One- and Two-Way Travel Time Ranging in Underwater Acoustic Mobile Networks

Antonio Montanari, Filippo Campagnaro, Michele Zorzi
University of Padova, Department of Information Engineering (DEI)
Italy
{montanar,campagn1,zorzi}@dei.unipd.it

ABSTRACT

In the last ten years several simulation studies on Autonomous Underwater Vehicle swarm fleet formation have been performed, and some preliminary sea demonstrations of proof-of-concept prototypes were carried out. However, their actual realization is hindered by the difficulties of keeping track of the vehicles' positions due to the long latency required by traditional Two-way travel-time (TWTT) ranging measurements. One-way travel-time (OWTT) halves the latency, at the cost of a high precision oscillator such as an atomic clock or an oven controlled crystal oscillator, installed in the modem processing unit. In this paper we present two Medium Access Control (MAC) schemes for underwater acoustic mobile networks: a first MAC based on a time-division scheme with a high-precision oscillator to perform OWTT ranging, and a second MAC based on the token bus paradigm which, like regular TWTT ranging, does not require a high-precision oscillator. The main contributions of this work are the complete description of the ranging algorithms implementation, including MAC parameters and packet structure, and their realistic evaluation in both sparse and dense network scenarios with the DESERT Underwater Framework.

KEYWORDS

Underwater acoustic networks; one-way travel-time (OWTT); atomic clock; oven controlled crystal oscillators (OCXO), ranging in underwater networks.

ACM Reference Format:
Antonio Montanari, Filippo Campagnaro, Michele Zorzi. 2022. One- and Two-Way Travel Time Ranging in Underwater Acoustic Mobile Networks. In *The 16th International Conference on Underwater Networks & Systems (WUWNet'22), November 14–16, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3567600.3568156

WUWNet'22, November 14–16, 2022, Boston, MA, USA

ACM ISBN 978-1-4503-9952-4/22/11…$15.00
https://doi.org/10.1145/3567600.3568156

1 INTRODUCTION

The development of small low-cost Autonomous Underwater Vehicles (AUVs) [1] has increased the need for underwater acoustic communication and ranging to support swarm operation for collaborative data collection missions. Several military and civilian applications can be enabled by AUV swarms, including, but not limited to, coastal monitoring, mine countermeasure systems, and rapid environmental assessment [7]. Still, their main limitation is the difficulty of keeping the correct swarm formation, due to the limited amount of positioning information that can be transmitted per minute in the acoustic channel, characterized by low bitrate and long propagation delay [13]. Moreover, ranging operations require the time of flight measurement observing the round trip time, from the moment a node transmits a ranging request to the instant that node receives a reply. This process can last several seconds, as it takes at least twice the time of flight: given that the speed of sound is, on average, 1500 m/s, this means that if two nodes are 1.5 km apart the ranging operation lasts more than 2 s. One-Way Travel-Time (OWTT) halves the latency, as it requires a single packet transmission, but requires the use of precise crystal oscillators. In [11], for instance, the authors used Chip Scale Atomic Clock (CSAC) to perform OWTT and keep a precise measure for up to a few weeks of missions. Although CSAC are very accurate and have a very low power consumption, they are very expensive (>5 kEUR), making them not suitable for low-cost AUVs. In [10], instead, the authors performed OWTT with Oven-Controlled Crystal Oscillator (OCXO) during a sea trial, proving that the system can perform ranging measurements with similar errors as in two-way ranging. Although OCXOs are less precise and more power demanding than CSACs, they are definitely more affordable, making them a good candidate for low-cost swarm deployments. A complete comparison of precise clocks is presented in [8], where the trade-off between clock precision, power consumption, costs and mission duration is discussed.

Our work focuses on dense network scenarios, in particular we consider a swarm of N AUVs, were each vehicle is required to know not only the $N-1$ distances from itself to the others, but also all the distances between other vehicles, which can be represented in a symmetric distance matrix of dimensions N by $N-1$ with $D = \frac{N\cdot(N-1)}{2}$ distances. The knowledge of the full distance matrix gives each node a global cognition of the swarm and allows to run distributed algorithms without the propagation delay towards and from the controlling node that would occur in a centralized setup. Distributed algorithms that rely on distance information may be used for node positioning and navigation but may also aim at implementing efficient routing and Medium Access Control (MAC) protocols: taking regular Time Division Multiple Access (TDMA) as an example, it is known that the guard time between slots must be at least equal to the maximum propagation time between any node to ensure a collision-free behavior, but the application of this principle in underwater links characterized by very high propagation times drastically reduces the channel utilization and the throughput. Knowing distances between nodes, however, allows the implementation of protocols [12] that exploit node position diversity to schedule the TDMA slots preventing collisions while increasing the utilization of channel capacity. The purpose of this paper is therefore to present two ranging algorithms: one performing OWTT over TDMA between synchronized nodes and one which exploits node coordination in a token bus access scheme to perform ranging with minimal packet transmissions. Although works regarding both OWTT and TWTT schemes are already present in the literature [6], in this paper we cover some aspects which are often overlooked: we provide a complete description of the algorithms implementation including MAC parameters and packet structure. Moreover we evaluate the performances in both sparse and dense network scenarios and in two different channel models by means of simulations in the DESERT Underwater Framework [4] (an underwater network simulator developed and maintained by the SIGNET group at the University of Padova) and the results of the simulations are hereby included. The rest of the paper is organized as follows: Section 2 describes the difference between OWTT and Two-Way Travel-Time (TWTT) ranging, Section 3 presents in detail the proposed MAC ranging protocols, Section 4 illustrates the simulated scenarios and Section 5 analyzes simulation results, showing the effectiveness of the OWTT ranging system. Finally, Section 6 draws some conclusions.

2 TWTT AND OWTT RANGING

2.1 Two-Way Travel-Time Ranging

Let us consider a network of N nodes: in a naive TWTT scheme each node sends an Individual Interrogation Signal (IIS) and collects an Individual Reply Signal (IRS) from each of the other $N-1$ nodes, with a total number of $2\cdot N\cdot(N-1)$ exchanged messages (replies included) to complete the round. Since each reply must be unambiguously associated with the generating interrogation and not confused with concurrent replies from other nodes, both IIS and IRS must carry $2\log_2 N$ bits to encode sender and recipient node IDs. After a node has calculated the $N-1$ distances, it needs to broadcast them to the other nodes: being d_{bit} the number of bits used to encode a distance (16 bits would be fair for most situations, giving a fixed precision of 15 cm if the maximum range is 10 km), the sum of the transmission times of all nodes after a round will be

$$T_{txtot} \propto N\cdot(N-1)(2\cdot 2\log_2 N + d_{bit}). \quad (1)$$

A more efficient approach would be that each node broadcasts a unique Common Interrogation Signal (CIS) to which the other $N-1$ nodes reply with an IRS after a fixed delay $t_d(n)$, known a priori and different for each node in order to lower the chance of collisions: node i sending the CIS at time t_{tx} and receiving the IRS from node j at time t_{rx} will calculate the distance as

$$d_{ij} = \frac{t_{rx} - t_{tx} - t_d(i)}{2\cdot s}, \quad (2)$$

being s the approximate speed of sound in the water. Each node can share the information on calculated ranges by including them in the next round CIS so a total of $N\cdot(N-1)$ transmissions are needed while the sum of the transmission times will be

$$T_{txtot} \propto \underbrace{(N)\cdot(\log_2 N + (N-1)\cdot d_{bit})}_{\text{CIS}} + \underbrace{N\cdot(N-1)\cdot\log_2 N}_{\text{IRS}}. \quad (3)$$

2.2 One-Way Travel-Time Ranging

If nodes are synchronized, it does not make sense talking of interrogations and replies since each node periodically broadcasts a unique ranging signal that for sake of conciseness will be referred to as Common Ranging Signal (CRS) from now on. The CRS must include information on its time of transmission t_0 so a node j receiving a CRS from node i can calculate the distance as

$$d_{ij} = \frac{t_{rx} - t_{tx}}{s}. \quad (4)$$

It might be convenient to have slotted transmission times where each node is assigned a slot of a TDMA frame, so the CRS could encode t_{tx} with just $\log_2 N$ bits and distance calculated as

$$d_{ij} = \frac{t_{rx} \bmod T_f - t_{slot}(i)}{s}, \quad (5)$$

where T_f is the frame time and $t_{slot}(i)$ is the transmission start time of the slot assigned to node i relative to the beginning of the frame. Being based on a TDMA scheme, we will refer to this algorithm as `UwTDMARanging`. It is worth noting that T_f, which is also the period between subsequent CRS from the same node, must be greater than the maximum travel time between any two nodes of the swarm, otherwise any travel time $tt = x + k \cdot T_f$ would be aliased as $tt = x$ due to the modulo operation in Eq. (5). A round takes just N CRS transmissions and the sum of transmission times T_{txtot} will be

$$T_{txtot} \propto \underbrace{N \cdot \log_2 N}_{\text{ranging}} + \underbrace{N \cdot (N-1) \cdot d_{bit}}_{\text{distances diffusion}} . \tag{6}$$

2.3 Minimizing Ranging Transmissions in TWTT Ranging

An alternative way for ranging with minimum transmissions and without precision clocks can be achieved by noting that, conversely to the aforementioned TWTT method where the IIS and IRS are "individual" signals, they are actually overheard and might provide useful information to the other nodes in the swarm: any node k who hears the IIS from node i at time t_0 and then the IRS from node j at time t_1 can say that

$$d_{ij} + d_{jk} - d_{ik} = \frac{t_1 - t_0}{s}. \tag{7}$$

To exploit this information, we propose an algorithm where the nodes that forms a connected network access the channel according to a token bus based protocol where each node has a fixed preceding and successive node, thus forming a logical ring. Instead of sending individual signals, each node broadcasts a CRS as soon as it receives the CRS of the preceding node, as if the CRS signal were the tokens of a token bus network. Hence, we will refer to this scheme as `UwTokenbusRanging`. With this setup, each transmission from a node j gives the preceding node i the information $d_{ij} = \frac{t_{rx}(j) - t_{tx}(i)}{2 \cdot s}$ while the remaining $N-2$ nodes learn a difference of distances according to Eq. (7). If each node broadcasts all the information acquired on distances by embedding it in the CRS, at the end of each round a node will have $N-1$ equations by its own measurements and $(N-1)^2$ shared by the other nodes, resulting in an over-determined linear system of $N \cdot (N-1) = 2 \cdot D$ equations in D unknowns that can be solved by a least squares regression. As anticipated, the number of transmissions needed for each round is exactly N as in OWTT and the total transmission time is the same of Eq. (6).

3 PROTOCOLS IMPLEMENTATION

In this section we first present the `UwTokenBus` MAC protocol (Section 3.1) used as the base for the TWTT `UwTokenBusRanging` algorithm (Section 3.2). Then, we describe the OWTT `UwTDMARanging` (Section 3.3) protocol that extends a simple TDMA MAC scheme. `UwTokenBus`, UwTokenBusRanging and `UwTDMARanging` have been implemented into the DESERT Underwater Framework, while a TDMA MAC protocol (called `UwTDMA`) was already available in DESERT.

3.1 UwTokenBus MAC Protocol

The basic idea of token bus channel access is that all the nodes are connected through a common bus and are therefore able to overhear all the network activity. A special "token" packet is circulated and only the node currently holding the token is entitled to transmit. In our realization this idea is detailed as follows: each node is assigned a $nodeId \in \{0, ..., N-1\}$ and the token is a particular packet with a $tokenId$ field which gets incremented each time the token is passed on. For reasons that will become clear after illustrating the token regeneration mechanism, $tokenId$ values are not restricted to the $nodeId$ range, thus the token holder is the node for which holds $nodeId = tokenId \bmod N$. As soon as the outgoing packet queue is depleted or a maximum token hold time has elapsed, the node broadcasts the token with an incremented $tokenId$.

3.1.1 Token Regeneration. Since tokens can be lost or corrupted by errors, it is of paramount importance that regeneration strategies are in place to prevent a complete network freeze. Two scenarios are considered.

(1) If node i passes the token to j but the token is lost, i will consider j's inactivity as a failure to implicitly acknowledge token reception and after a timeout set by a `token_pass_timer` is expired, it will resend the token once. The timeout is set to the maximum TWTT between nodes $T_{tkpass} = 2 \cdot slot\_time$ where $slot\_time$ is a term borrowed from the ANSI/IEEE standard 802.4-1990[3] to indicate the maximum OWTT between any node of the network and is therefore proportional to the network geometric diameter.

(2) If the second attempt of passing the token also fails, or if the node holding the token stops working or gets isolated due to bad link conditions, the following node will autonomously regenerate the token after the timeout set by a `bus_idle_timer` is expired. The idea is that each node expects to receive the token after a maximum timeout proportional to the number of nodes between itself and the node currently holding the token. In fact every time a token is passed to a node j, each node k updates the timer according to

$$T_{busidle} = 3(n_{hop}(k, j) + 1) \cdot slot\_time, \tag{8}$$

where

$$n_{hop}(k, j) = (N + (k - j) \bmod N) \bmod N \tag{9}$$

returns the number of nodes between k and j in the logical ring. This timeout represents the upper limit for waiting the token reception since it considers the worst case scenario when each of the preceding token passings fails the first time.

3.1.2 Token Suppression. The deployment of a token regeneration mechanism implies the concurrent application of token suppression policies, in fact the token may be erroneously regenerated and shall be removed in order not to have multiple tokens circulating in the ring, that would hinder the collision-free behavior of the network. This happens if node j has correctly received the token from i and has already passed it on to k but i did not hear this implicit acknowledgment and as soon as its `token_pass_timer` expires, it resends the token to j. To prevent j from recirculating the second invalid token, each node keeps track of the last `tokenId` transmitted by itself or heard in the network, so it can discard all the tokens received with an older (lower) `tokenId`. The same principle is applied in case a node becomes deaf and is unable to hear any activity on the bus. In this case, as soon as its `bus_idle_timer` expires, it regenerates the token. However, due to the way the timeout of the timer is set, the regenerated token will reach the following node with an old `tokenId` and will be discarded.

3.2 UwTokenBusRanging

The `UwTokenBusRanging` class extends `UwTokenBus` by incorporating in the token packet additional information to perform the ranging:

(1) the `tt_vec` vector contains the $N-1$ Time Difference Of Arrivals (TDOA)s of the tokens measured by the sending node;
(2) the `token_resend` flag is used to distinguish the first token passing from a second attempt triggered by `token_pass_timer`;
(3) the `token_hold` field allows each node to withhold the token while notifying to the other nodes the time elapsed from token reception to token passing, and has a dual purpose: a minimum `min_token_hold` value is used to limit the traffic generated ranging packets while any greater value, up to a `max_token_hold` can be used to piggyback the ranging token onto regular data packets that might be present in the node outgoing queue.

A value of -1 in the `token_hold` field is used to alert that the token has been regenerated by `bus_idle_timer` expiration.

The token management has been slightly modified with respect to `UwTokenBus` since, beside ensuring correct regeneration and suppression, it must provide the tools to distinguish between resent and regenerated tokens and to assess whether token passing continuity is preserved, in order to calculate meaningful TDOAs as per Eq. (7). With regard to this aspect, the behavior of a node n passing a token to its successor $n+1$ is characterized as follows:

- node n passes the token to the following node by broadcasting a packet with $tokenId = tokenId + 1$;
- if the token is resent due to `token_pass_timer` expiration, the `token_resend` flag is set while $tokenId = tokenId + 1 + N$, thus introducing a gap of N in the `tokenId` progression which allows all the overhearing nodes to know if node $n+1$ passes on the first token or the resent one,
- if the token is resent due to `token_bus_timer` expiration, instead of using a second flag, the `token_hold` field is set to -1 to indicate that the receive time of this token will not be meaningful since the token transmission is not consequent to a token reception.

When node n overhears a token passing from node i to its successor j (may be $j = n$), n performs the following actions:

- always updates its matrix holding the travel times measured by all nodes (`tt_mat`) with the values carried by the `tt_vec` in the packet payload;
- checks if it has heard the previous token passing, that is if $tokenId = last\_heard\_tokenId{+}1$, then if `token_hold` is valid (non negative), it updates the corresponding element in `tt_mat` as $t_1 - t_0 - token\_hold$.

The travel times collected in `tt_mat` constitutes the vector of known terms **B** in the over-determined linear system $\mathbf{Ax} = \mathbf{B}$ where $\mathbf{x}$ is the vector of D unknown distances and **A** is the constant $2D \times D$ matrix of coefficients. The system's solution is calculated via non-negative least squares regression.

3.3 UwTDMARanging

This class extends the TDMA DESERT module (`UwTDMA`) and operates as follows: as soon as the assigned transmission slot begins, the node broadcasts the CRS packet, which consists of a `slotId` unsigned 16-bit integer field for identifying the transmission slot number, followed by an array of 16-bit floats of length $D = \frac{N\cdot(N-1)}{2}$ with the OWTT values measured by the node. Using a `slotId` field of $\log_2 N$ bits to enumerate the N slots of the TDMA frame would be sufficient but, according to the constraint highlighted after Eq. (5), limits the frequency of range updates. This limit can be easily overcome by using instead a larger number of bits ($nbits$) to incrementally enumerate the slots from 0 to $S_{max} = 2^{nbits} - 1$, allowing to recover the origin node as $n = slotId \bmod N$ and the OWTT between origin node i and receiving node j as

$$owtt_{ij} = t_{rx} \bmod T_{slot}(S_{max} + 1) - slotId \cdot T_{slot}. \qquad (10)$$

The OWTT measured upon reception of the packet, as well as the values carried in the payload are used to update the local vector holding the distances; a commutative function maps the tuples $(i, j), (j, i)$ to the same distance d_k so that the reception of either $owtt_{ij}$ or $owtt_{ji}$ updates the same element d_k. The elements to be sent in the payload by each node i are

$$distance\_vec[i, j], j \in \{0, ..., N-1\}, j \neq i, \tag{11}$$

and after the CRS packet has been sent, the control is returned to the parent class `UwTDMA` which starts transmitting data packets from the outgoing queue, if any, until the transmission slot expires.

4 SIMULATION SCENARIO AND SYSTEM SETTINGS

We tested the proposed protocols implemented in the DESERT Underwater framework with the following configuration:

- at time $t = 0$ s the nodes are equally spaced along a circle of 5 m diameter and start moving away radially from the initial position at a constant speed over ground of 1 m/s;
- at time $t = 300$ s the course is reversed until the nodes return to the starting position at $t = 600$ s;
- `UwTokenBus slot_time` is set according to the maximum network geometric diameter of 1200 m;
- the `guard_time` between TDMA slots is set to 0.25 s as lower values led to a high number of packet collisions in this setup as shown in Figure 4;
- the `frame_duration` is set so that each node has a `slot_time` 10% longer than the sum of `guard_time` plus the time needed for transmitting the ranging packet with a bitrate of 4800 bps and a Hamming 7/4 coding;
- the distances computed by the nodes are sampled every 5 seconds and compared to the true distances, assuming a fixed sound speed of 1500 m/s;
- the packet error probability due to interference is computed using the "meanpower" model present in DESERT.

Two different models are used to compute the error probability. First, the legacy DESERT physical layer, that implements the model in [13], is used to observe the results in an ideal channel. Then, a more realistic way to compute the error probability and the channel variability [5] is used to observe the protocol behavior in the case of a disruptive channel. In the second model, packet error and transition probabilities of the two-state Hidden Markov Model (HMM) are obtained using one of the acoustic links of the ASUNA dataset [9], a collection of data measures obtained during 14 experiments performed around Europe and Israel, freely available online [2]. The model considers two states, GOOD and BAD: in the former state the packet error rate is very low, while in the latter it is quite high. Specifically, using the procedure in [5], the following bit error rate (BER) and transition probabilities are obtained: $BER(GOOD) = 0.0051$, $BER(BAD) = 0.0193$, $P(GOOD \rightarrow BAD) = 0.053$, $P(BAD \rightarrow GOOD) = 0.192$. In order to model the fact that nearby nodes experience the same channel evolution, while nodes that are far away may have a different channel state, as long as the nodes are close, i.e., from time $t = 0$ s to $t = 200$ s and again from $t = 400$ s to $t = 600$ s, the same HMM is used to model the time evolution of all the links, while from $t = 200$ s to $t = 400$ s, a different HMM instance is used to model the state of each link. The state of each link is updated every second according to the transition probabilities.

5 SIMULATION RESULTS

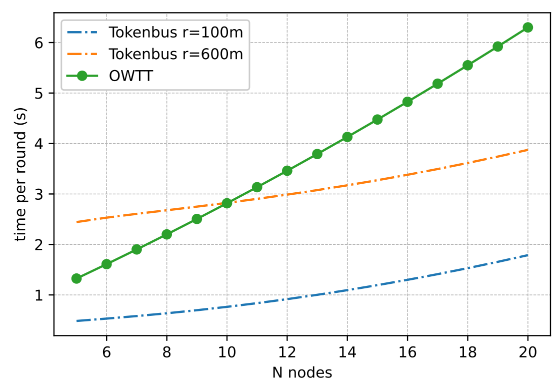


Figure 1: Time to complete a ranging round.

As performance metric we chose to represent the time series of the Root Mean Squared Error (RMSE) calculated for each node over the measured distances (Eq. (12)),

$$\text{RMSE(n,t)} = \sqrt{\frac{1}{D}\sum_{d=0}^{D-1}\left(\widehat{dist_{d,n}(t)} - dist_{d,n}(t)\right)^2}, \tag{12}$$

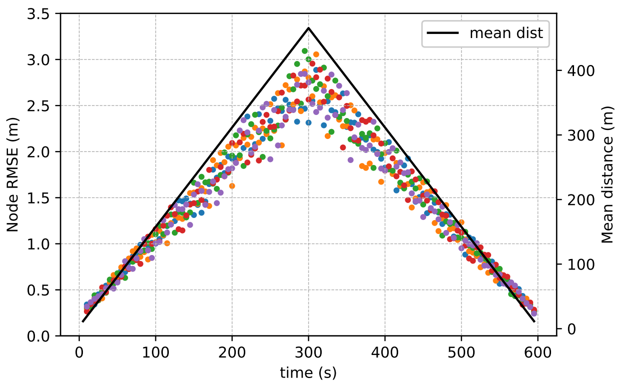
where $\widehat{dist_{d,n}(t)}$ is the distance d calculated by node n at time t and $dist_{d,n}(t)$ is the true distance value. Before discussing the simulation results in the ideal channel (Section 5.1) and realistic channel (Section 5.2), it is useful to point out that RMSE is strictly related to the frequency at which each distance is measured and disseminated to other nodes: as an indicator for this parameter we computed the time it takes to complete a ranging round (T_{round}), which is given by Eq. (13) and (14) for `UwTDMARanging` and `UwTokenbusRanging` respectively, and plotted in Figure 1.

$$T_{roundOWTT} = N \cdot \left(\frac{id_{bit} + d_{bit} \cdot (N-1)}{B} + T_{guard}\right) \tag{13}$$

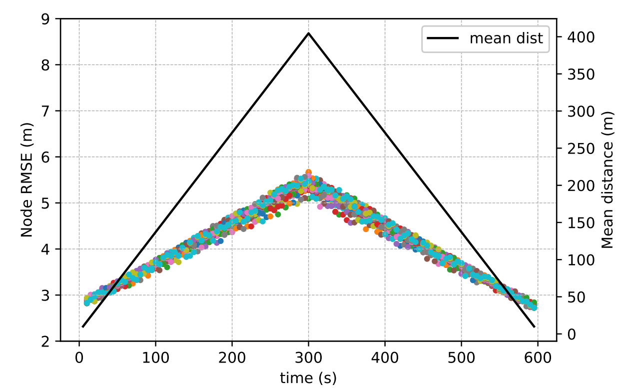
$$T_{roundTokenBus} = N \cdot \left(\frac{id_{bit} + d_{bit} \cdot N + 1}{B} + T_{hold}\right) + \frac{P}{s} \tag{14}$$

where id_{bit} has the value of 16 bits used for node/slot identification, $B = 4800$ bps is the modem bitrate, $T_{guard} = 0.25$ s is the TDMA guard time, $T_{hold} = 0.0001$ s is the minumum token hold time, s is the propagation speed of sound in water and P is the sum of all the distances between adjacent nodes in the logical ring, which in our scenario corresponds to the perimeter of the regular polygon inscribed in a circle of radius r : $P = 2 \cdot N \cdot r \cdot \sin(\frac{\pi}{N})$.

5.1 Results with an Ideal Channel



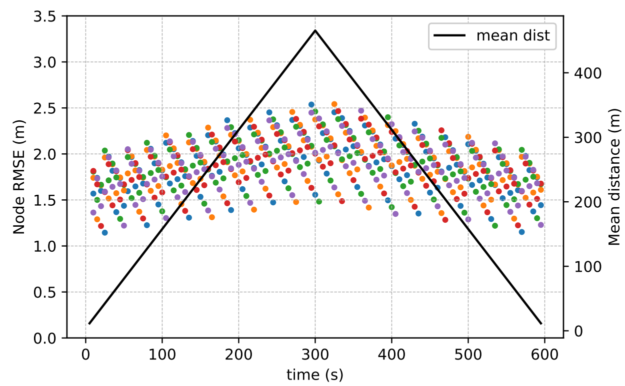
(a) 5 nodes.



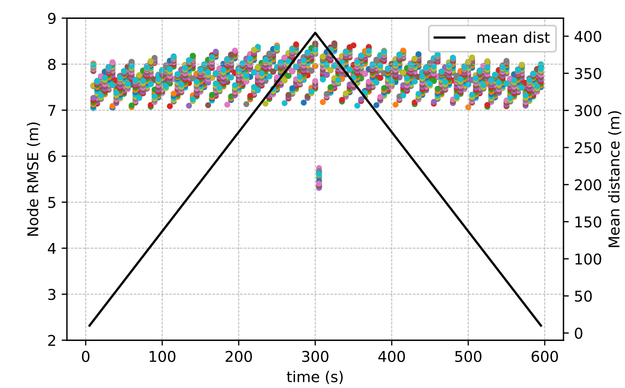
(b) 20 nodes.

Figure 2: `UwTokenBusRanging` RMSE in an error-free channel with 5 (Figure 2a) and 20 (Figure 2b) nodes.

Figures 2 and 3 present the RMSE for `UwTokenBusRanging` and `UwTDMARanging`, respectively. The black line indicates the mean distance between nodes at a certain simulation time, while the colored circles the RMSE experienced by a node at a certain simulation time. Results are presented for 5 (Figures 2a and 3a) and 20 (Figures 2b and 3b) nodes. The results for `UwTokenBusRanging`(Figure 2) indicate a linear proportionality between error and mean node distance as was expected according to Eq. (14), since the rate of range updates decreases as the time for the token to complete a round increases. The same plots for `UwTDMARanging` (Figure 3) show a little proportionality between RMSE and mean node distance since, although the update rate is constant and not distance-dependent, still the mean age of the information in each update increases with travel times. The fluctuating pattern is due to the beat between distance update period (equal to TDMA `frame_time`) and the time period to compute the RMSE (5 s). Comparing how the two ranging protocols scale with an increasing number of nodes we can note that the contribution of the term T_{guard} in Eq. (13), which is three orders of magnitude greater than the chosen minimum T_{hold} in Eq. (14), is responsible for the higher relative error increase between `UwTDMARanging` plots for 5 and 20 nodes, when compared to the corresponding relative increase for `UwTokenBusRanging`. While the proportionality of error with distance is an undesirable side effect for `UwTokenBusRanging` it is useful to remember that in order to prevent collisions between ranging packets we had to fix a 0.25 s `guard_time` parameter for `UwTDMARanging`, thus limiting its update rate and causing a higher RMSE when the nodes are close. Furthermore, while token bus ensures a collision-free behavior in any network geometry, this cannot



(a) 5 nodes.



(b) 20 nodes.

Figure 3: `UwTDMARanging` RMSE in an error-free channel with 5 (Figure 3a) and 20 (Figure 3b) nodes.

be guaranteed by a fixed `guard_time`, unless using a overly conservative value equal to the maximum propagation time. Reducing the `guard_time` parameter to 0.2 s leads to packet collisions as shown in Figure 4.

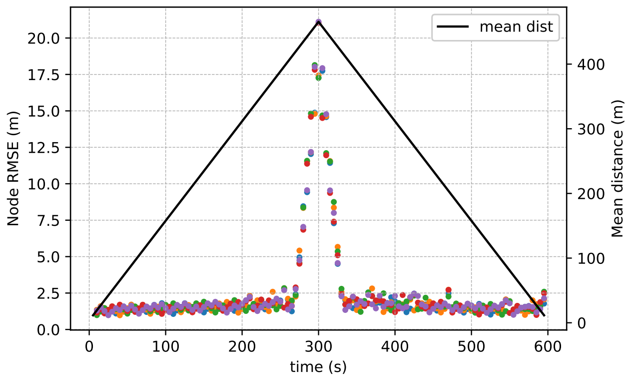


Figure 4: `UwTDMARanging` error due to packet collisions with 5 nodes.

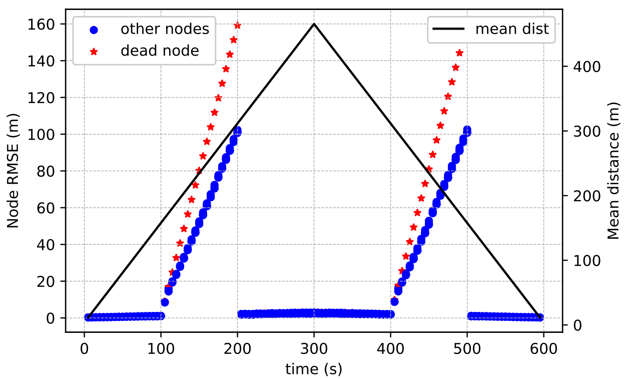


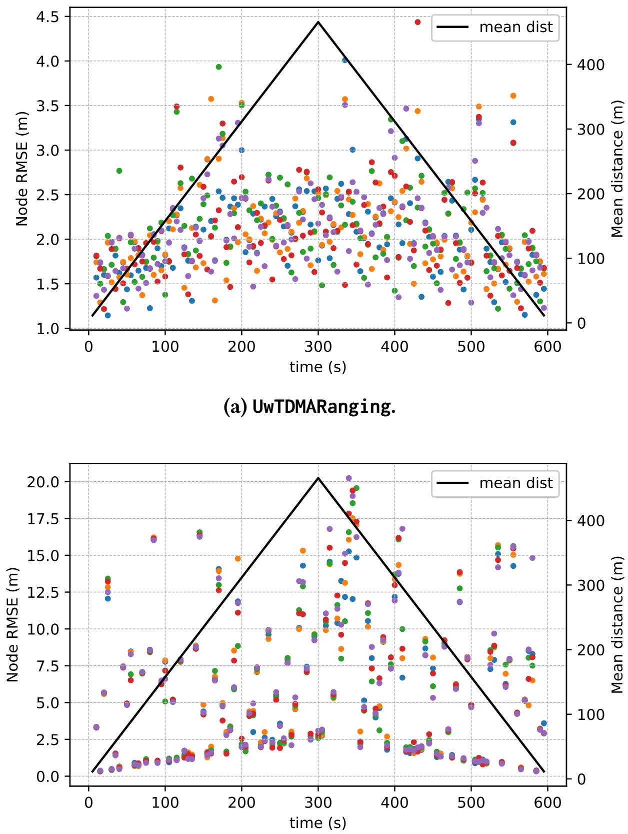
Figure 5: `UwTokenBusRanging` with 5 nodes, node 0 stops working in time intervals [100,200] and [400,500].

Although token bus ranging offers more flexibility since its performance is not affected by parameters such as the `guard_time`, it present a major weakness in case a node stops working: while the effects on `UwTDMARanging` are limited to the impossibility of updating the distances involving the dead node, the consequences on the token bus are more severe since the network freezes until token regeneration occurs, meanwhile preventing updates for all the nodes, as shown in Figure 5 where node number 0 stops working in time intervals [100,200] and [400,500].

Comparing the plots seen above, we can conclude that in the error-free channel, excluding the critical case of a node becoming unresponsive, `UwTokenBusRanging` can match or exceed the performances of `UwTDMARanging`, obtaining lower RMSE figures when nodes are within a limited range.

5.2 Realistic Channel

Contrary to the results obtained with the ideal channel, the simulations on the HMM channel exposed all the weakness of the token bus while `UwTDMARanging` was able to maintain similar performance. In fact we can see that, except for a few sparse outliers, `UwTDMARanging` error distribution in Figure 6a does not differ from Figure 3a and a consistent behavior can be found between Figures 3b and 7a. The RMSE

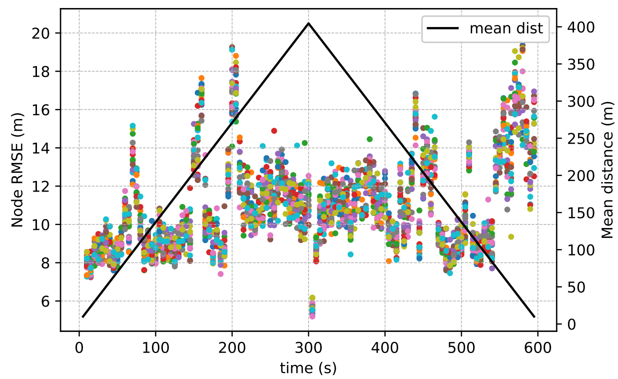


(a) `UwTDMARanging`.

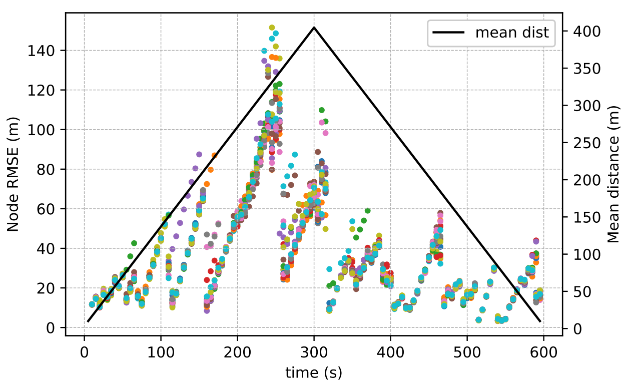
(b) `UwTokenBusRanging`.

Figure 6: Ranging with 5 nodes in the HMM channel.

plot for `UwTokenBusRanging` (Figure 6b) instead, presents distributed peaks in the error that reach considerable values and this is accentuated in Figure 7b with 20 nodes. Looking at the plots in Figures 6a and 7a we can also note how having independent HMM models for each link limits the error peaks between seconds 200 and 400 for the `UwTDMARanging`, while `UwTokenBusRanging` cannot take advantage of this since an error in a single link is sufficient to corrupt the token and freeze the network. In fact, although token bus is able to recover a token lost due to the noisy channel, this happens after a timeout (which is a multiple of the maximum signal travel time) is elapsed, thus allowing the build-up of large errors which make this ranging method not viable in real applications.



(a) `UwTDMARanging`.



(b) `UwTokenBusRanging`.

Figure 7: Ranging with 20 nodes in the HMM channel.

6 CONCLUSION AND FUTURE WORK

In this paper we presented two MAC protocols to perform OWTT and TWTT ranging, discussing advantages and disadvantages of both. The simulation results highlight that an OWTT ranging system based on a TDMA scheme performs satisfactorily in noisy channels, while a TWTT ranging system based on a token bus MAC outperforms OWTT in the case of an ideal channel, but can hardly be used in a realistic channel. Further improvements of `UwTDMARanging` may be performed by compressing the distance data shared in each packet, in order to reduce the payload size and allow for more robust channel coding schemes to be applied. Regarding `UwTokenBusRanging`, some attempts should be made to implement a more efficient token recovery and suppression mechanism to make it suitable for underwater networks.

ACKNOWLEDGMENTS

This work has been partially supported by the Italian General Directorate of Naval Armaments (NAVARM) and the Italian National Plan for Military Research (PNRM) (contract 20593), and by the European Union - FSE REACT EU, PON Research and Innovation 2014-2020 (DM 1062/2021).

REFERENCES

[1] [n. d.]. Advanced, small, low cost AUV technology. https://www.ecosub.uk/ . ([n. d.]). Last time accessed: Aug. 2022.

[2] [n. d.]. A Shared Underwater Network emulAtion dataset. https://sites.google.com/marsci.haifa.ac.il/asuna/. ([n. d.]). Last time accessed: Aug. 2022.

[3] 1990. IEEE Standard for Information processing systems – Local area networks –Part 4: Standard for Token-Passing Bus Access Method and Physical Layer Specifications. *ANSI/IEEE Std 802.4-1990 (Revision of ANSI/IEEE 802.4-1985) (Adopted by ISO/IEC and redesignated as ISO/IEC 8802-4: 1990)* (1990), 1–278. https://doi.org/10.1109/IEEESTD.1990.7229456

[4] Filippo Campagnaro, Roberto Francescon, Federico Guerra, Federico Favaro, Paolo Casari, Roee Diamant, and Michele Zorzi. 2016. The DESERT Underwater Framework v2: Improved Capabilities and Extension Tools. In *Proc. Ucomms*. Lerici, Italy.

[5] Filippo Campagnaro, Nicola Toffolo, and Michele Zorzi. 2022. Modeling Acoustic Channel Variability in Underwater Network Simulators from Real Field Experiment Data. *Electronics* 11, 14 (July 2022). https://doi.org/10.3390/electronics11142262

[6] Riccardo Costanzi, Davide Fenucci, Simone Giagnoni, Andrea Munafo, and Andrea Caiti. 2017. An Evaluation of Deep Water Navigation Systems for Autonomous Underwater Vehicles. *IFAC-PapersOnLine* 50 (07 2017), 13680–13685. https://doi.org/10.1016/j.ifacol.2017.08.2532

[7] Henry Dol. 2019. EDA-SALSA: Towards smart adaptive underwater acoustic networking. In *IEEE/MTS OCEANS 2019*. Marseille, France. https://doi.org/10.1109/OCEANSE.2019.8867361

[8] E. Coccolo, F. Campagnaro, *et al.* 2022. Underwater Acoustic Modem for a MOrphing Distributed Autonomous Underwater Vehicle (MODA). In *Proc. MTS/IEEE Oceans*. Chennai, India.

[9] P. Casari et al. 2021. ASUNA: A Topology Data Set for Underwater Network Emulation. *IEEE J. Oceanic Engineering* 46, 1 (Mar. 2021), 307–318.

[10] Gabriele Ferri, Roberto Petroccia, Tommaso Fabbri, Alessandro Faggiani, and Alessandra Tesei. 2021. A Network Navigation System With Opportunistic Use of One-Way and Two-Way Acoustic Ranging: the DANS20 Experience. In *IEEE/MTS OCEANS 2021*. San Diego – Porto Vitual Oceans. https://doi.org/10.23919/OCEANS44145.2021.9706011

[11] K.G. Kebkal, O.G. Kebkal, E. Glushko, V.K. Kebkal, L. Sebastião, A. Pascoal, J. Gomes, J. Ribeiro, H. Silva, M. Ribeiro, and G. Indivery. 2017. Underwater Acoustic Modems with Integrated Atomic Clocks for One-Way Travel-Time Underwater Vehicle Positioning. In *Proc. UACE*. Skiathos, Greece.

[12] K. Kredo II, P. Djukic, and P. Mohapatra. 2009. STUMP: Exploiting Position Diversity in the Staggered TDMA Underwater MAC Protocol. In *IEEE INFOCOM 2009*. 2961–2965. https://doi.org/10.1109/INFCOM.2009.5062267

[13] M. Stojanovic. 2007. On the Relationship Between Capacity and Distance in an Underwater Acoustic Communication Channel. *ACM SIGMOBILE Mobile Computing and Communications Review* 11, 4 (Oct. 2007), 34–43. https://doi.org/10.1145/1347364.1347373