Design and Evaluation of a Reliable Transport Protocol for Underwater Networks

Enrico Disaro'[†], Vincenzo Cimino[†], Filippo Campagnaro[†]

***Abstract*—Reliable data communication in underwater networks is crucial for a wide range of applications: transmission of multimedia content such as images for environmental monitoring, underwater exploration and surveillance. However, the harsh characteristics of the underwater acoustic channel (high latency, limited bandwidth, and high bit error rate) pose significant challenges to achieving reliable end-to-end data delivery, especially in multi-hop networks.**

The goal of this work is to design, implement and evaluate via simulations a transport layer protocol within the DESERT underwater network simulator [1]. The protocol needs to ensure in-order delivery of messages to the application layer and incorporate an end-to-end Automatic Repeat reQuest (ARQ) mechanism to handle packet loss and ensure data reliability.

***Index Terms*—Underwater Acoustic Networks, Transport Protocol, DESERT Underwater.**

I. Introduction

Some applications of underwater networks are observation and control of marine life, environment, infrastructures, and many others [2]. However, underwater communication is an entire different world from terrestrial communications: conventional radio-frequency communication is unviable due to the high absorption property of the water medium. The main alternatives found until now are acoustic, visible light (VLC), and magnetic induction (MI) based communication [2], but they all present different challenges and tradeoffs in terms of reliability, distance and speed.

In this work, we focus on acoustic communication, which works on a relatively long distance compared to the alternatives, but suffers from a low bitrate and a non-negligible latency due to the propagation speed of sound of just 1500 m/s in the water. The communication is possible but not always successful, because of effects like Doppler and multi-path fading; this calls for a *transport protocol* that can ensure reliability, but it is not possible to simply use TCP because the latency makes handshakes and connections between nodes very hard, and the low bitrate means that headers need to be as small as possible for the throughput to be unaffected. Finally, the devices used for underwater communication are not necessarily very powerful, and they can introduce further constraints.

The goal of this work is to build a transport layer protocol that offers reliability and can be adapted to different scenarios, despite the limitations of the devices and the channel. Using the DESERT framework, the protocol will be tested through a series of simulations set under slightly different conditions, and its performance will be evaluated in terms of *throughput*, *packet delivery ratio* and *delay*.

The rest of this work is structured as follows: Sec. II and III describe, respectively, the scenario of the simulations and the common protocol stack that has been employed. Sec. IV lists the settings and the goal of each simulation and Sec. V shows the results; Sec. VI concludes the paper.

II. Scenario

The protocol is tested on a small underwater network composed of 3 nodes, placed as shown in Fig. 1 1000 m deep. Each node communicates directly with the other two through an acoustic channel, subject to noise and interference. Traffic is generated at a constant average rate for 3600 s (1 hour) between each pair of nodes.

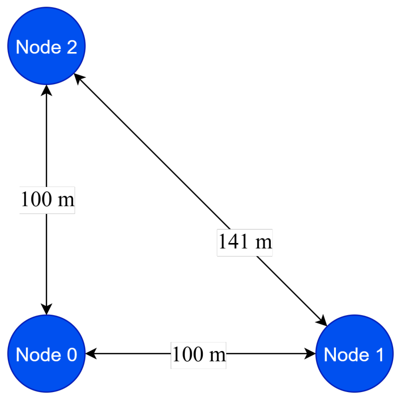


Fig. 1: Placement of the nodes.

The performance of the system is evaluated under slightly different conditions of the nodes, in order to show how the protocol can be tweaked to work better in each situation.

III. Protocol stack description

The simulations are performed with the DESERT underwater network simulator, where each node is equipped with the protocol stack shown in Fig. 2. In the following, we provide a brief description of each protocol:

- `UW/CBR` is an application layer protocol that generates traffic with a *constant* bitrate. It generates packets of a fixed size according to a Poisson random variable with averages equal to the average generation time.

[†]Department of Information Engineering, University of Padova, email: enrico.disaro.1@studenti.unipd.it, vincenzo.cimino@unipd.it, campagn1@dei.unipd.it

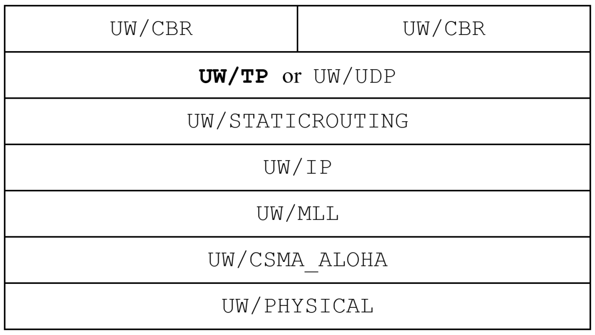


Fig. 2: Protocol stack of the nodes.

Only in-order packets are accepted, and there are no retransmissions.

- `UW/TP` is a connection-oriented transport layer protocol that offers reliability to the application layer by introducing retransmissions. It uses *ACKs* and *NACKs* to control the communication; one of its goals is to reduce overhead as much as possible, so ACKs are cumulative and the reception of a NACK is treated as an ACK for all packets preceding the NACKed one.
  There are three main parameters that define the behavior of the protocol and make it capable of adapting to various situations: the retransmission time of NACKs (*NACK_retx_time*), the maximum number of retransmissions of a NACK (*NACK_max_retx*), and the number of consecutive packets that must be received correctly before sending an ACK (*cum_ACK_param*).
- `UW/UDP` is a best-effort and connection-less transport layer with no retransmission capabilities.
- `UW/IP` and `UW/STATICROUTING` are used to assign addresses to the nodes and setup static routes between them; `UW/MLL` maps MAC addresses to IP addresses.
- `UW/CSMA_ALOHA` is a contention based MAC protocol, with retransmissions and reordering.
- `UW/PHYSICAL` simulates transmission using the Urick-Thorp formula (empirical equation that estimates the absorption of sound in seawater).

IV. Simulation Settings

For each simulation there are three devices equipped with a medium frequency acoustic modem. The channel used to communicate is under average conditions, with interference caused by wind, shipping activity, and concurrent transmissions. The power used for transmission is such that the nodes can easily reach each other, but interference is relevant. The simulation parameters are summarized in Table 1.

Four different simulation settings are conducted to show some strengths and weaknesses of the protocol; they differ in buffer sizes, generated traffic, and `UW/TP` parameters.

- **Scenario 1**: The goal is to study the behavior of the protocol in case of unlimited resources; for this reason, the buffer size and *NACK_max_retx* are set to values sufficiently large so that they do not constrain the system. Simulations are performed with each node generating traffic between 100 and 1000 bps, using both `UW/TP` and `UW/UDP`.
- **Scenario 2**: The goal is to study how the *NACK_max_retx* parameter affects the transmission in real conditions. Several simulations are performed varying this parameter between 1 and 100.
- **Scenario 3**: The goal is to study how, once the max number of retransmission is fixed, the *NACK_retx_time* parameter influences the performance of the system. Test values range from 1 to 100 s.
- **Scenario 4**: The goal is to understand the effect of the *cum_ACK_param* parameter on the communication. Values between 1 and 100 are used in different simulations.

TABLE 1: Underwater network settings

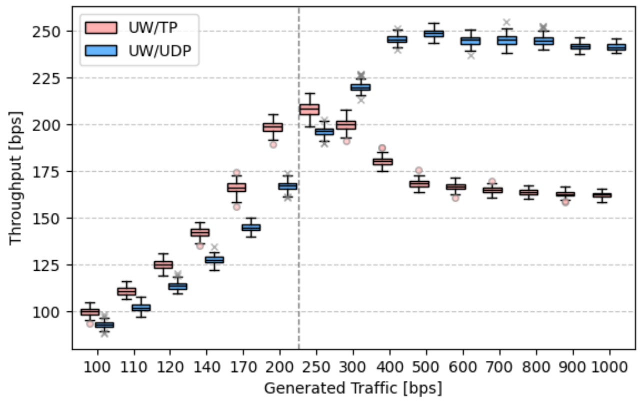
Parameter	Value
Wind speed	14 m/s
Acoustic modem frequency	25 kHz
Acoustic modem bandwidth	5 kHz
TX acoustic power	150 re 1μPa
Bitrate	4800 bps
Packet size	1000 bit

V. Results

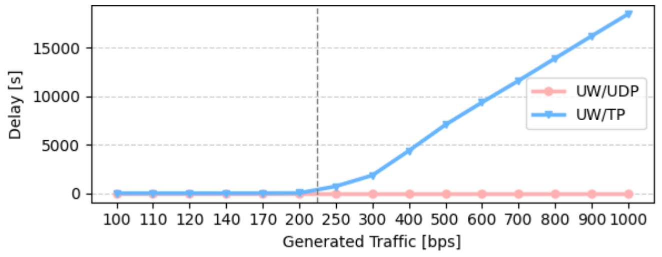
To assess the system performance, we perform 100 runs for each scenario. The metrics used for evaluation are:

- *Application layer packet delivery ratio* (PDR): the percentage of successfully delivered packets over to the total number of packets sent by the application layer.
- *Delay*: the time it takes to `UW/TP` to empty the send buffer after receiving the last packet from the application layer. For `UW/UDP` it is always considered 0 because it does not employ buffers and retransmissions.
- *Average troughput*: throughput is the ratio between the number of bits received by an application and the time required for their transmission. We consider the average throughput of all applications.

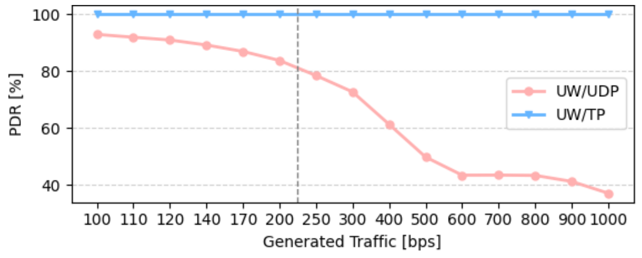
The first scenario employs infinite buffers and infinite retransmissions, allowing the protocol to always reach a PDR of 100 % at the cost of some delay. We also try to use `UW/UDP` for comparison. Figures 3a and 3b show that for low traffic `UW/TP` can easily reach a higher throughput than `UW/UDP` with no delay, because the extra traffic generated by the protocol, like NACKs and retransmission, does not exceed the channel capacity and manages to use the available bitrate more efficiently. However, when the generated traffic exceeds a certain threshold (in these settings around 225 bps), the control traffic increases enough to become relevant; this causes the application layer throughput to drop, and the delay to increase. After this point `UW/UDP` has a better throughput due to its smaller overhead, but this occurs at the cost of a rapidly decreasing PDR (see Fig. 3c). In contrast, UW/TP suffers from an increasingly larger delay and lower throughput to guarantee a perfect PDR.



(a) Throughput comparison.



(b) Delay comparison.



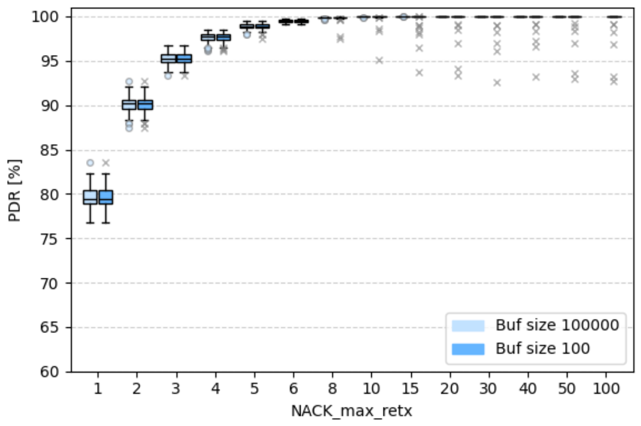
(c) PDR comparison.

Fig. 3: Scenario 1 results.

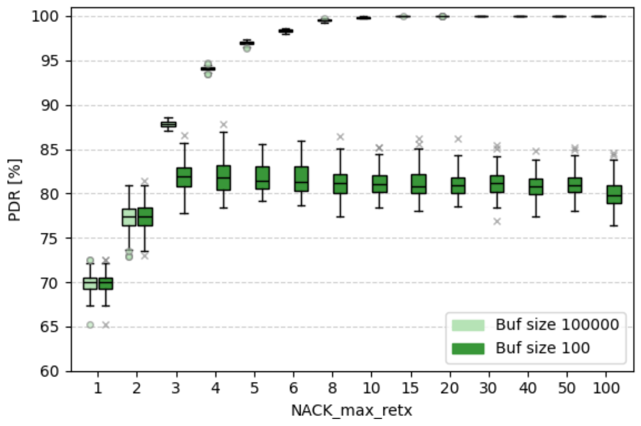
For bitrates greater than 225 bps we notice that some delay starts appearing, meaning that such rate is greater than what the dynamics of the protocol can bear. The next simulations are performed using two different rates of traffic generation, to study the behavior of the protocol below (200 bps, we call it 'normal' load) and above (250 bps, we call it 'high' load) the 225 bps threshold.

The second scenario examines the effect of the number of retransmissions on the performance of the protocol. Figure 4 and 5 show the results for normal and high load. In case of unlimited buffers, the PDR is increasing in the number of retransmissions, and also is the delay, even if with very low values. However, in case of limited buffers (with a capacity of 100 packets), the performance gets notably worse in the high load case, both in terms of PDR (Fig. 4b) and delay (Fig. 5b). This happens because the buffer eventually fills up, and packets need to be discarded; the receiver does not know this, so it keeps asking for their retransmission until the maximum number of NACK transmissions is reached. This slow dynamic of the receiver Performance is degraded so much that it is only slightly better than UW/UDP. However, for normal load, a reduced buffer size does not seem to particularly affect performance, except for some outlier cases (Fig. 4a).

Notably, in both cases a number of retransmissions that is too small (like 1 or 2) causes a low PDR independently from the buffer size.



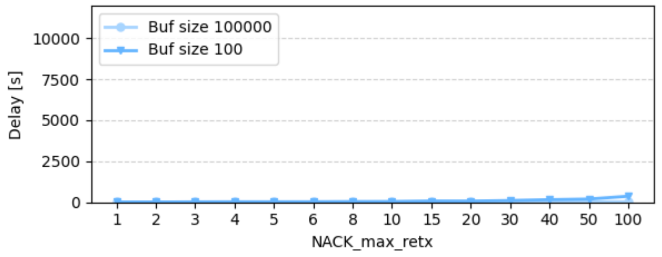
(a) Comparison for different buffer sizes in the normal load case.



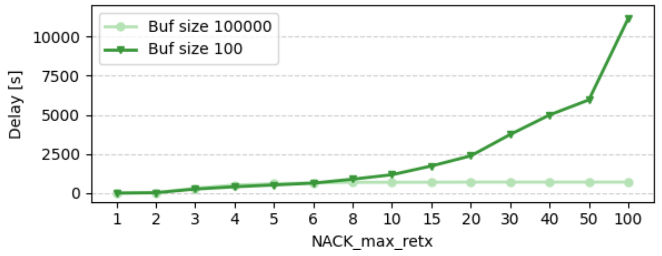
(b) Comparsion for different buffer sizes in the high load case.

Fig. 4: Scenario 2 results for PDR.

The last simulation showed that values between 6 and 8 produce very good results, so for the third scenario we fix *NACK_max_retx* to 7, and observe the impact of the time between retransmissions on the PDR. The value of *NACK_retx_time* does not seem to have a great effect in the high load case, as the plateau in Fig. 6a demonstrates, but instead it's really important in the normal load scenario. We can see in Fig. 6b that retransmission times between 4 and 8 seconds allow for an almost perfect PDR with a low delay, but values greater than 8 cause a worse PDR and a higher delay, because the slow retransmission causes the buffer to fill up and



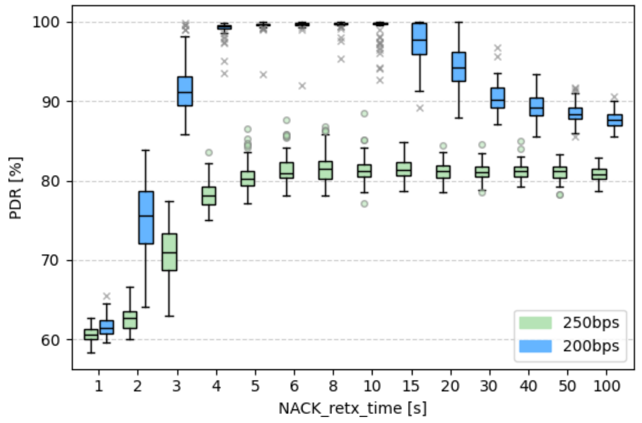
(a) *Comparison for different buffer sizes in the normal load case.*



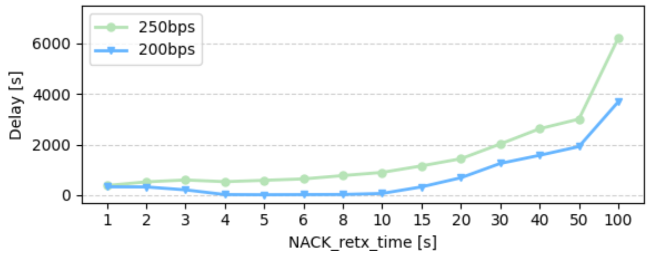
(b) *Comparison for different buffer sizes in the high load case.*

Fig. 5: Scenario 2 results for delay.

packets gets discarded. In both cases very low values seem to have a very negative impact on the performance.



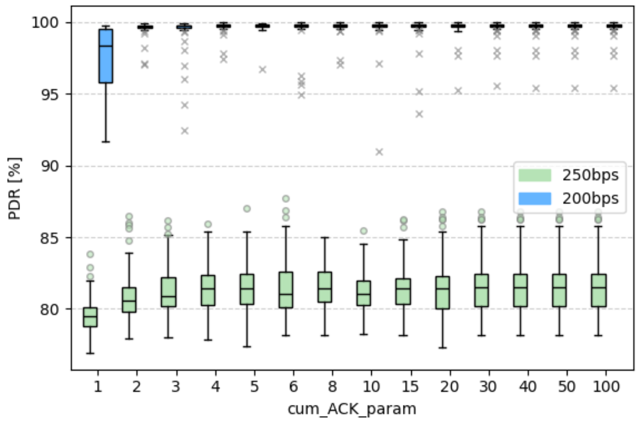
(a) *Results for PDR.*



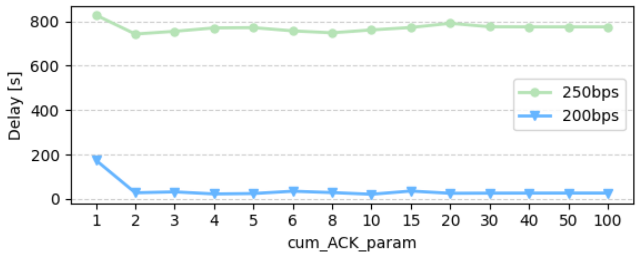
(b) *Results for delay.*

Fig. 6: Scenario 3 results. Comparison for different traffic generation rates: 200 bps and 250 bps.

The last scenario aims to understand if the frequency of cumulative ACKs impacts the performance of the protocol. However, as we can see from Figs. 7a and 7b, the parameter *cum_ACK_param* does not appear to have any strong effect on the communication, except when it has very low values; in these cases the performance gets worse, likely because we are flooding the network with not so useful packets.



(a) *Results for PDR.*



(b) *Results for delay.*

Fig. 7: Scenario 4. Comparison for different traffic generation rates: 200 bps and 250 bps.

VI. Conclusions and Future Work

In this work we analyzed the performance of a reliable transport layer protocol for underwater communication, with some soft parameters that can be modified by the user. The choice for the parameters is ruled by the PDR-delay tradeoff, and the flexibility of the protocol allows to adjust its performance based on the situation and preferences. The protocol behaves in two different ways depending on the generated traffic: if it is below a certain threshold, the protocol can achieve high PDR and low delay, even in case of limited buffers; in contrast, traffic above such threshold can only be withstood for limited periods, depending on the available buffer size. This makes the protocol resistant to bursts in the traffic, but it cannot be used in the long run for such higher rates. Moreover, the design makes ACKs not so relevant, but they are still needed to end the communication and to signal that the communication is alive in case of good channels that don't cause the transmission of many NACKs.

Future works can see an evolution of the protocol can autonomously probe the channel and adapt its parameters in case of time-varying channel. Moreover, under its actual conditions, the protocol does not implement a handshake system but needs a manual setup of both devices by an operator, which can be easy in the context of a simulator, but inconvenient in a real setting.

References

[1] SIGNET Group, "DESERT underwater," 2023. [Online]. Available: https://github.com/signetlabdei/desert_underwater.git, [Accessed: May 9, 2025].

[2] A. Pal, F. Campagnaro, K. Ashraf, R. Rahman, A. Ashok, and H. Guo, "Communication for underwater sensor networks: A comprehensive summary," *ACM Transactions on Sensor Networks*, vol. 19, 07 2022.