UWGS: Geometry-Based Enhanced Time Synchronization in Underwater Acoustic Mobile Networks

Matin Ghalkhani
Department of Information Engineering
University of Padova
Padova, Italy
matin.ghalkhani@phd.unipd.it

Jianyu Zhang
Department of Information Engineering
University of Padova
Padova, Italy
jianyu.zhang@studenti.unipd.it

Filippo Campagnaro
Department of Information Engineering
University of Padova
Padova, Italy
campagn1@dei.unipd.it

Michele Zorzi
Wireless and More s.r.l.

Padova, Italy
michele.zorzi@wirelessandmore.it

***Abstract*—In underwater networks, precise time synchronization is critical for several functions, including sensor event time stamping, localization, and channel access. However, achieving synchronization is challenging due to the significant propagation delays and the mobility of nodes in underwater environments. Although methods like D-Sync [1], Mobi-Sync [2], and MU-Sync [3] address these issues in dynamic underwater settings, they typically rely on the estimation of radial speed from the Doppler effect. Furthermore, two-way message exchanges in these protocols result in more Signaling overhead or respond time.**

Building on recent advancements in one-way ranging techniques [4], this paper introduces a novel synchronization approach for mobile networks. Our method, Underwater Geometry-Based Synchronization (UWGS), uses a geometric approach to optimally estimate the clock skew and clock offset of unsynchronized nodes, and find the synchronization error. Instead of estimating radial speed through Doppler factors, our approach leverages multiple one-way pulse signals to construct the geometric relationship between node velocity and ranging. To evaluate its performance, we simulated both our proposed UWGS and the D-Sync protocol, comparing the results. The simulations demonstrate significant improvements in synchronization accuracy and efficiency with UWGS compared to the state-of-the-art D-Sync method.

***Index Terms*—Time Synchronization, Doppler-Based Time Correction, Geometry-Based Time Synchronization, Mobile Underwater Nodes, UWGS**

I. Introduction

In recent years, underwater sensor networks (UWSNs) have attracted significant attention from both academic and industrial sectors because of their potential in a range of underwater applications, including oceanographic data collection and environmental monitoring [5]. The synchronization of sensor nodes is crucial because discrepancies between local clocks can significantly degrade network performance. For instance, in applications such as mapping ocean currents using distributed networked sensors [6], asynchronous operation among sensor nodes increases the estimation error of current measurements, compromising data accuracy. Time synchronization can be categorized into two types: synchronizing all sensors to the Universal Time Controller (UTC) outside the network, or synchronizing them to a designated reference node clock within the network.

In underwater acoustic sensor networks, where operational schedules are meticulously structured, any misalignment in the nodes' clocks can lead to improper scheduling, disrupting network coordination. This misalignment stems from clock drift is a common issue with crystal oscillator-based hardware clocks that naturally deviate over time. Consequently, periodic synchronization is necessary to correct these drifts and ensure the proper functioning of the network.

Addressing these synchronization challenges in underwater environments presents unique difficulties. Although in terrestrial sensor networks, the propagation delay is negligible due to the high speed of electromagnetic waves, it becomes critical in UWSNs, where the relatively slow speed of acoustic signals makes propagation delay much more pronounced [7]. Such propagation delay, along with non-deterministic network behaviors—such as variable medium access times and interrupt handling delays—complicate the task of achieving precise time synchronization [8].

Effective network time synchronization relies on the exchange of messages between nodes to establish a unified sense of time. However, underwater environments introduce additional challenges like variable propagation delays, making

This work has been partially supported by the European Union - FSE REACT EU, PON Research and Innovation 2014-2020 (DM 1062/2021), and by the Italian National Recovery and Resilience Plan (NRRP) under NextGenerationEU, as part of the partnership on "Telecommunications of the Future" (PE0000001 - program "RESTART"). Additionally, EvoLogics GmbH, Berlin, Germany, is one of the sponsors of this project.

the achievement of high precision synchronization even more difficult. These challenges become even more complex, particularly when the nodes are in motion. For example, in real-world scenarios like underwater environments, the movement of nodes induces Doppler effects due to their varying speed. The Doppler effect has been extensively studied in previous works on underwater communication [2], [5], [9]–[11]. Lu et al. [1] highlighted this issue in the context of underwater time synchronization. Hence, for a more comprehensive analysis, alongside the high latency, the Doppler effect must also be taken into account.

In underwater communication systems, the speed of sound is typically five orders of magnitude lower than that of electromagnetic (EM) waves. The average velocity of Autonomous Underwater Vehicles (AUVs) and other mobile nodes in such environments is around 3 m/s, which is non-negligible compared to the speed of sound. Therefore, propagation delay and mobility introduce significant challenges in the synchronization process. Although simplified Doppler models exist, they only account for specific cases of node movement. In this paper, we propose a novel method that improves synchronization accuracy while also enhancing performance.

To address these issues and improve the functionality of existing synchronization protocols, we propose a new approach. This research was conducted in collaboration with EvoLogics GmbH, aiming to develop a time synchronization protocol specifically designed for underwater nodes, ensuring compatibility with EvoLogics' acoustic modems. While EvoLogics had previously achieved synchronization accuracy on the order of milliseconds, the goal of this collaboration was to refine the protocol to achieve microsecond-level synchronization. Another objective was to create a protocol that is more flexible and adaptable to other network stacks, providing a significant improvement over existing protocols in terms of response time and ease of integration.

The aim of this research was to enhance the synchronization process, achieving greater precision and reliability for underwater applications, while also reducing response time and increasing flexibility.

II. Related Works

TSHL [12] was the first synchronization method specifically designed to address the challenge of long propagation delays in underwater networks. It operates in two distinct phases: in the first phase, skew synchronization is achieved by performing linear regression on timing information from multiple beacon transmissions. In the second phase, the clock offset is corrected through a two-way message exchange. This approach assumes that the distance between nodes, and thus the propagation delay, remains constant throughout the skew estimation phase. However, this assumption of a static network often falls short in underwater environments, where node mobility is common.

To improve on this, MU-Sync [3] was developed, which estimates clock skew by performing linear regression twice on local timing data collected through a two-way message exchange with a cluster head. While this method offers some improvement, it requires a large number of two-way message exchanges, making it less energy-efficient than TSHL. Additionally, MU-Sync assumes that the one-way propagation delay can be approximated by the average round-trip time. This assumption introduces bias in underwater systems, especially those with even slight mobility, as varying distances between nodes distort the accuracy of the approximation over time.

Another limitation of MU-Sync arises from the network's growing size, as increased channel contention forces nodes to delay their transmissions randomly before responding to the cluster head. These cumulative delays worsen with network expansion, resulting in a noticeable degradation in MU-Sync's performance as the number of nodes increases.

D-Sync [1] was introduced to address the limitations of previous methods by providing more accurate synchronization for mobile underwater networks. In D-Sync, the process begins with a central beacon node broadcasting a request message to unsynchronized nodes, which then respond after random delays. These request-reply exchanges provide the necessary timing measurements to estimate both the clock offset and skew between the beacon and the nodes. Unlike previous protocols, D-Sync leverages Doppler shift measurements to compensate for changes in distance and relative speed, offering a solution tailored to the dynamic nature of mobile underwater systems.

One of the major challenges in synchronizing mobile nodes is the constantly changing distance between them, which complicates the synchronization process. D-Sync effectively addresses this issue by deriving equations that link clock skew and offset to changes in distance and relative velocity, using repeated message exchanges to update these measurements. A linear estimator is then applied to solve for the unknown parameters, ensuring that all nodes are synchronized to the beacon even in the presence of movement. This method avoids the assumption of fixed distances, which is crucial in underwater environments where mobility is prevalent.

The key distinction between MU-Sync and D-Sync lies in their handling of node mobility. While MU-Sync simplifies the synchronization process by assuming constant distances between nodes during message exchanges, this assumption limits its effectiveness in dynamic environments. In contrast, D-Sync's use of Doppler shift measurements allows it to account for changes in both distance and speed, significantly improving synchronization accuracy in mobile networks. By not relying on static distances, D-Sync provides better performance in scenarios where nodes are in constant motion [1].

III. Description of UWGS Method

UWGS employs one-way messaging, providing a more efficient approach to synchronization compared to traditional underwater time synchronization protocols. In particular, we will compare its performance against the D-Sync method, which relies on two-way communication. By eliminating the need for acknowledgments during the synchronization process, UWGS is expected to offer faster synchronization. Consider two nodes, as depicted in Figure 1: a reference node (denoted as *Ref*) and another potentially unsynchronized node, *Node*.

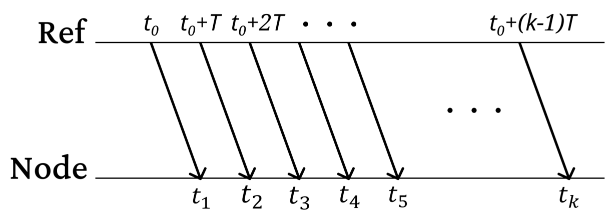


Fig. 1. Two-node synchronization setup with reference node *Ref* and moving node *Node*.

The synchronization procedure operates as follows: the *Ref* node transmits messages at times t_0, t_0+T, t_0+2T, $\ldots$, $t_0+(k-1)T$, which are received by the *Node* at times t_1, t_2, t_3, $\ldots$, t_k. In our scenario, T refers to the time interval between each message exchange.Since we are performing synchronization at the second layer (see Figure 2), we do not implement TDMA or other MAC protocols in this model, which allows us to manually set T. For clarity, while we can utilize the full protocol network stack -as they are relevant in real-world scenarios—we decided to focus on the layers related to synchronization and the physical layer. This simplification helps us address packet and header sizes when preparing the packets for transmission via an underwater channel.

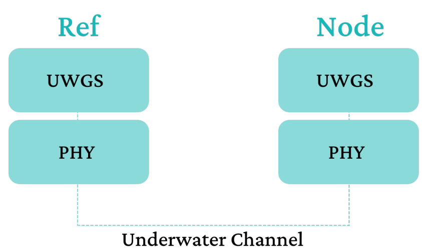


Fig. 2. Network model utilized in the proposed approach.

Figure 3 illustrates the configuration under the assumption that the *Ref* node remains stationary during the synchronization process, while the *Node* moves along the straight line. This assumption is justified by considering the *Node* to be moving at a constant velocity through its direction.

As the *Node* advances along a straight line, it sequentially receives the transmitted messages. For example, the first message sent at time t_0 is received at t_1, followed by the second message sent at t_0+T, which is received at t_2, and so on. This process continues until the final message is received at t_k. As will be discussed in detail later, a minimum of five messages is required to reliably estimate the parameters α and β.

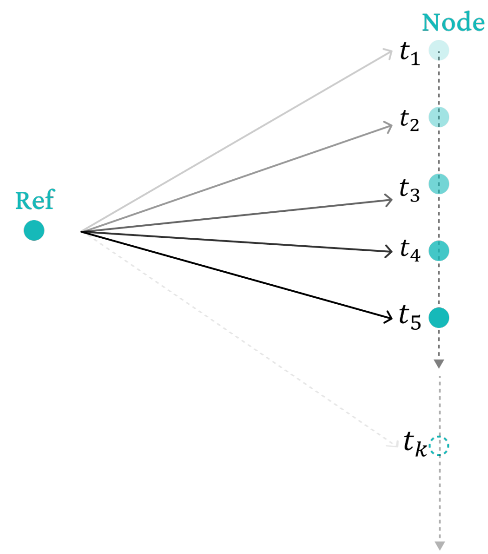


Fig. 3. Structure showing stationary reference node *Ref* and moving node *Node* along a straight line.

Figure 4 depicts the movement of the *Node* horizontally in part (a), while part (b) provides a simplified representation to enhance the clarity of the movement. We illustrated it this way to make it more understandable and similar to Stewart's theorem for the subsequent calculations.

To further analyze this movement and the synchronization process, we apply Stewart's geometric theorems. These theorems help us understand the spatial relationships and accuracy of synchronization, similar to the work done by Jianyu et al. [4] in underwater node synchronization.

In Figure 5, let a, b, and c represent the sides of a triangle, with a cevian d drawn to side a. If the cevian divides a into two segments of lengths m and n, Stewart's theorem is applied as follows:

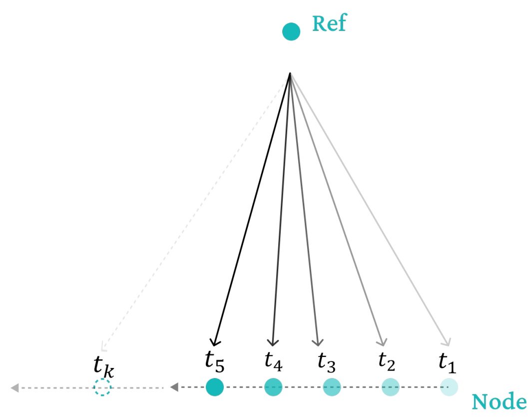
$$b^2m + c^2n = a(d^2 + mn) \tag{1}$$

Consider the first three packets, which are related to the receiving times t_1, t_2, and t_3.Consider that we are performing the synchronization process at the same depth underwater, allowing us to assume a constant sound speed of $c = 1500\,\text{m/s}$ for our calculations. Let p_1, p_2, and p_3 denote the distances along the propagation path for the first three packets:

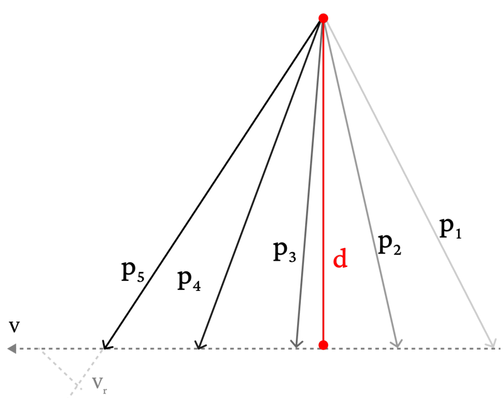
$$p_1 = (t_1 - t_0)c \tag{2}$$

$$p_2 = (t_2 - t_0 - T)c \tag{3}$$

$$p_3 = (t_3 - t_0 - 2T)c \tag{4}$$



(a)



(b)

Fig. 4. (a) *Node* movement shown in the horizontal. (b) Simplified representation of the *Node*'s movement.

As demonstrated in previous works on MU-Sync [3], and D-Sync [1], and also consider our definitions we can define the local *Node* time as follows:

$$t'_n = \alpha(t_n) + \beta \tag{5}$$

where α and β represent the clock skew and clock offset, respectively. Using Stewart's Theorem, the speed v of the mobile node can be determined:

$$\begin{aligned} &p_1^2\,(t_3 - t_2)\,v + p_3^2\,(t_2 - t_1)\,v = \\ &\left(p_2^2 + v^2\,(t_3 - t_2)\,(t_2 - t_1)\right)(t_3 - t_1)\,v \end{aligned} \tag{6}$$

Solving for v^2, we obtain:

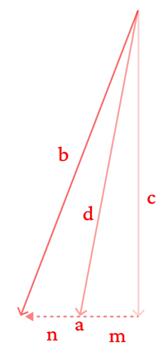


Fig. 5. Stewart's theorem.

$$\begin{aligned} v^2 =& \frac{p_1^2}{(t_3 - t_1)(t_2 - t_1)} + \frac{p_3^2}{(t_3 - t_1)(t_3 - t_2)} \\ &- \frac{p_2^2}{(t_3 - t_2)(t_2 - t_1)} \end{aligned} \tag{7}$$

Now, let's apply Stewart's theorem to the next two messages, which are related to the receiving times t_4 and t_5. If we define similarly:

$$p_4 = (t_4 - t_0 - 3T)\,c \tag{8}$$

$$p_5 = (t_5 - t_0 - 4T)\,c \tag{9}$$

We can also determine v^2 based on the received times at t_2, t_3, and t_4, while replacing p_2, p_3, and p_4.

$$\begin{aligned} v^2 =& \frac{p_2^2}{(t_4 - t_2)(t_3 - t_2)} + \frac{p_4^2}{(t_4 - t_2)(t_4 - t_3)} \\ &- \frac{p_3^2}{(t_4 - t_3)(t_3 - t_2)} \end{aligned} \tag{10}$$

We can also calculate v^2 based on the received times at t_3, t_4, and t_5, while replacing p_3, p_4, and p_5.

$$\begin{aligned} v^2 =& \frac{p_3^2}{(t_5 - t_3)(t_4 - t_3)} + \frac{p_5^2}{(t_5 - t_3)(t_5 - t_4)} \\ &- \frac{p_4^2}{(t_5 - t_4)(t_4 - t_3)} \end{aligned} \tag{11}$$

As we considered that the velocity is constant during the synchronization procedure, we can say that the left side of Equations (7), (10), and (11) are equal. So, we can combine and rewrite Equations (7) and (10) as a new equation which is (12), and also combination and rewriting of Equations (7) and (11) as a new equation which is (13) as follows:

$$\frac{p_1^2}{(t_3 - t_1)(t_2 - t_1)} + \frac{p_3^2}{(t_3 - t_1)(t_3 - t_2)} - \frac{p_2^2}{(t_3 - t_2)(t_2 - t_1)} = \frac{p_2^2}{(t_4 - t_2)(t_3 - t_2)} + \frac{p_4^2}{(t_4 - t_2)(t_4 - t_3)} - \frac{p_3^2}{(t_4 - t_3)(t_3 - t_2)} \tag{12}$$

$$\frac{p_1^2}{(t_3 - t_1)(t_2 - t_1)} + \frac{p_3^2}{(t_3 - t_1)(t_3 - t_2)} - \frac{p_2^2}{(t_3 - t_2)(t_2 - t_1)} = \frac{p_3^2}{(t_5 - t_3)(t_4 - t_3)} + \frac{p_5^2}{(t_5 - t_3)(t_5 - t_4)} - \frac{p_4^2}{(t_5 - t_4)(t_4 - t_3)} \tag{13}$$

For equation (12), the left side (LS_1) and right side (RS_1) are:

$$\mathrm{LS}_1 = \frac{p_1^2}{(t_3 - t_1)(t_2 - t_1)} + \frac{p_3^2}{(t_3 - t_1)(t_3 - t_2)} - \frac{p_2^2}{(t_3 - t_2)(t_2 - t_1)} \tag{14}$$

$$\mathrm{RS}_1 = \frac{p_2^2}{(t_4 - t_2)(t_3 - t_2)} + \frac{p_4^2}{(t_4 - t_2)(t_4 - t_3)} - \frac{p_3^2}{(t_4 - t_3)(t_3 - t_2)} \tag{15}$$

Similarly, for equation (13), the left side (LS_2) and right side (RS_2) are:

$$\mathrm{LS}_2 = \frac{p_1^2}{(t_3 - t_1)(t_2 - t_1)} + \frac{p_3^2}{(t_3 - t_1)(t_3 - t_2)} - \frac{p_2^2}{(t_3 - t_2)(t_2 - t_1)} \tag{16}$$

$$\mathrm{RS}_2 = \frac{p_3^2}{(t_5 - t_3)(t_4 - t_3)} + \frac{p_5^2}{(t_5 - t_3)(t_5 - t_4)} - \frac{p_4^2}{(t_5 - t_4)(t_4 - t_3)} \tag{17}$$

To form a system of equations for α and β, set $\mathrm{LS}_1 = \mathrm{RS}_1$ and $\mathrm{LS}_2 = \mathrm{RS}_2$:

$$\begin{cases} \mathrm{LS}_1 = \mathrm{RS}_1 \\ \mathrm{LS}_2 = \mathrm{RS}_2 \end{cases} \tag{18}$$

These equations can be represented as:

$$f_1(\alpha, \beta) = \mathrm{LS}_1 - \mathrm{RS}_1 = 0$$
$$f_2(\alpha, \beta) = \mathrm{LS}_2 - \mathrm{RS}_2 = 0$$

To solve the system for α and β, you can employ numerical methods such as Newton's method or use symbolic computation, provided that the values for t_1, t_2, t_3, t_4, t_5, and c are known.

IV. Results

The simulation was conducted using the following parameters:

TABLE I
SIMULATION PARAMETERS

Parameter	**Description**	**Value**
α	Uniformly distributed random number	$1\ \text{ppm} \leq \alpha \leq 100\ \text{ppm}$
β	Uniformly distributed random number	$1\ \text{s} \leq \beta \leq 5\ \text{s}$
Sound speed (c)	Speed of sound	1500 m/s
Node speed (V)	Movement speed of the node	4 m/s
Initial position	Coordinates of the starting position	$(X_0, Y_0) = (1000, 1000)$
Time slots (T)	Duration of each time slot	10 s
Backoff Timer (BT)	Duration of the backoff timer	5 s

After setting the parameters, we initiated a two-way ping-pong messaging style for the D-Sync method and a one-way messaging approach for UWGS. Each method was tested over 20 iterations to obtain reliable estimations.

Figure 6 illustrates that as the simulation time progresses, the node moves further from its initial position, resulting in increased synchronization error. We also calculated the mean synchronization error over all 20 iterations.

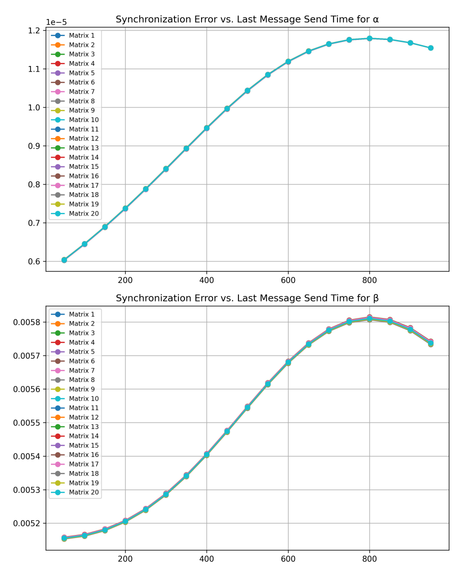


Fig. 6. D-Sync Synchronization Error vs Last Message Send Time

Figure 7 shows that an increase in the node's movement speed leads to higher synchronization error, thereby reducing the accuracy of the synchronization procedure. Additionally,

Figure 8 demonstrates that increasing the Backoff Timer (BT) results in greater synchronization error due to higher latency in the two-way messaging procedure of D-Sync.

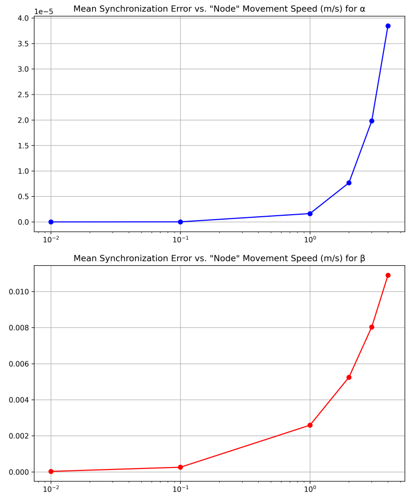


Fig. 7. D-Sync Average Synchronization Error vs *Node* Movement speed.

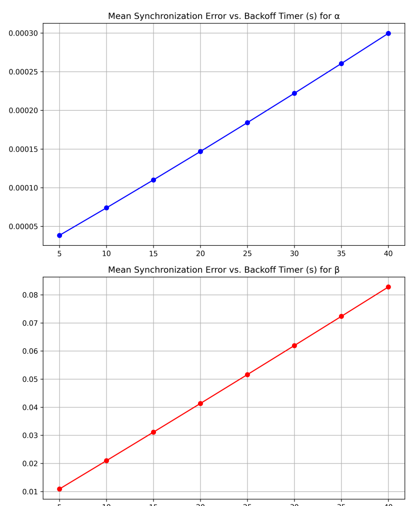


Fig. 8. D-Sync Average Synchronization Error vs BT

In contrast, UWGS exhibits improved synchronization accuracy as the node moves, as shown in Figure 10, compared to the D-Sync method (Figure 7). Furthermore, Figure 9 highlights the effect of time slot duration on synchronization accuracy. Increasing the time slot duration enhances the results; however, there is a limitation to this approach. It is essential to determine the optimal time slot duration (T) to ensure that the node maintains a constant speed within each interval. These time slots function similarly to TDMA time slots used in TDMA-based time synchronization protocols.

The simulation results indicate that the D-Sync method's synchronization accuracy deteriorates as node speed and backoff timer duration increase. Conversely, the UWGS method

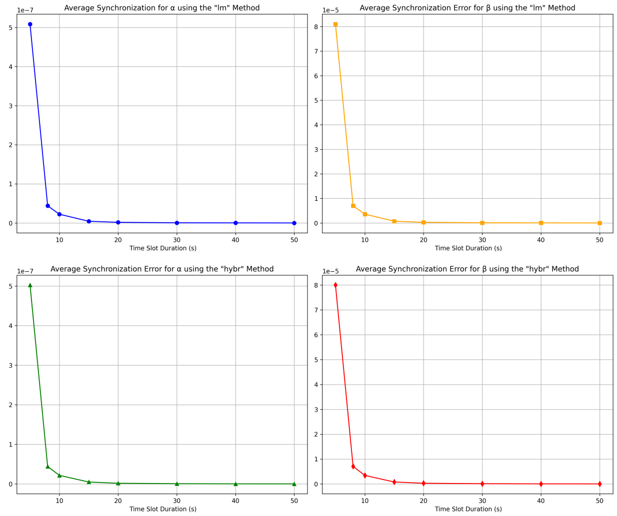


Fig. 9. Average Synchronization Error vs Time Slots in the proposed geometry-based approach.

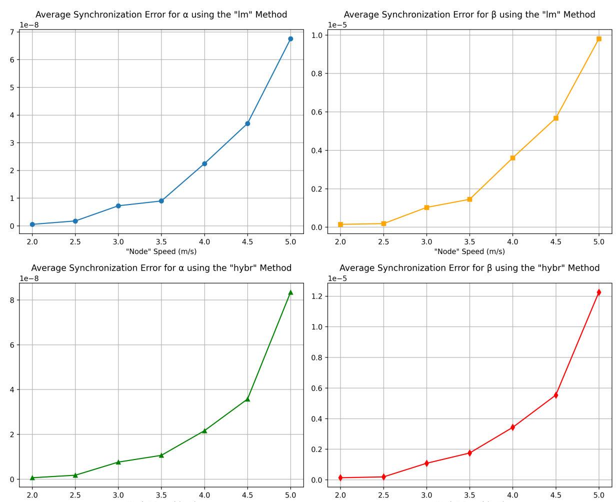


Fig. 10. Average Synchronization Error vs *Node* Movement in the proposed geometry-based approach.

demonstrates enhanced synchronization accuracy under similar conditions. Optimizing time slot duration is crucial for maintaining synchronization accuracy, particularly in dynamic environments where node speed varies.

V. Conclusion

By comparing the results of UWGS and D-Sync, we observed significant improvements in synchronization precision. In our study, we achieved microsecond-level synchronization, with the key advantage being that our method accomplishes this with only a one-way message exchange, unlike other protocols like D-Sync or MU-Sync that require two-way exchanges. Given these promising results, we plan to proceed with further steps, including simulations in the DESERT Underwater Simulation environment. Additionally, we aim to integrate other network layers to test UWGS's synchronization functionality across different layers. A real-world scenario is also being planned for future investigation, and this paper represents only the initial step in the development of this protocol.

REFERENCES

[1] F. Lu, D. Mirza, and C. Schurgers, "D-sync: Doppler-Based Time Synchronization for Mobile Underwater Sensor Networks," in *Proceedings of the 5th International Workshop on Underwater Networks*. Woods Hole, Massachusetts: Association for Computing Machinery, 2010.

[2] B. Li, S. Zhou, M. Stojanovic, L. Freitag, and P. Willett, "Multicarrier communication over underwater acoustic channels with nonuniform Doppler shifts," *IEEE Journal of Oceanic Engineering*, vol. 33, no. 2, pp. 198–209, Apr 2008.

[3] N. Chirdchoo, W.-S. Soh, and K. Chua, "MU-Sync: A Time Synchronization Protocol for Underwater Mobile Networks," in *Proceedings of the Annual International Conference on Mobile Computing and Networking (MOBICOM)*, 2008, pp. 35–42.

[4] J. Zhang, F. Campagnaro, A. Montanari, and M. Zorzi, "One-way Ranging for Mobile Underwater Acoustic Networks with Long Interaction Periods," in *Proceedings of the IEEE/MTS Oceans*. Singapore: IEEE, 2024, contribution in conference proceedings.

[5] "Editorial underwater acoustic communications: Where we stand and what is next?" *IEEE Journal of Oceanic Engineering*, vol. 44, no. 1, pp. 1–6, 2019.

[6] C.-F. Huang, T. Yang, J.-Y. Liu, and J. Schindall, "Acoustic Mapping of Ocean Currents Using Networked Distributed Sensors," *The Journal of the Acoustical Society of America*, vol. 134, no. 3, pp. 2090–2105, 2013.

[7] S. Kim and Y. Yoo, "SMP-Sync: Time Synchronization Using Seawater Movement Pattern for Underwater Wireless Networks," in *2014 International Conference on Computing, Networking and Communications (ICNC)*, Honolulu, HI, USA, 2014, pp. 842–846.

[8] F. Sivrikaya and B. Yener, "Time Synchronization in Sensor Networks: A Survey," *IEEE Network*, vol. 18, pp. 45–50, 2004.

[9] B. Sharif, J. Neasham, O. Hinton, and A. Adams, "A computationally efficient Doppler compensation system for underwater acoustic communications," *IEEE Journal of Oceanic Engineering*, vol. 25, no. 1, pp. 52–61, Jan 2000.

[10] N. Parrish, S. Roy, and P. Arabshahi, "Symbol by symbol Doppler rate estimation for highly mobile underwater OFDM," in *WUWNet '09: Proceedings of the Fourth ACM International Workshop on UnderWater Networks*. New York, NY, USA: ACM, 2009, pp. 1–8.

[11] S. F. Mason, C. R. Berger, S. Zhou, and P. Willett, "Detection, synchronization, and Doppler scale estimation with multicarrier waveforms in underwater acoustic communication," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 9, pp. 1638–1649, 2008.

[12] A. Syed and J. Heidemann, "Time Synchronization for High Latency Acoustic Networks," in *Proceedings of the IEEE Infocom*. Barcelona, Spain: IEEE, 2006, page to appear.