A Lightweight AI-powered Framework for Multimodal Underwater Networks

Filippo Donegà[1], Filippo Bragato[2], Filippo Campagnaro[3], and Michele Zorzi[4]

[1-4]Department of Information Engineering, University of Padova

Filippo Donegà, Department of Information Engineering, University of Padova, Italy
E-mail: filippo.donega@unipd.it

Abstract: *Underwater acoustic communication enables numerous long-range applications, but the low bitrate and long propagation delay that characterize the underwater acoustic channel make it difficult to support demanding applications like real-time control of Remotely Operated Vehicles (ROVs). Alternative technologies such as optical, electromagnetic radio frequency, and magneto inductive communications could be preferable, providing significantly higher rates albeit at much shorter distances. Underwater multimodal networks, combining multiple communication technologies, currently represent the best strategy for optimizing the communication performance, given that no single underwater communication technique is universally superior. It is especially important to understand how to combine the transmission technologies optimally, and how to select which one to use depending on the channel conditions and the requirements imposed by the intended application. In particular, multimodal communication can be seen as an unknown Markov Decision Problem, and Reinforcement Learning (RL) is a practical way to deal with it. In this work we showcase how Artificial Intelligence can be used to optimize multimodal communication by presenting and evaluating through DESERT simulations a lightweight QoS-based RL framework that optimally selects the best transmission medium in multimodal underwater networks. Results highlight how this framework outperforms traditional approaches, achieving better performance in many communication parameters while still being suitable for practical deployments onboard lightweight single-board computers.*

Keywords: *Underwater wireless sensor networks, multimodal communication, reinforcement learning, AI, network simulation.*

1. INTRODUCTION

Acoustic communication represents the most mature and established underwater transmission technology, enabling robust connectivity over ranges of several kilometers. This makes it well-suited for many common Underwater Wireless Sensor Network (UWSN) applications, including coastal surveillance, underwater pipeline monitoring, and water quality assessment among others [1, 2]. However, the underwater acoustic channel is significantly affected by environmental noise from vessels and wind, and further suffers from multipath propagation and Doppler effects, which particularly degrade performance in shallow, mobile, or noisy environments. Additionally, the slow speed of sound in water results in considerable propagation delays, and the data rates supported by acoustic modems are typically low [3].

These limitations make acoustic communication unsuitable for applications requiring higher throughput, such as real-time drone control or video streaming. In these instances, alternative communication methods—namely optical, Radio Frequency (RF), or Magneto Inductive (MI) technologies—may offer better performance.

Optical communication, in particular, supports significantly higher data rates, reaching several Gbps under ideal conditions, and is therefore more appropriate for data-intensive applications. However, while immune to multipath effects and acoustic noise, optical signals are highly susceptible to environmental factors such as water turbidity (which causes signal scattering and attenuation) and ambient light noise near the surface [4], and their effective range is typically limited to a few tens of meters.

Since no single technology performs best under all conditions, multimodal underwater networks (integrating multiple transmission technologies) have emerged as a promising solution, presenting the novel challenge of dynamically selecting the most appropriate communication mode in response to varying environmental conditions and application requirements.

Nevertheless, most existing multimodal solutions struggle with scalability and adaptability in dynamic underwater environments, often requiring the choice of arbitrary thresholds or parameters in order to work properly. Moreover, established solutions for handling multimodal communication in terrestrial networks are generally too expensive (computationally and/or energetically) and therefore difficult to apply to real-life underwater deployments, which are normally characterized by resource-constrained devices.

The use case guiding the design and evaluation of the proposed framework involves two mobile Autonomous Underwater Vehicles (AUVs) cooperating to manipulate a submerged object, each one equipped with both an acoustic and an optical modem. In such context, advanced computer vision and/or AI mechanisms would be used in close proximity, requiring the exchange of real-time position updates and state information between the drones, but acoustics would still be needed for exchanging reliable ranging data to allow target approach from a distance. Therefore each traffic is assumed to be related to a specific application, which in turn poses specific Quality of Service (QoS) requirements to be satisfied.

This paper presents a lightweight, adaptive, and scalable QoS-driven Reinforcement Learning (RL) framework that autonomously selects the optimal communication mode based on real-time channel conditions and specific application requirements. The proposed approach is designed to be efficient enough for low-power underwater nodes, with the additional advantage of supporting offline training for enhanced flexibility and deployment readiness in challenging environments.

2. REINFORCEMENT LEARNING AND MARKOV DECISION PROCESSES

Reinforcement Learning (RL) is a machine learning paradigm in which an agent learns to make decisions by interacting with an environment, receiving feedback in the form of rewards. More precisely, an RL agent aims at learning a policy that maximizes its cumulative expected reward, which is a problem that can be formalized with a Markov Decision Process (MDP) [5].

At each discrete time step t, the agent observes a state $S_t \in \mathcal{S}$, selects an action $A_t \in \mathcal{A}$, receives a reward $R_t = \mathcal{R}(S_t, A_t)$, and transitions to a new state S_{t+1} according to the transition probability function $\mathcal{P}(S_{t+1}|S_t, A_t)$. Its behavior is governed by a policy π, defined as a probability distribution over the set of actions given a state:

$$\pi(a|s) = P(A_t = a \mid S_t = s). \tag{1}$$

The expected cumulative future reward G_t to be maximized is defined as the sum of the rewards obtained from the current time step to the end of the episode, and discounted by γ at each time step:

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}. \tag{2}$$

The agent does not have access to the MDP's $\mathcal{P}$ and $\mathcal{R}$, and must learn it by interacting with the environment. The algorithm proposed in this work belongs to a specific class of RL algorithms, called *value-based* methods, in which the agent learns the value of each state-action pair $q_\pi(s, a)$, defined as the expected cumulative future reward when starting from state s and taking action a, and following policy π after the first step:

$$q_\pi(s, a) = \mathbb{E}_\pi[G_t|S_t = s, A_t = a]. \tag{3}$$

Afterwards, it can select the best action in each state by following a policy that is greedy with respect to the learned values, i.e.,

$$\pi^*(s) = \arg\max_a q_\pi(s, a). \tag{4}$$

Value-based methods typically alternate between two phases: the evaluation phase, where the agent learns the value of each state-action pair, and the improvement phase, where it updates its policy to be greedy with respect to the learned values. This guarantees that the agent will eventually converge to the optimal policy. Another important concept in RL is the distinction between on-policy and off-policy algorithms. In on-policy algorithms, the agent learns the value of the policy it is following, while in off-policy algorithms, the agent learns the value of a different policy at the expense of weaker convergence guarantees.

3. SYSTEM ARCHITECTURE

The system consists of two components at each node: the *agent* and the *wait system*. The agent is responsible for selecting the optimal communication technology for each transmission, while the wait system handles a finite-size queue of packets to be transmitted. When an application generates a packet, it enters the wait system's queue. This triggers the agent, which evaluates each packet and chooses the best transmission method. If a packet cannot be sent (e.g., due to poor channel conditions), it is re-queued and the process is paused to preserve packet order. Once all applications stop producing packets, the agent performs a final transmission attempt for any remaining packets, after which the system stops.

The state of the system is configurable and composed by a subset of 16 possible parameters, selectable through a mask, and normalized. These parameters include noise readings and power of the last received transmission for each technology, acoustic and optical channel-related metrics (i.e., temperature, attenuation coefficient, wind speed and intensity of shipping activity), spatial information like distance and depth, and size of Medium Access Control (MAC) and wait queues. The mask used for this work includes all of the aforementioned parameters.

For each packet stored in the wait queue, the RL agent can undertake three possible actions: forward it to the acoustic MAC module (i.e., send in acoustic), forward it to the optical MAC module (i.e., send in optical) or wait to send the packet (i.e., leave it in the queue and pause the process until next iteration).

Each agent receives a reward based on the QoS requirements of the packet being sent. The reward function is defined to guarantee that the agent learns to select the best transmission technology for each packet, while also considering the stability of the training.

The framework as presented in this paper involves two different types of traffic, as will be clarified in Section 4, and two dedicated agents: one aiming to maximize throughput (i.e., Maximum Throughput Agent (MTA)), and the other minimizing jitter (i.e., minimum Jitter Agent (mJA)).

Concerning the MTA, the reward function is based on the packet size, the reception of the packet, and the delay experienced by the packet. In particular, if the packet is successfully received, the reward is calculated as the packet size divided by the delay. If the agent chooses to wait and not send the packet, the reward is zero. Conversely, if the packet is sent but fails to reach its destination, a negative reward is assigned.

The mJA agent's reward function is based on the jitter experienced by the packet. In particular, if the packet is received, the reward is equal to a constant minus the difference between the current and previous packet's delay. If the packet is not sent, the reward is strongly negative to discourage the agent from waiting. If the packet is instead sent but not received, the reward is negative.

The learning agent is based on the Q-learning algorithm, which is a value-based RL algorithm. In particular, we implemented a continual non-episodic version with a linear approximator of the standard Q-learning algorithm. The agent is similar to the Differential semi-gradient Sarsa presented in [5], but it uses an off-policy approach to learn the value of the policy.

It is important to note that the introduction of the off-policy approach does not guarantee the convergence of the algorithm, since we are training the agent using a bootstrapped version of an off-policy algorithm with a linear approximator. The choice of this type of algorithms, that do not have a proof of convergence, is common in the RL ecosystem. Indeed, the algorithm is still able to learn the value of the policy and to select the best action in each state, like many other well-known algorithms that fall into this category, such as the DQN algorithm [6].

4. EXPERIMENTAL RESULTS

Each presented result is obtained by averaging over outcomes from 15 campaigns, where a campaign is defined as a set of simulations (or epochs) that are related to the same agent. Simulations are performed using DESERT Underwater [7], a publicly available underwater network simulator developed and maintained by the SIGNET group at the University of Padova. In this work, we define training and evaluation epochs differently: during a training epoch, the agent is trained using the RL algorithm described in Section 3, weights are updated at each step using the semi-gradient technique, and the agent utilizes an ϵ-greedy policy to explore the environment. Training epochs belonging to the same campaign share the same environment setup but with different realization (i.e., different RNG seed), hence the agent is trained in independent and identically distributed (iid) scenarios. Instead, during an evaluation epoch, the agent is always tested in the same environment, its weights are not updated, and ϵ is set to 0, meaning that the agent always selects the best action. Each campaign alternates 100 training epochs and 100 evaluation epochs, until reaching a total of 1000 training epochs. This way, we can measure the performance of the agent during the training process, and evaluate its final performance.

The simulated scenario, approximating the use case described in Section 1, consists of a Remotely Operated Vehicle (ROV) approaching a static node at exponentially decreasing speed, starting from a distance of 2600 m. The communication is bidirectional and involves two different ongoing traffics, each one paired to a dedicated multimodal controller (and agent) that optimizes a different reward function. Table 1 reports traffic details and respective QoS requirements optimized by the relative RL agent.

#	Data type	Period	Packet size	Agent
1	Measurement data	60 s $\rightarrow$ 17 s Exponentially decreasing with distance	64 Bytes	MTA
2	Real-time control data	10 s	100 Bytes $\rightarrow$ 750 Bytes Exponentially increasing with distance	mJA

Table 1: Simulated traffic types and respective optimization goals.

To simulate acoustic transmission, DESERT's acoustic PHY layer (which models acoustic propagation according to [8]) is employed. Furthermore, acoustic interference is considered by utilizing DESERT's MEANPOWER model. Optical transmission is instead simulated using an extended version of DESERT's optical PHY layer, equipped with the capability of providing channel statistics to the multimodal controller via cross-layer messages. This layer models optical propagation according to [9], further adding background ambient light noise in the form of a per-depth look-up table. Optical interference is also taken into account by employing DESERT's MInterference/MIV plugin. Acoustic and optical noise factors are slowly and randomly varied throughout the simulation. More specifically, wind speed starts at 10 m/s and can get to a minimum of 0 m/s (calm) or a maximum of 20 m/s (gale). Instead, the attenuation coefficient starts at 0.30 and is capped between 0.15 (clear ocean) and 0.40 (coastal ocean). Acoustic PHY bitrate and central frequency are set to 2500 bps and 26 kHz respectively, whereas optical PHY bitrate and carrier are set to 1 MHz and 10 MHz.

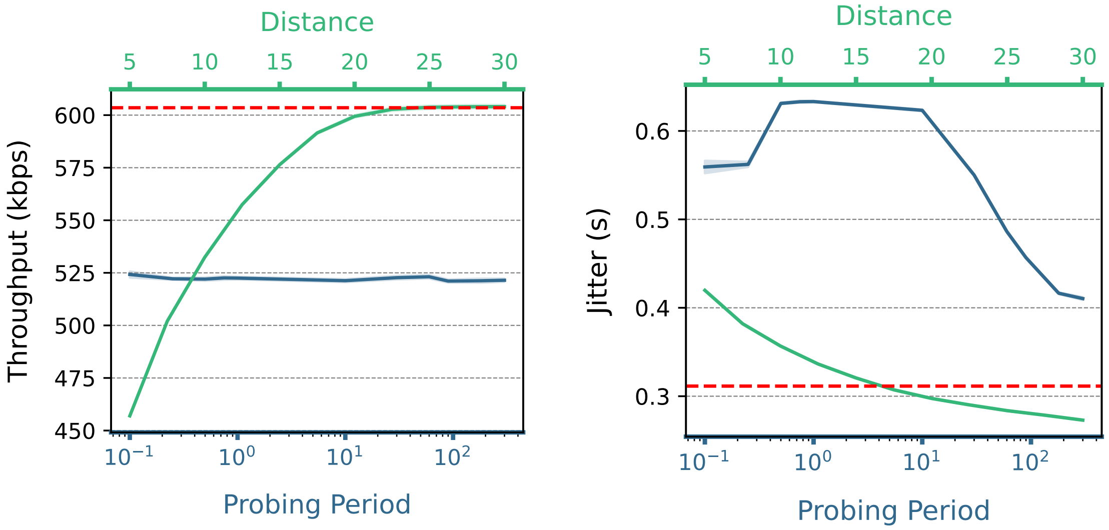
The performance of the proposed system is evaluated by comparing it against two baselines, representative of much simpler methods for multimodal control:

- *Range-based:* The choice of which technology to use for each packet transmission depends upon the distance between the two nodes. If the ROV is farther than a predefined distance d from the sink, acoustic communication is employed. Otherwise, every packet is sent in optical;
- *Optical probing:* Small probe packets are periodically sent in optical, and we call the period between probe transmissions τ. Once a probe packet is received, optical transmission is employed for a predetermined amount of time. In simulations, this time is set as equal to the inter-generation time of optical probes, allowing continuous optical transmission in case subsequent probes are correctly received.

Fig. 1 depicts the performance of the proposed framework in terms of throughput and jitter, compared with the two baselines. Results achieved by the RL agent are rendered as red dashed lines, while the performance results for range-based and optical probing are represented by green and blue solid lines respectively.

In Fig. 1a, the displayed throughput is computed as the ratio between the total number of received packets and the overall time taken to receive them. The proposed framework (red dashed line) achieves a throughput over $600\,\text{kbps}$, which is more than $13\,\%$ higher than the throughput obtained by the probing method (blue solid line) and outperforms the range-based performance (green solid line) for $d < 20\,\text{m}$, getting instead similar results for $d > 20\,\text{m}$. This is due to the fact that the range-based method is not able to adapt to the changing conditions of the environment, while the proposed framework is able to adapt and to select the best transmission technology for each packet.

In Fig. 1b, jitter (also known as Packet Delay Variation (PDV)) is defined as the average difference between the delay of each packet and that of the subsequent one. The proposed framework achieves a jitter of $0.3\,\text{s}$, heavily outperforming the probing method, which tends to switch more often between the two technologies, and the range-based method when the distance is < 20 m.



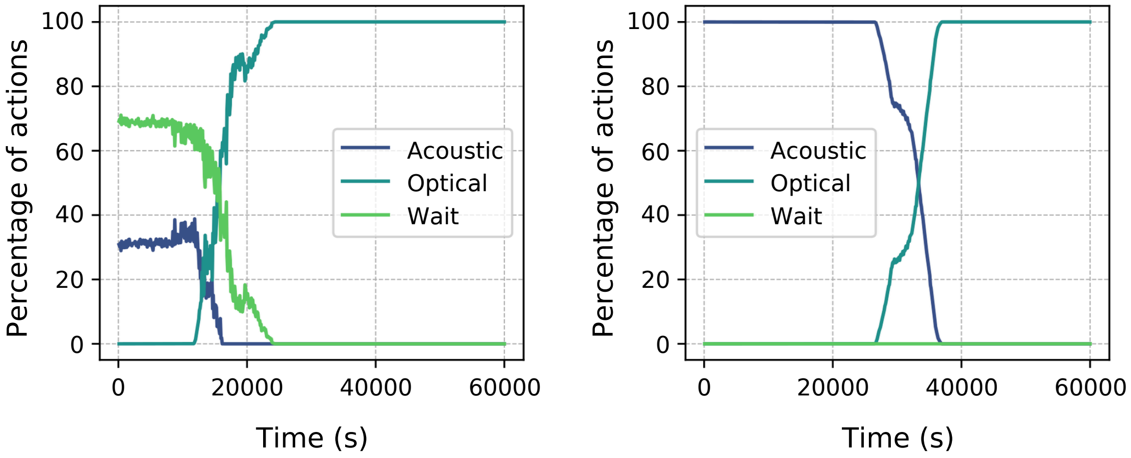
(a) Average throughput.

(b) Average jitter.

Figure 1: Comparison of the proposed framework (red dashed line) with the two baselines (green and blue solid lines for range-based and optical probing respectively) in terms of throughput and jitter.

In Fig. 2, the percentage of packets sent in each technology is displayed over time, showing how the RL agent changes the transmission technology throughout the course of a simulation. In particular, Fig. 2a shows that the MTA initially transmits a few packets acoustically while holding most of them in the wait queue. It begins sending packets only when the optical link is available. This behavior arises because using the acoustic link exclusively would congest the network, ultimately reducing throughput. Instead, in Fig. 2b, we can see that the mJA has a sudden transition from sending packets in the acoustic channel to sending them in the optical channel. This is due to the fact that the mJA agent is trained to minimize the jitter, and each switch between the two technologies introduces some jitter, hence the agent tries to minimize the number of switches between the two technologies.

To demonstrate that the presence of multiple agents does not affect the performance of the system, we also plot the actions taken by each agent over the training epochs. In Fig. 3, we



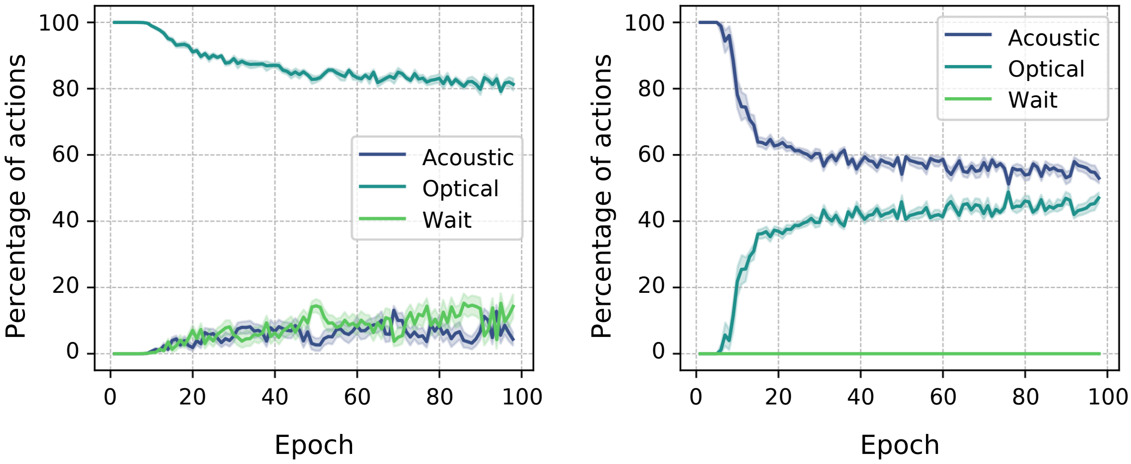
(a) MTA's actions.

(b) mJA's actions.

Figure 2: Actions taken by the MTA and mJA agents over time.

display the actions taken by each agent during the training. The actions are the ones taken in the evaluation phase, and the results are averaged over 15 campaigns. Results show that the mJA agent is able to learn its policy faster than the MTA agent, but the evolution of the policy of the MTA does not affect the stability of the mJA agent.

This ensures that the training of the two agents is stable and that the presence of multiple agents does not affect the convergence of the system. Even if the two agents are trained in the same environment, they are able to learn their policy independently and to converge to the optimal policy.



(a) MTA's actions.

(b) mJA's actions.

Figure 3: Actions taken by the MTA and mJA agents over training.

5. CONCLUSION

The proposed QoS-based AI framework for handling multimodal communication in underwater networks was tested in simulation and proved to perform better than the two baselines in most cases. The main strengths of the proposed framework are its high customizability, its ease of extension to new applications with different QoS requirements, and its adaptability, since the RL agent can automatically adapt to new scenarios and use cases. Future work will involve agents supervision through multi-agent RL techniques, that enable the coordination between agents to maximize the total reward, the addition of new traffic types (e.g., video stream) and new QoS requirements to be optimized (e.g., Packet Reception Rate (PRR) optimization).

6. ACKNOWLEDGMENTS

This work was partially supported by the PRIN project BEASTIE (2022FZJ7MR), funded by the Italian Ministry of University and Research; and the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGeneration EU, partnership on "Telecommunications of the Future" (PE0000001-program "RESTART").

REFERENCES

[1] E. Felemban, F. K. Shaikh, U. M. Qureshi, A. A. Sheikh and S. B. Qaisar: "Underwater Sensor Network Applications: A Comprehensive Survey", *International Journal of Distributed Sensor Networks*, 11(11), 896832 (2015).

[2] F. Campagnaro, F. Steinmetz and B. C. Renner: "Survey on Low-Cost Underwater Sensor Networks: From Niche Applications to Everyday Use", *Journal of Marine Science and Engineering*, 11(1), 125 (2023).

[3] F. Campagnaro, R. Francescon, P. Casari, R. Diamant and M. Zorzi: "Multimodal Underwater Networks: Recent Advances and a Look Ahead" in *Proceedings of the 12th International Conference on Underwater Networks & Systems (WUWNet)*, 2017.

[4] A. Signori, F. Campagnaro and M. Zorzi: "Modeling the Performance of Optical Modems in the DESERT Underwater Network Simulator" in *IEEE Fourth Underwater Communications and Networking Conference (UComms)*, 2018.

[5] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*, (MIT press Cambridge, 2018).

[6] H. Van Hasselt, A. Guez, and D. Silver: "Deep reinforcement learning with double q-learning" in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2016.

[7] F. Campagnaro, R. Francescon, F. Guerra, F. Favaro, P. Casari, R. Diamant and M. Zorzi: "The DESERT Underwater Framework v2: Improved Capabilities and Extension Tools" in *IEEE Third Underwater Communications and Networking Conference (UComms)*, 2016.

[8] M. Stojanovic: "On the Relationship Between Capacity and Distance in an Underwater Acoustic Communication Channel" in *ACM SIGMOBILE Mobile Computing and Communications Review*, 11(4), 34-43, 2007.

[9] D. Anguita, D. Brizzolara, G. Parodi, and Q. Hu: "Optical Wireless Underwater Communication for AUV: Preliminary Simulation and Experimental Results" in *IEEE Oceans - Spain*, 2011.