Linear TDA-MAC: Unsynchronized Scheduling in Linear Underwater Acoustic Sensor Networks

Nils Morozs, *Member, IEEE*, Paul D. Mitchell, *Senior Member, IEEE*, and Yuriy Zakharov, *Senior Member, IEEE*

***Abstract*—Underwater acoustic sensor networks (UASNs) are a key enabling technology for live monitoring of subsea assets. This often involves networks with line topologies, e.g. sensor nodes attached to oil & gas pipelines. In this paper we propose Linear Transmit Delay Allocation MAC (LTDA-MAC) for efficient packet scheduling in linear UASNs without clock synchronization at the sensor nodes. It is achieved via online heuristic optimization that produces schedules tailored to a given deployment scenario. Simulations of a subsea pipeline monitoring use case show that LTDA-MAC significantly outperforms Spatial-TDMA in networks with long propagation delays.**

***Keywords*—Linear Multi-Hop Network, Medium Access Control, TDA-MAC, Underwater Acoustic Network**

I. Introduction

The use of underwater acoustic sensor networks (UASNs) for monitoring the underwater environment is becoming an increasingly popular research subject enabled by the recent developments in underwater acoustic modem technologies [1]. Acoustic waves are the preferred practical medium for underwater communications, since they exhibit significantly better propagation characteristics compared with electromagnetic and optical waves. However, acoustic communications are fundamentally limited by the low sound propagation speed, approximately 1500 m/s in water, and by the low available frequency bandwidth. These severe physical constraints necessitate the design of Medium Access Control (MAC) protocols dedicated specifically to UASNs.

One of the major applications of UASNs is subsea asset monitoring in the oil & gas industry, which often requires sensor networks with line topologies, e.g. for leakage detection and corrosion monitoring in underwater pipelines [1][2]. A key feature of Linear UASN (LUASN) topologies is sparse connectivity. Packets are routed via multiple hops between neighbouring nodes as shown in Fig. 1. The typical connection ranges in LUASNs are sufficient for nodes to communicate with their neighbours, but short enough to avoid interfering with more distant nodes. Such sparse connectivity can be exploited by designing MAC protocols with spatial reuse of resources, i.e. multiple nodes transmitting and receiving packets simultaneously without collisions.

The state-of-the-art research on MAC protocols for LUASNs, and more generally UASNs with sparsely connected topologies, focuses on the design of spatial reuse patterns of slots in TDMA schedules [3][4][5]. Although MAC protocols based on the classical TDMA frame structure can achieve scalable collision-free packet scheduling, many spatial reuse TDMA, also referred to as Spatial-TDMA (STDMA), protocols proposed in the literature are limited to fixed connectivity and interference patterns to produce efficient and analytically tractable solutions, e.g. [3][4]. Furthermore, the main drawback of the protocols based on the classical TDMA frame structure is the need for extensive guard intervals to account for long propagation delays of acoustic waves, which often has a large negative impact on the network throughput. They also require clock synchronization at all network nodes, which is more challenging in UASNs than terrestrial systems, although typically only loose clock synchronization is required due to the long duration of the time slots.

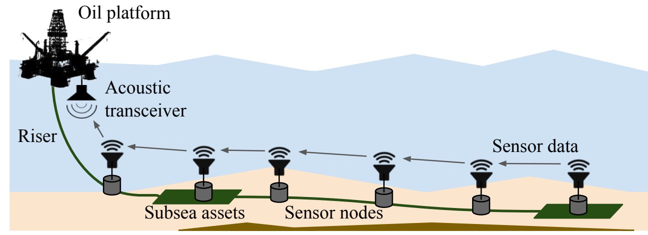


Fig. 1. Example of a linear UASN deployment for subsea asset monitoring.

The purpose of this paper is to propose the Linear Transmit Delay Allocation MAC (LTDA-MAC) protocol that enables unsynchronized packet scheduling in LUASNs without any constraints on the connectivity pattern or propagation delays. LTDA-MAC incorporates heuristic optimization of a packet schedule that is tailored to a given network deployment, taking into account the propagation delays to achieve high channel utilization. This extends our previous work on the application of TDA-MAC to single and dual hop network topologies in simulations and sea trials [6][7].

The rest of the paper is organized as follows: Section II presents the details of LTDA-MAC, Section III discusses the simulation results for a subsea pipeline monitoring scenario, and Section IV concludes the paper.

II. LTDA-MAC

TDA-MAC is an unsynchronized MAC protocol proposed in [6]. It was designed for centralized scheduling of data transmissions from underwater acoustic sensor nodes connected to a single gateway node, and extended to dual-hop network topologies in [7]. Its main advantage over other MAC protocols found in the literature is that it can achieve network throughputs close to the channel capacity without clock synchronization at the sensor nodes. Instead, the timing of transmissions is achieved at every sensor node locally by counting a particular delay after receiving a request packet.

Fig. 1 depicts an example of a LUASN. Every sensor node only uses two connections - a node one hop closer to the

The authors are with the Department of Electronic Engineering, University of York, Heslington, York YO10 5DD, United Kingdom.

Corresponding author: Nils Morozs (nils.morozs@york.ac.uk).

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

Simulation code DOI: 10.15124/b3d822c5-c20b-4e98-8ff1-4ac6f37328fd

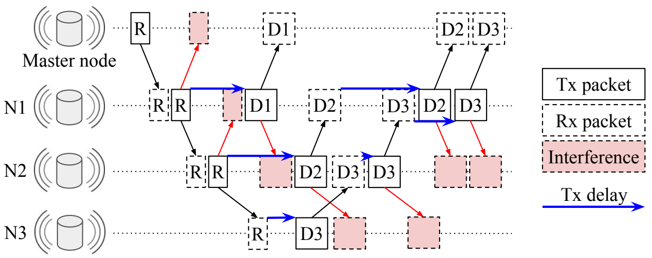


Fig. 2. Illustrative example of an LTDA-MAC schedule, where the master node gathers the data from three sensor nodes arranged in a linear topology. R - REQ packet, D - data packet.

sink node (*up the chain*) and a node one hop further *down the chain*. The job of a sensor node is to transmit its own packets up the chain and forward data packets from sensor nodes down the chain.

LTDA-MAC scheduling. The goal of the LTDA-MAC protocol proposed in this paper is to achieve high channel utilization in LUASNs using only the core basic TDA-MAC functions at every sensor node:

1) *Respond with data packet* - receive a data request (REQ) packet from the node up the chain and transmit its own data packet back after a transmit delay.
2) *Forward REQ packet* - receive REQ packet from the node up the chain and forward it down the chain.
3) *Forward data packet* - receive a data packet and forward it up the chain after a transmit delay.

Fig. 2 shows an illustrative example of an LTDA-MAC schedule, in this case based on a scenario with one-hop interference range. Nodes 1 and 2 are tasked with forwarding the REQ packets down the chain and forwarding data packets up the chain, in addition to sending their own data packets. The LTDA-MAC schedule is fully defined by the delays between sensor nodes receiving a REQ packet and responding with their own data packet, and by the delays between receiving a data packet from the node down the chain and forwarding it up the chain. For a network comprising N_{sn} sensor nodes, the transmit delays that define an LTDA-MAC schedule can be expressed by a triangular matrix $\boldsymbol{T}_{\textbf{tx}}$, where $T_{\text{tx}}[i,i]$ is the transmit delay assigned to node i for sending its own data packets, and $T_{\text{tx}}[i,j]$ $(i<j)$ is the transmit delay assigned to node i for forwarding data packets originating at node j. The triangular form of the matrix is because every sensor node has to schedule its own packet plus a packet from every sensor node down the chain.

Deriving an optimal LTDA-MAC schedule for a given linear UASN deployment requires a joint optimization of $N_{\text{sn}}(N_{\text{sn}}+1)/2$ transmit delays in the $\boldsymbol{T}_{\textbf{tx}}$ matrix, that minimize the LTDA-MAC frame duration:

$$\text{Minimize } \tau_{\text{frame}} = \max_{n}\{\tau_{\text{rx}}[n]\}, \text{ s.t. } n_{\text{col}} = 0 \qquad (1)$$

where n_{col} is the number of collisions, caused by the transmitted/received packets overlapping in time at any node, and $\tau_{\text{rx}}[n]$ is the packet delay of node n, defined as the time between the sink node sending the initial REQ packet and receiving the data packet from node n. It is calculated as:

$$\tau_{\text{rx}}[n] = \sum_{i=1}^{n}\Big(\tau_{\text{rp}} + 2\,\tau_{\text{p}}[i] + T_{\text{tx}}[i,n] + \tau_{\text{dp}}\Big) + (n-1)\tau_{\text{g}} \quad (2)$$

where τ_{rp} and τ_{dp} are the REQ and data packet duration respectively, $\tau_{\text{p}}[i]$ is the propagation delay on the i^{th} link of the linear topology (starting from the sink node), and τ_{g} is the guard interval used by the nodes between receiving a REQ packet and forwarding it down the chain.

The way of minimizing the LTDA-MAC frame length whilst eliminating the packet collisions, i.e. solving (1), depends on the pattern of interfering links and propagation delays, different for every deployment scenario. Therefore, we found it intractable to solve this optimization problem analytically for an arbitrary network topology. Instead, we propose the use of heuristic optimization, where the number of collisions is calculated empirically by simulating an LTDA-MAC frame inside the optimization loop.

LTDA-MAC Schedule Derivation. Our proposed heuristic optimization approach generates LTDA-MAC schedules based on the interference and propagation conditions measured during the network deployment. It evaluates the performance of an LTDA-MAC schedule by simulating a single frame using the given network deployment parameters.

We define the network deployment parameters as a tuple $\boldsymbol{\mathcal{N}} = \{\boldsymbol{I}, \boldsymbol{R}, \boldsymbol{T}_{\textbf{p}}, \tau_{\text{rp}}, \tau_{\text{dp}}\}$, where $\boldsymbol{I}$ is the interference matrix, $\boldsymbol{R}$ is the routing matrix, and $\boldsymbol{T}_{\textbf{p}}$ is the matrix of propagation delays. $\boldsymbol{I}$, $\boldsymbol{R}$ and $\boldsymbol{T}_{\textbf{p}}$ are established during the network discovery and setup stage, and are then periodically updated based on received data packets, e.g. as described in [6]. This process is sufficient to maintain an accurate topology estimate of a quasi-stationary underwater sensor network. $\boldsymbol{I}$, $\boldsymbol{R}$ and $\boldsymbol{T}_{\textbf{p}}$ are $N \times N$ matrices, where $N = 1 + N_{\text{sn}}$ is the total number of nodes, including one master (sink) node and N_{sn} sensor nodes. $\boldsymbol{I}$ and $\boldsymbol{R}$ are binary matrices. $I[i,j] = 1$ if node j is within the interference range of node i, and $I[i,j] = 0$ otherwise. $R[i,j] = 1$ if node i is the data packet destination for node j, and $R[i,j] = 0$ otherwise. $T_{\text{p}}[i,j]$ is the propagation delay from node i to node j.

The key requirement for our proposed simulation-in-the-loop approach is the design of an *objective function* that determines "how good" a particular solution for $\boldsymbol{T}_{\textbf{tx}}$ is, given the network deployment parameters $\boldsymbol{\mathcal{N}}$. We propose the following objective function to be minimized:

$$Q(\boldsymbol{\mathcal{N}}, \boldsymbol{T}_{\textbf{tx}}, \tau_{\text{g}}) = \Big(n_{\text{col}}(\boldsymbol{\mathcal{N}}, \boldsymbol{T}_{\textbf{tx}}, \tau_{\text{g}}) + 1\Big) \sum_{n=1}^{N_{\text{sn}}} \tau_{\text{rx}}[n], \qquad (3)$$

which is a function of $\boldsymbol{\mathcal{N}}$, $\boldsymbol{T}_{\textbf{tx}}$ and the *desired guard interval* τ_{g}. $n_{\text{col}}(\boldsymbol{\mathcal{N}}, \boldsymbol{T}_{\textbf{tx}}, \tau_{\text{g}})$ is the number of packet collisions, and $\tau_{\text{rx}}[n]$ is the packet delay for node n defined in (2).

Algorithm 1 LTDA-MAC protocol at the master node

```
1:  Create 𝓝 via initial network discovery
2:  Optimize LTDA-MAC schedule T_tx, given 𝓝 and τ_g
3:  Distribute rows of T_tx to corresponding sensor nodes
4:  while No changes in topology reported do
5:     Transmit data request packet (REQ)
6:     Receive data packets from sensor nodes
7:     if Topology change reported by sensor node then
8:        Update 𝓝
9:        go to Step 2
10:    end if
11: end while
```

The number of collisions $n_{\text{col}}(\boldsymbol{\mathcal{N}}, \boldsymbol{T}_{\text{tx}}, \tau_{\text{g}})$ is determined empirically via simulating an LTDA-MAC frame inside the optimization loop. If any pair of Tx/Rx packets overlaps in time at the same node, *or is separated by less than* τ_g, a collision in the schedule is detected and $n_{\text{col}}(\boldsymbol{\mathcal{N}}, \boldsymbol{T}_{\text{tx}}, \tau_{\text{g}})$ is incremented. The desired guard interval τ_{g} is a key feature in the derivation of an LTDA-MAC schedule. It enforces a minimum time separation between scheduled packets which makes the network robust against small changes in node positions, errors in propagation delay estimates, and the multipath delay spread. The objective function proposed above drives the optimization algorithm towards providing the key characteristics of an efficient transmission schedule: *zero collisions* as the main priority, *short packet delays* and *short frame duration* (the total time between sending the first REQ packet and receiving the last data packet), both driven by minimizing the sum of the packet delays in (3).

Two-Stage Heuristic Optimization. We propose the use of a two-stage heuristic optimization approach depicted in Fig. 3, comprising a Genetic Algorithm (GA) for ”coarse” optimization of the LTDA-MAC schedule and Particle Swarm Optimization (PSO) as the ”fine” optimization stage, utilizing the advantages and mitigating the drawbacks of both algorithms [8]. Driven by simple random mutations of the chromosomes representing candidate solutions, GAs are good at exploring a large solution space without getting stuck in local optima. In contrast, PSO is based on a continuous search, where the particles (representing candidate solutions) ”travel” through the solution space in particular directions with particular velocities. Therefore, PSO exhibits better convergence behaviour, but is more prone to getting stuck in local optima, compared with GAs.

Our proposed two stage approach works as follows:

1) The GA performs coarse optimization and produces an intermediate LTDA-MAC schedule $\boldsymbol{T}'_{\text{tx}}$, with the lower and upper bound on $\boldsymbol{T}'_{\text{tx}}$ values of $\left[\tau_{\text{g}}, N^2(\tau_{\text{d}}{+}\tau_{\text{g}})\right]$
2) $\boldsymbol{T}'_{\text{tx}}$ is used to reduce the upper bound of the solution space to $\left(\max\{\boldsymbol{T}'_{\text{tx}}\} + N(\tau_{\text{d}}{+}\tau_{\text{g}})\right)$ for the PSO stage
3) Initialized with $\boldsymbol{T}'_{\text{tx}}$ and using the reduced search space, the PSO algorithm resumes the optimization process to produce the final LTDA-MAC schedule $\boldsymbol{T}_{\text{tx}}$.

The GA and PSO algorithms were implemented using MATLAB R2018b with the Global Optimization Toolbox. We fixed the algorithm parameters (shown in Fig. 3) such that the maximum number of schedule evaluations is 10^6. We impose this fixed limit because the LTDA-MAC protocol is designed to perform heuristic optimization during the network deployment, as shown in Algorithm 1. Therefore, it requires an acceptable upper bound on the computation time to derive an LTDA-MAC schedule. We assume that the master node deployed at the sea surface is not as resource-constrained as the sensor nodes, and is capable of evaluating 10^6 schedules in a short time, e.g. under 1 min. For example, our MATLAB implementation takes 0.4 ms per single schedule evaluation on a standard Desktop PC - Intel Core i7-4790 CPU @ 3.60GHz, 8 GB RAM, Windows 10. We believe that a more efficient implementation (e.g. in C/C++) and the use of parallel computation on a multi-core CPU or GPU will make it feasible to perform millions of schedule evaluations in under 1 min, thus enabling online heuristic optimization as part of the LTDA-MAC protocol. Furthermore, another way of speeding up the proposed heuristic optimization approach is to reduce the solution space explored by the GA+PSO algorithm, e.g. by discretizing the possible values of $\boldsymbol{T}_{\text{tx}}$ into a finite set. Reducing the search space without significant degradation of performance is a subject of further work.

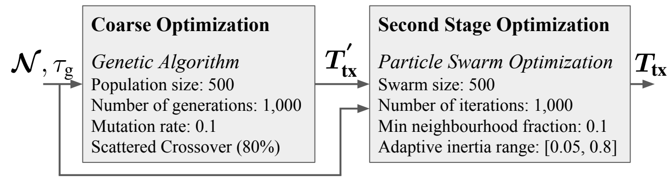


Fig. 3. Two-stage GA+PSO derivation of LTDA-MAC schedules.

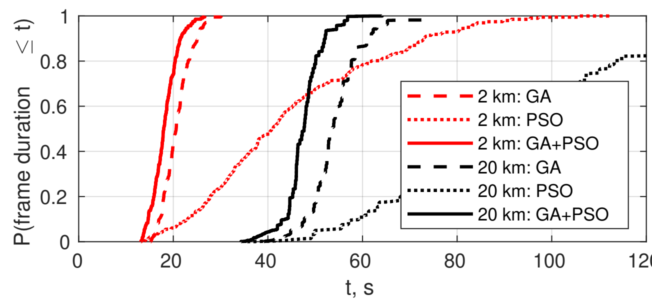


Fig. 4. The two-stage GA+PSO algorithm derives reliably better LTDA-MAC schedules than the Genetic Algorithm (GA) or Particle Swarm Optimization (PSO) applied separately.

III. Simulation Results

Simulation Setup. In this section we present the results of deriving LTDA-MAC schedules for a pipeline monitoring scenario depicted in Fig. 1. The maximum sea depth is 500 m. The pipeline at 480 m depth is connected to the platform at the sea surface through a riser, whose shape is modelled as a quarter-circle. We simulate LTDA-MAC using two different pipeline lengths: 2 km and 20 km including the riser. 11 nodes (1 sink + 10 sensor nodes) are spread across the length of the pipeline, initially at equidistant points. We then generate 50 sets of node positions with random horizontal offsets (0-20 m and 0-200 m for the two scenarios, respectively) in random directions from the initial equidistant points. This ensures statistical validity of the results under random variations in the channel characteristics. The wideband multipath channel with 24 kHz centre frequency and 7.2 kHz bandwidth was modelled using the ray tracing method described in [7]. We assume the threshold for interfering link detection of 0 dB Signal-to-Noise Ratio (SNR), i.e. all links with SNR $\geq$ 0 dB were marked with 1 in the interference matrix $\boldsymbol{I}$, while all of the wanted links in the linear topology also have SNR $>$ 0 dB. We use the ambient noise model with 10 m/s wind speed and 0.5 shipping activity factor [7]. The other parameters for the 2 km and 20 km pipeline scenarios, respectively, are:

1) 140 dB re μPa^{2}m^2 source level, 200 ms data packets, 50 ms REQ packets, 25 ms guard interval,
2) 170 dB re μPa^{2}m^2 source level, 500 ms data packets, 100 ms REQ packets, 100 ms guard interval.

Discussion. Fig. 4 demonstrates the benefits of our proposed two-stage GA+PSO approach, which reliably produced better solutions than GA or PSO applied on their

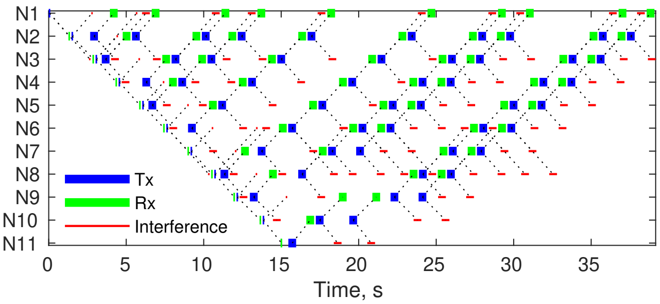
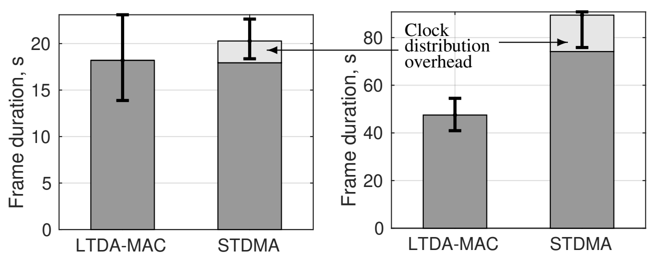


Fig. 5. LTDA-MAC schedule derived by the GA+PSO algorithm for the linear UASN deployed on a 20 km long pipeline.



(a) 2 km long network (b) 20 km long network

Fig. 6. Comparison of the average frame duration achieved by LTDA-MAC and STDMA. The error bars show the 5th and 95th percentiles.

own using the same overall number of evaluations. Fig. 5 shows an example of the LTDA-MAC schedule produced by the GA+PSO algorithm for the 20 km pipeline scenario. It demonstrates how long propagation delays of 2 km long links were exploited to schedule the transmissions, receptions and interference in a collision-free pattern, thus reducing the time it takes for all 10 sensor nodes to send their data through the linear network back to the master node.

Fig. 6 provides a baseline comparison of the LTDA-MAC schedules produced by the heuristic optimization algorithm with a state-of-the-art TDMA approach with spatial reuse of time slots (STDMA) [4][5], tailored to the linear network scenario studied in this paper. STDMA is based on the classical TDMA frame structure, i.e. a set of synchronized time slots for collision-free multiple access. To avoid inter-slot interference, the duration of the time slot is determined as follows:

$$\tau_{\text{slot}} = \tau_{\text{dp}} + \max_{I[i,j]=1} \{T_{\text{p}}[i,j]\} + \tau_{\text{g}}, \tag{4}$$

comprising the duration of the data packet, the longest propagation delay among all detected links and the guard interval. The spatial reuse pattern is achieved by deriving an $N_{\text{sn}} \times L$ matrix, where L is the number of time slots, indicating which node transmits in which time slot, such that L is minimized with no collisions. The STDMA frame length is then calculated as: $\tau_{\text{frame}} = L\tau_{\text{slot}}$. The mean, 5th and 95th percentile results in Fig. 6 were obtained based on 50 network topologies with small deviations in node positions and changes in interference patterns. Furthermore, since LTDA-MAC is based on heuristic optimization with stochastic behaviour, we ran it with 50 different random seeds for every topology to ensure statistical validity of the results.

Fig. 6b shows that LTDA-MAC significantly outperforms STDMA in the 20 km pipeline scenario. LTDA-MAC achieves 36% shorter frames on average, despite no requirement for clock synchronization. If the clock distribution delay is included in the STDMA frame duration for a fairer comparison (because LTDA-MAC implicitly distributes the local time reference via the REQ packet), then LTDA-MAC is on average 47% faster than STDMA. The 2 km pipeline scenario is less favourable for LTDA-MAC due to shorter propagation delays and more interfering links; whereas it is more favourable for STDMA due to a significantly reduced time slot duration (because of shorter propagation delays). Nevertheless, Fig. 6a shows that LTDA-MAC still provides a slight improvement over STDMA, if the clock distribution overhead is considered, demonstrating the efficacy of LTDA-MAC for networks with short or long propagation delays, and arbitrary interference patterns. All LTDA-MAC schedules produced in these simulations have zero collisions.

IV. Conclusion

The LTDA-MAC protocol proposed in this paper is capable of providing efficient scheduling of data packets in LUASNs without prerequisite assumptions about propagation delays and the interference pattern among the sensor nodes. The key part of the LTDA-MAC protocol is the online application of a heuristic optimization algorithm comprising a GA and PSO to derive a collision-free transmission schedule for the given deployment scenario. Simulations of a LUASN deployed on a 20 km underwater pipeline showed that, despite no requirement for clock synchronization, LTDA-MAC can exploit long propagation delays to reduce the frame duration by 36% on average, compared with the conventional STDMA approach. Continuing from our previous work on single-hop and dual-hop TDA-MAC, this paper demonstrates the efficacy of the TDA-MAC protocol framework in a wide range of network topologies, and is a key step in our further work on developing the TDA-MAC protocol for unsynchronized scheduling in UASNs with arbitrary multi-hop topologies.

References

[1] E. Felemban, et al., “Underwater sensor network applications: A comprehensive survey,” *International Journal of Distributed Sensor Networks*, vol. 11, no. 11, p. 896832, 2015.

[2] S. Ali, et al., “Simplimote: A wireless sensor network monitoring platform for oil and gas pipelines,” *IEEE Systems Journal*, vol. 12, no. 1, pp. 778–789, 2018.

[3] S. Lmai, et al., “Throughput-efficient super-TDMA MAC transmission schedules in ad hoc linear underwater acoustic networks,” *IEEE Journal of Oceanic Engineering*, vol. 42, no. 1, pp. 156–174, 2017.

[4] M. Luque-Nieto, J.M. Moreno-Roldán, J. Poncela, and P. Otero, “Optimal fair scheduling in S-TDMA sensor networks for monitoring river plumes,” *Journal of Sensors*, no. 8671516, 2016.

[5] R. Diamant, G. N. Shirazi, and L. Lampe, “Robust spatial reuse scheduling in underwater acoustic communication networks,” *IEEE Journal of Oceanic Engineering*, vol. 39, no. 1, pp. 32–46, 2014.

[6] N. Morozs, P. Mitchell, and Y. Zakharov, “TDA-MAC: TDMA without clock synchronization in underwater acoustic networks,” *IEEE Access*, vol. 6, pp. 1091–1108, 2018.

[7] N. Morozs, P. Mitchell, and Y. Zakharov, “Unsynchronized dual-hop scheduling for practical data gathering in underwater sensor networks,” in *Proceedings of IEEE UComms’18*, 2018.

[8] P. Ghamisi and J. A. Benediktsson, “Feature selection based on hybridization of genetic algorithm and particle swarm optimization,” *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 2, pp. 309–313, 2015.