes were connected directly to the gateway, and in ∼95% of cases—three or less out of ten. This suggests that a key issue was poor gateway connectivity. A likely cause for this was weak acoustic connectivity at the PHY layer (e.g., the modem rising close to the sea surface and/or being partially obstructed by the rope/canister or other objects); however, it could also be caused by a range of possible hardware/software implementation issues or protocol vulnerabilities that were not observed in preceding water tank tests. The black solid curve in Figure 24 shows

that only in ∼3% of cases all 10 sensor nodes were discovered by the gateway (the original aim was to keep it close to 100%), and in ∼50% of network discovery cycles at most half of the nodes found a connection to the gateway. These statistics highlight the overall challenging conditions of the communication environment experienced in this deployment.

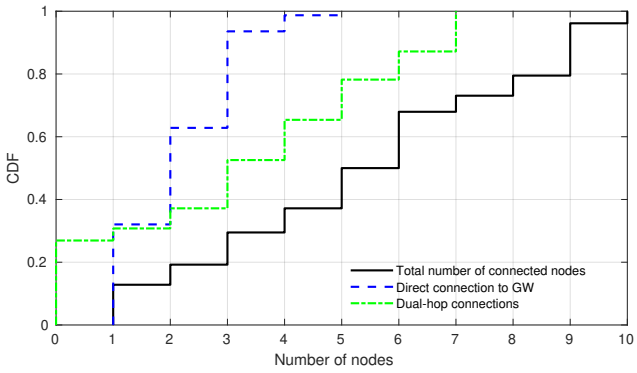


Figure 24. Overall connectivity statistics from the North Sea Deployment show that many nodes failed to find any links to the gateway, single-hop or dual-hop.

Figure 25 highlights the issue of weak gateway connectivity; it shows the distribution of the average link quality in each network discovery cycle, separating the first hop (gateway—sensor node) and second hop (relay sensor node—leaf sensor node) links. The CDFs show a significant difference in the link quality between the first and second hop links. Those distributions were expected to tend towards the same curve over a large statistical sample (hundreds of links), as both the sensor nodes and the gateway used identical hardware.

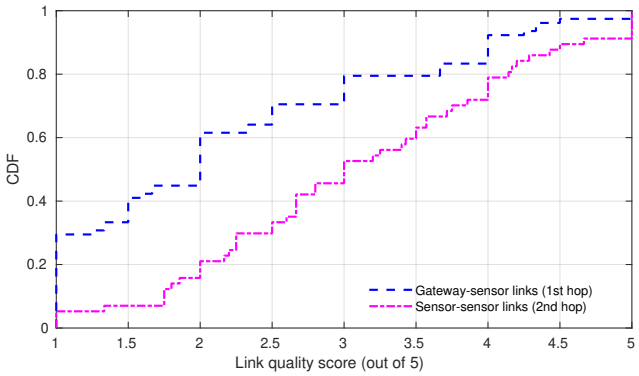


Figure 25. Overall link quality statistics from the North Sea Deployment. When the sensor nodes did manage to find single- or dual-hop links to the gateway, the link quality on the best links was often poor. Notably, the quality of the gateway-sensor links was visibly poorer than than of sensor-sensor (second hop) links; this corroborates the conclusion that the gateway placement (and therefore poor gateway connectivity) was likely the cause of the poor overall network performance.

4.6. Data Gathering Performance

Figure 26 shows the packet delivery timeline, where every blue dot represents a successful packet delivery from the corresponding node. The graph shows long periods of node outage, where some nodes were unable to connect to the gateway via the network discovery protocol. It also shows that, when the nodes did manage to connect to the network, the packet delivery performance was patchy, with many data packets failing to be delivered. at 13:15 on 18 November 2021, the network was configured to run in the data gathering mode indefinitely (with no new network discoveries), and with the data gathering frequency increased to once every 10 min. Eight out of ten sensor nodes were connected in this phase, and their data gathering performance improved, owing to a relatively good topology discovered by the protocol and an improvement in the channel conditions.

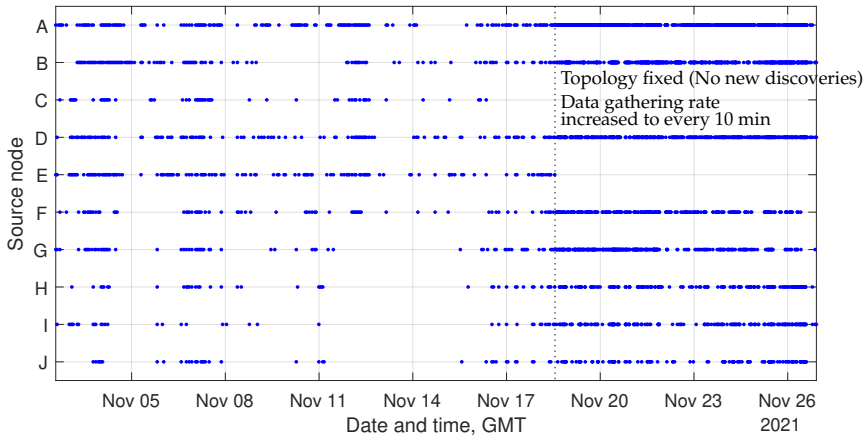


Figure 26. End-to-end packet delivery timeline from the North Sea deployment. It shows long periods of node outage, when the gateway could not connect to several nodes.

Figure 27 shows the average packet delivery ratio (PDR) for every node referenced A–J (corresponding to the annotated deployment map in Figure 17). The overall network performance was poor, with the closest node (A) managing to deliver just over 50% of its packets to the gateway, while other nodes achieved lower PDR; particularly node E, which was likely due to particularly weak acoustic connectivity. The PDR was calculated as follows:

$$PDR = \frac{N_{\text{rx packets}}[n]}{N_{\text{dgc}}[n]}, \tag{14}$$

where $N_{\text{rx packets}}[n]$ is the total number of data packets gathered from node n, and $N_{\text{dgc}}[n]$ is the total number of data gathering cycles where node n was connected to the network. Therefore, the PDR is defined the proportion of data gathering cycles where both the REQ packet from the gateway was delivered to the sensor node (via single- or dual-hop) and the data packet was delivered back to the gateway. If any packets were lost along this chain of transmissions (and were not resolved via ARQ), this resulted in a failed packet delivery, with high packet loss rates on many acoustic links producing a compound negative effect on the PDR.

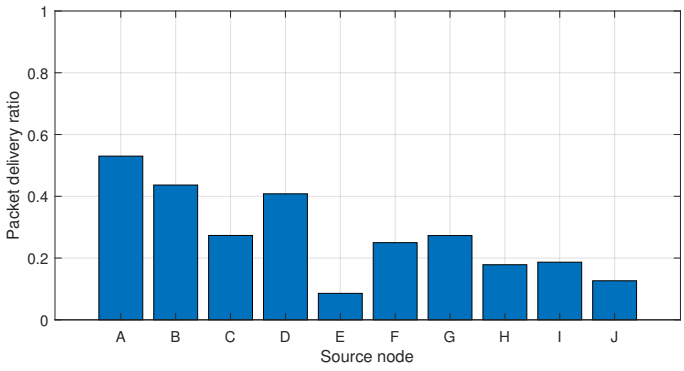


Figure 27. End-to-end packet delivery ratio from the North Sea deployment. It shows the overall poor performance for many nodes due to excessive link outage.

4.7. Long-Term Energy Efficiency Performance

Figure 28 plots the real-time battery voltage data gathered from each sensor node throughout the duration of the sea deployment. It shows that the battery voltage decreased at approximately the same rate during the first month for all sensor nodes, except Node A which was using more energy as it was selected by the gateway as a relay node significantly more often than other nodes (due to limited network connectivity discussed earlier in this section). This difference in the energy consumption is especially visible in the last phase

of the deployment (18–26 November), where the topology was fixed, the data gathering interval was reduced from 1 h to 10 min, and Node A was relaying data for three other sensor nodes (see the network topology in Figure 21d).

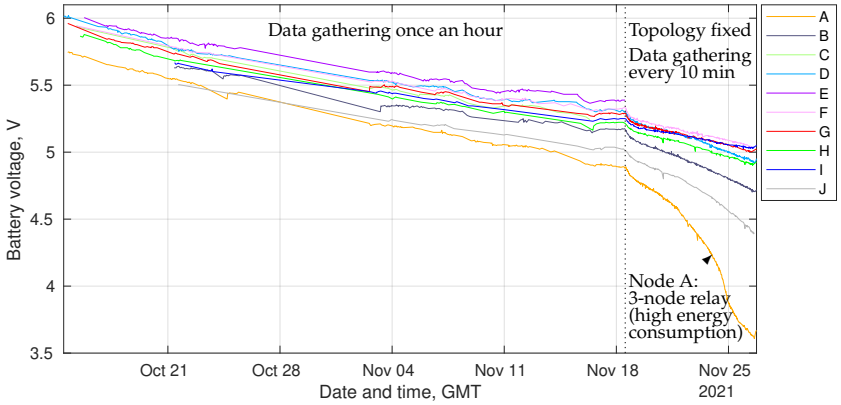


Figure 28. Sensor node battery voltage timeline throughout the North Sea deployment.

These data show that the hardware design of the USMART sensor nodes and the NMv3 modems was successful, with all nodes remaining operational for approximately 1.5 months, despite long periods of node outage (see Figure 26) which prevented them from exploiting low power sleep modes; instead, they remained in the listening mode with the main CPU and the modem power on. We are confident that these nodes are capable of supporting long-term data gathering missions at sea lasting several months, with better gateway connectivity and placement and a more robust network discovery protocol.

5. Conclusions and Further Work

A complete cross-layer networking solution for data gathering in UASNs was proposed in this paper, including a network discovery protocol, intelligent routing with relay load balancing, scheduling via TDA-MAC, multi-node ARQ, and sleep mode management. The protocol stack was implemented in hardware and tested in lake trials on the University of York Heslington East campus lake. The trials demonstrated the network's capability to self-organize into effective dual-hop topologies for gathering the data from 7 underwater sensor nodes at a designated master node. However, in some network discovery cycles several nodes suffered from highly variable connectivity and link quality, resulting in less reliable network topologies for subsequent data gathering. Furthermore, the network discovery process lasted between 2–4 min due to a large number of unresponsive links causing long timeouts when measuring their quality via ping and test message exchanges. To address this issue, an extension of the network discovery protocol was designed that exploits past information about the performance of particular links and helps the gateway streamline the discovery process, and thus repeat it more frequently if necessary.

A USMART network prototype, comprising 10 sensor nodes and one surface gateway, was developed and implemented in hardware, with the aim of supporting long-term data gathering missions at sea. It was deployed in the North Sea on 11 October 2021, and remained operational until 26 November 2021, despite long periods of node outage caused by poor network connectivity which prevented the sensor nodes from fully exploiting low power sleep modes. This demonstrates that the hardware design of the USMART network prototype is successful and capable of supporting long-term data gathering missions in future, with potential to further significantly extend their battery lifetime by ensuring robust network connectivity, e.g., with more optimal gateway placement and a more robust network discovery protocol.

The development of an improved network discovery protocol is the key direction of further work. The current contention-free, sequential protocol performs well in stable channel conditions with good network connectivity, but its performance deteriorates significantly with poorer node connectivity and time-varying link quality. An alternative

solution based on a more opportunistic network discovery process is needed, which is scalable for large networks and is not affected by the channel conditions.

Author Contributions: Conceptualization, J.A.N. and P.D.M.; Data curation, B.S.; Formal analysis, N.M.; Funding acquisition, J.A.N., P.D.M. and Y.Z.; Investigation, N.M., B.S., J.A.N., P.D.M. and Y.Z.; Methodology, N.M., B.S. and J.A.N.; Project administration, J.A.N. and P.D.M.; Resources, B.T.H.; Software, N.M., B.S. and B.T.H.; Visualization, N.M. and B.S.; Writing – original draft, N.M. and B.S.; Writing – review & editing, B.T.H., J.A.M, P.D.M. and Y.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the UK Engineering and Physical Sciences Research Council (EPSRC) through the USMART (EP/P017975/1), Full-Duplex (EP/R003297/1) and COUSIN (EP/V009591/1) projects.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data from the lake and sea trials reported in this paper, together with the MATLAB code for processing it, will be made publicly available on the York Research Database at https://pure.york.ac.uk/portal (accessed on 5 August 2022).

Conflicts of Interest: The authors declare no conflict of interest..

References

1. Dol, H.S.; Casari, P.; van der Zwan, T.; Otnes, R. Software-Defined Underwater Acoustic Modems: Historical Review and the NILUS Approach. *IEEE J. Ocean. Eng.* **2017**, *42*, 722–737. [CrossRef]
2. Demirors, E.; Sklivanitis, G.; Santagati, G.E.; Melodia, T.; Batalama, S.N. A High-Rate Software-Defined Underwater Acoustic Modem With Real-Time Adaptation Capabilities. *IEEE Access* **2018**, *6*, 18602–18615. [CrossRef]
3. Renner, C.; Golkowski, A. Acoustic Modem for Micro AUVs: Design and Practical Evaluation. In Proceedings of the WUWNet'16: Proceedings of the 11th ACM International Conference on Underwater Networks & Systems, Shanghai, China, 24–26 October 2016; pp. 2:1–2:8.
4. Sherlock, B.; Tsimenidis, C.C.; Neasham, J.A. Signal and receiver design for low-power acoustic communications using m-ary orthogonal code keying. In Proceedings of the IEEE OCEAN, Genova, Italy, 18–21 May 2015. [CrossRef]
5. Cario, G.; Casavola, A.; Gjanci, P.; Lupia, M.; Petrioli, C.; Spaccini, D. Long lasting underwater wireless sensors network for water quality monitoring in fish farms. In Proceedings of the IEEE OCEANS'17, Aberdeen, UK, 19–22 June 2017; pp. 1–6.
6. Mohapatra, A.K.; Gautam, N.; Gibson, R.L. Combined Routing and Node Replacement in Energy-Efficient Underwater Sensor Networks for Seismic Monitoring. *IEEE J. Ocean. Eng.* **2013**, *38*, 80–90. [CrossRef]
7. Boom, B.; He, J.; Palazzo, S.; Huang, P.; Beyan, C.; Chou, H.M.; Lin, F.P.; Spampinato, C.; Fisher, R. A research tool for long-term and continuous analysis of fish assemblage in coral-reefs using underwater camera footage. *Ecol. Inform.* **2014**, *23*, 83–97. [CrossRef]
8. Ali, S.; Ashraf, A.; Qaisar, S.B.; Afridi, M.K.; Saeed, H.; Rashid, S.; Felemban, E.A.; Sheikh, A.A. SimpliMote: A Wireless Sensor Network Monitoring Platform for Oil and Gas Pipelines. *IEEE Syst. J.* **2018**, *12*, 778–789. [CrossRef]
9. Palisetti, S.; Chandavarkar, B.R.; Gadagkar, A.V. Intrusion Detection of Sinkhole Attack in Underwater Acoustic Sensor Networks. In Proceedings of the 12th International Conference on Computing Communication and Networking Technologies ICCCNT'21, Kharagpur, India, 6–8 July 2021; pp. 1–7. [CrossRef]
10. Lanbo, L.; Shengli, Z.; Jun-Hong, C. Prospects and problems of wireless communication for underwater sensor networks. *Wirel. Commun. Mob. Comput.* **2008**, *8*, 977–994. [CrossRef]
11. Heidemann, J.; Stojanovic, M.; Zorzi, M. Underwater sensor networks: Applications, advances and challenges. *Philos. Trans. R. Soc. A* **2012**, *370*, 158–175. [CrossRef]
12. Jiang, S. State-of-the-Art Medium Access Control (MAC) Protocols for Underwater Acoustic Networks: A Survey Based on a MAC Reference Model. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 96–131. [CrossRef]
13. Erol-Kantarci, M.; Mouftah, H.T.; Oktug, S. A Survey of Architectures and Localization Techniques for Underwater Acoustic Sensor Networks. *IEEE Commun. Surv. Tutor.* **2011**, *13*, 487–502. [CrossRef]
14. Chen, K.; Ma, M.; Cheng, E.; Yuan, F.; Su, W. A Survey on MAC Protocols for Underwater Wireless Sensor Networks. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 1433–1447. [CrossRef]
15. Chirdchoo, N.; Soh, W.; Chua, K. Aloha-Based MAC Protocols with Collision Avoidance for Underwater Acoustic Networks. In Proceedings of the 26th IEEE International Conference on Computer Communications INFOCOM'07, Anchorage, AK, USA, 6–12 May 2007; pp. 2271–2275. [CrossRef]
16. Liao, W.H.; Huang, C.C. SF-MAC: A Spatially Fair MAC Protocol for Underwater Acoustic Sensor Networks. *IEEE Sens. J.* **2012**, *12*, 1686–1694. [CrossRef]

17. Diamant, R.; Casari, P.; Campagnaro, F.; Zorzi, M. A Handshake-Based Protocol Exploiting the Near-Far Effect in Underwater Acoustic Networks. *IEEE Wirel. Commun. Lett.* **2016**, *5*, 308–311. [CrossRef]
18. Rahman, M.; Lee, Y.; Koo, I. An adaptive network allocation vector timer-based carrier sense multiple access with collision avoidance medium access control protocol for underwater acoustic sensor networks. *Int. J. Distrib. Sens. Netw.* **2017**, *13*. [CrossRef]
19. Molins, M.; Stojanovic, M. Slotted FAMA: A MAC protocol for underwater acoustic networks. In Proceedings of the IEEE OCEANS, Singapore, 16–19 May 2006.
20. Noh, Y.; Lee, U.; Han, S.; Wang, P.; Torres, D.; Kim, J.; Gerla, M. DOTS: A Propagation Delay-Aware Opportunistic MAC Protocol for Mobile Underwater Networks. *IEEE Trans. Mob. Comput.* **2014**, *13*, 766–782. [CrossRef]
21. Anjangi, P.; Chitre, M. Propagation-Delay-Aware Unslotted Schedules with Variable Packet Duration for Underwater Acoustic Networks. *IEEE J. Ocean. Eng.* **2017**, *42*, 977–993. [CrossRef]
22. Diamant, R.; Lampe, L. Spatial Reuse Time-Division Multiple Access for Broadcast Ad Hoc Underwater Acoustic Communication Networks. *IEEE J. Ocean. Eng.* **2011**, *36*, 172–185. [CrossRef]
23. Lmai, S.; Chitre, M.; Laot, C.; Houcke, S. Throughput-efficient super-TDMA MAC transmission schedules in ad hoc linear underwater acoustic networks. *IEEE J. Ocean. Eng.* **2017**, *42*, 156–174. [CrossRef]
24. Lmai, S.; Chitre, M.; Laot, C.; Houcke, S. Throughput-Maximizing Transmission Schedules for Underwater Acoustic Multihop Grid Networks. *IEEE J. Ocean. Eng.* **2015**, *40*, 853–863. [CrossRef]
25. Kredo, K., II; Djukic, P.; Mohapatra, P. STUMP: Exploiting Position Diversity in the Staggered TDMA Underwater MAC Protocol. In Proceedings of the IEEE INFOCOM, Rio de Janeiro, Brazil, 19–25 April 2009.
26. Luo, J.; Chen, Y.; Wu, M.; Yang, Y. A Survey of Routing Protocols for Underwater Wireless Sensor Networks. *IEEE Commun. Surv. Tutor.* **2021**, *23*, 137–160. [CrossRef]
27. Islam, T.; Lee, Y.K. A Comprehensive Survey of Recent Routing Protocols for Underwater Acoustic Sensor Networks. *Sensors* **2019**, *19*, 4256. [CrossRef]
28. Bhattacharyya, D.; Kim, T.; Pal, S. A Comparative Study of Wireless Sensor Networks and Their Routing Protocols. *Sensors* **2010**, *10*, 10506–10523. [CrossRef] [PubMed]
29. Shin, D.; Hwang, D.; Kim, D. DFR: An efficient directional flooding-based routing protocol in underwater sensor networks. *Wirel. Commun. Mob. Comput.* **2012**, *12*, 1517–1527. [CrossRef]
30. Ayaz, M.; Abdullah, A.; Faye, I.; Batira, Y. An efficient Dynamic Addressing based routing protocol for Underwater Wireless Sensor Networks. *Comput. Commun.* **2012**, *35*, 475–486. [CrossRef]
31. Jornet, J.M.; Stojanovic, M.; Zorzi, M. Focused Beam Routing Protocol for Underwater Acoustic Networks. In *Proceedings of the 3rd ACM International Workshop on Underwater Networks, San Francisco, CA, USA 15 September 2008*; Association for Computing Machinery: New York, NY, USA, 2008; pp. 75–82. [CrossRef]
32. Carlson, E.A.; Beaujean, P.P.; An, E. Location-Aware Routing Protocol for Underwater Acoustic Networks. In Proceedings of the OCEANS 2006, Boston, MA, USA, 18–21 September 2006; pp. 1–6. [CrossRef]
33. Yan, H.; Shi, Z.J.; Cui, J. DBR: Depth-Based Routing for Underwater Sensor Networks. In *Proceedings of the NETWORKING 2008 Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet, Singapore, 5–9 May 2008*; Das, A., Pung, H.K., Lee, F.B.S., Wong, L.W.C., Eds.; Springer: Berlin/Heidelberg, Germany, 2008; pp. 72–86.
34. Khan, T.; Ahmad, I.; Aman, W.; Azam, I.; Khan, Z.A.; Qasim, U.; Avais, S.; Javaid, N. Clustering Depth Based Routing for Underwater Wireless Sensor Networks. In Proceedings of the 30th IEEE International Conference on Advanced Information Networking and Applications (AINA), Crans-Montana, Switzerland, 23–25 March 2016; pp. 506–515. [CrossRef]
35. Ronai, M.A.; Kail, E. A simple neighbour discovery procedure for Bluetooth ad hoc networks. In Proceedings of the IEEE GLOBECOM'03, San Francisco, CA, USA, 1–5 December 2003; Volume 2, pp. 1028–1032.
36. Ye, W.; Heidemann, J.; Estrin, D. Medium Access Control with Coordinated Adaptive Sleeping for Wireless Sensor Networks. *IEEE/ACM Trans. Netw.* **2004**, *12*, 493–506. [CrossRef]
37. Orlinski, M.; Filer, N. Neighbour discovery in opportunistic networks. *Ad Hoc Netw.* **2015**, *25*, 383–392. [CrossRef]
38. Othman, A.K.; Adams, A.E.; Tsimenidis, C.C. Node Discovery Protocol and Localization for Distributed Underwater Acoustic Networks. In Proceedings of the AICT-ICIW'06, Guadeloupe, French Caribbean, 19–25 February 2006; pp. 1–6.
39. Watfa, M.; Selman, S.; Denkilkian, H. UW-MAC: An underwater sensor network MAC protocol. *Int. J. Commun. Syst.* **2010**, *23*, 485–506. [CrossRef]
40. Petroccia, R. A distributed ID assignment and topology discovery protocol for underwater acoustic networks. In Proceedings of the IEEE UComms'16, Lerici, Italy, 30 August–1 September 2016; pp. 1–5.
41. Zhuo, X.; Qu, F.; Yang, H.; Wei, Y.; Wu, Y.; Li, J. Delay and Queue Aware Adaptive Scheduling-Based MAC Protocol for Underwater Acoustic Sensor Networks. *IEEE Access* **2019**, *7*, 56263–56275. [CrossRef]
42. EPSRC USMART (EP/P017975/1). 2018. Available online: https://research.ncl.ac.uk/usmart/ (accessed on 6 August 2022).
43. Morozs, N.; Mitchell, P.; Zakharov, Y. TDA-MAC: TDMA Without Clock Synchronization in Underwater Acoustic Networks. *IEEE Access* **2018**, *6*, 1091–1108. [CrossRef]
44. Morozs, N.; Mitchell, P.; Zakharov, Y.; Mourya, R.; Petillot, Y.; Gibney, T.; Dragone, M.; Sherlock, B.; Neasham, J.; Tsimenidis, C.; et al. Robust TDA-MAC for Practical Underwater Sensor Network Deployment: Lessons from USMART Sea Trials. In Proceedings of the ACM WUWNet'18, Shenzhen, China, 3–5 December 2018.

45. Morozs, N.; Mitchell, P.; Zakharov, Y. Dual-Hop TDA-MAC and Routing for Underwater Acoustic Sensor Networks. *IEEE J. Ocean. Eng.* **2019**, *44*, 865–880. [CrossRef]
46. Morozs, N.; Mitchell, P.D.; Zakharov, Y. Routing Strategies for Dual-Hop TDA-MAC: Trade-Off Between Network Throughput and Reliability. In Proceedings of the UACE'19, Crete, Greece, 30 June–5 July 2019.
47. Sherlock, B.; Morozs, N.; Neasham, J.; Mitchell, P. Ultra-Low-Cost and Ultra-Low-Power, Miniature Acoustic Modems Using Multipath Tolerant Spread-Spectrum Techniques. *Electronics* **2022**, *11*, 1446. [CrossRef]