Simulation-based analysis of clock synchronization for underwater wireless sensor networks

Erik Diaz*, *Victor* Ramos, and *Rafael* Aguilar-Gonzalez

Universidad Autónoma Metropolitana, WiNetSys Team, 09340 Iztapalapa, Mexico City, Mexico

Abstract. Clock synchronization in underwater wireless sensor networks (UWSN) is a key procedure for the operation of this type of networks. Besides, due to the characteristics of the underwater channel, UWSNs face challenging constraints that make them different from other ad hoc wireless networks such as limited bandwidth, high error rates, high propagation delay, just to mention a few. Most of the existing literature on clock synchronization for UWSNs has been done through mathematical models. Thus, in this paper, we present a simulation-based approach to Light-Sync, which is a protocol synchronizing sensor nodes placed on the sea to a beacon on the surface. We use NS-2 along with AquaSim to study a UWSN with Light-Sync and prove its operation by means of discrete-event simulation. Moreover, our results are consistent with those found in Light-Sync and match accurately the synchronization accuracy regardless the number of nodes in the UWSN. Thus, we provide a simulation-based validation approach that makes more robust the proposal in Light-Sync.

1 Introduction

Underwater sensor networks are deployed to support a great variety of applications in the oceans such as temperature monitoring, disaster prevention, water salinity and water oxygen levels, to mention a few. Such applications are much more challenging to develop than traditional sensor networks since underwater communications present characteristics that make quite difficult the transmission in the channel. Traditional wireless sensor networks use radio or optical channels as the communications medium; thus, transmissions use the electromagnetic spectrum [1]. On the other hand, the downside of using the electromagnetic spectrum for UWSNs is that water greatly affects communications since it disperses the electromagnetic frequencies. Therefore, for underwater communications acoustic waves are preferred. Using acoustic waves makes communication possible in UWSNs, but this does not mean it is easy. In fact, several phenomena impact the quality of underwater communications. This is because the propagation speed of acoustic signals is about 1500 m/s, which clearly limits the maximum throughput of UWSNs.

Nowadays, communications in underwater networks are very useful for the range of applications cited above. An underwater wireless sensor network is commonly composed of

* Corresponding author: migueldyaz@xanum.uam.mx

© The Authors, published by EDP Sciences. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (http://creativecommons.org/licenses/by/4.0/).

a network of sensors deployed in the ocean sending data to a beacon placed at the surface. Figure 1 shows the general architecture for a UWSN. The sensor network relays data through multihop paths from the seabed to a station or a buoy, which acts as a supernode at the surface. Such a supernode may be equipped with an acoustic transducer, allowing multiple communications through surface towers and satellites.

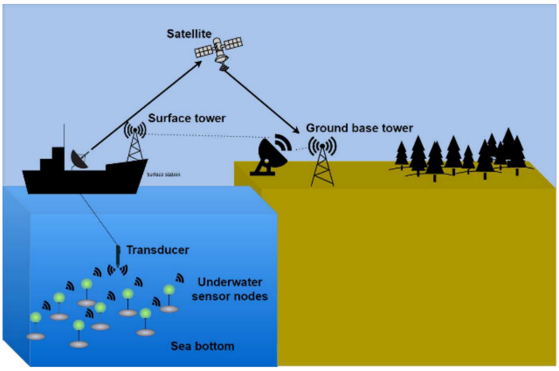

Fig. 1 General architecture of an underwater wireless sensor network (UWSN).

Often, for long time periods, the network topology is considered static, allowing the network to provide connectivity in this way. Nevertheless, such connectivity may change due to the small scale of ocean currents. With the aim of avoiding and reducing the impact of the characteristics of the acoustic channel, clock synchronization has been widely investigated in the last decade. Clocks in a UWSN lose synchronization mainly because of skew and drift. Most of the existing literature on clock synchronization for UWSNs considers the issues of high loss rates as well as the large propagation delays. The solutions provided are mostly mathematical models implemented as algorithms. On the other hand, an approach that has been less used is the use of a discrete-event simulator. Using a simulator has the advantage of considering the impact of a most complete set of phenomena rather than a mathematical model since in the latter several assumptions must be made to simplify analysis. Thus, in this paper by means of simulation with NS-2 and AquaSim, we evaluate the performance of Light-Sync, which is a clock synchronization protocol for UWSNs. In Section 2, we provide a brief review of the state-of-the art on synchronization in UWSNs; Section 3 describes the simulation of Light-Sync, a protocol to synchronize clocks in UWSNs. Section 4, presents the results obtained from validating Light-Sync by computing the time it takes the nodes to synchronize with respect to their distance to the buoy as well as the minimum-square error of the number of iterations. Finally, we sketch our conclusions in the last section.

2 State-of-the-art on synchronization in UWSNs

Work on synchronization for UWSNs mainly focuses on the low reliability of the underwater communications channel as well as on the characteristics of the network, i.e., a static or mobile topology. A series of exchanges of round-trip packets, followed by a linear regression is implemented to estimate clock skew and drift.

2.1 MU-Sync

The goal of MU-Sync is to compute clock skew and drift through two approximation phases [2]. The first one is called *skew and offset acquisition* phase, where sensor clocks are estimated with a two-step linear regression on a set of reference beacons, which are transmitted by a cluster that responds to the neighbor nodes. During the second phase, called *synchronization* phase, the cluster transmits clock skew and drift values of all neighbors. The approach followed by MU-Sync is by means of a cluster on a master-slave setup. However, the limiting factor in this protocol is the large message exchange between each node and the cluster, which requires several beacons, resulting in a message overload. Besides, it is assumed that the propagation delay is estimated with the average round-trip time, which causes additional errors.

2.2 D-Sync

In D-Sync, a cluster starts the synchronization process by transmitting a reference packet to which all the neighbors respond [3]. Through an estimation of the Doppler effect, the propagation delay is inferred by applying least squares estimators for the neighbor clock nodes with respect to the cluster, reducing in this way the estimation error. Like MU-Sync, D-Sync uses a master-slave approach; however, to get better efficiency in energy consumption, the approach is more complex which makes it even more sensitive to packet loss.

2.3 Mobi-Sync

Mobi-Sync considers networks composed of three different types of nodes: buoys at the surface equipped with GPS, supernodes operating with reference clocks always synchronized with the buoys, and low-complexity and low-cost nodes [4]. Mobi-Sync creates a spatial correlation between the mobile nodes to estimate the propagation delay using a model to infer the speed of ordinary nodes as a function of the speed of the supernodes. During the packet exchange phase, a node transmits a message to request clock synchronization. Each supernode neighbor schedules two response messages containing the speed vector and the timestamp of the MAC layer. The packet exchange phase ends when a node has collected enough data for a linear regression with a preset precision. After a fixed time interval, t_{r1}, each supernode transmits its first response packet trusting each supernode is located at a different distance with respect to the reference node to avoid collisions. In the interval t_{r2}, the supernodes transmit a second response packet. On the other hand, accounting with a more precise synchronization implies having a larger message exchange, which results in a higher energy consumption, which is undesirable.

2.4 Light-Sync

Light-Sync assumes the network is static since the nodes are anchored to the seabed. Thus, through a gateway it accurately gets the time by means of a GPS receiver [5]. Light-Sync focuses mainly on reducing energy consumption. It is executed in two phases, the first one consists of a series of full-duplex packet exchanges between a buoy P and a node A, which is the closest neighbor. The time between transmission and reception, ε, over P is fixed for node A. The first phase ends when node A has accurately estimated the clock parameters with a node B located within the communications range of node A, but not within that of P.

2.5 TSHL

TSHL (Time Synchronization for High Latency) is a clock synchronization algorithm for long-range and short-range networks [6]. It is executed in two phases; in the first one, each node computes clock bias through a linear regression from multiple beacons. During the second phase, clock drift is computed through a full-duplex packet exchange with a MAC timestamp. TSHL assumes a constant propagation delay, thus nodes must be stationary. The multihop network is formed of sensor nodes attached to buoys to platforms or boats.

3. Simulation of Light-Sync

As mentioned above, there is little work on evaluating clock synchronization protocols for UWSNs by means of simulation. Using a discrete-event simulator has the advantage of considering a wider range of phenomena at the physical and MAC layers since it makes no simplifying assumptions as mathematical models do. Moreover, providing a simulation-based approach validated with a mathematical model, provides a strongest approach for a protocol. Thus, in this paper we evaluate Light-Sync with the NS-2 simulator along with Aqua-Sim, which is an extension for UWSNs. Aqua-Sim allows to simulate the attenuation of the underwater acoustic channel, packet collisions caused by long delays, and also it is able to support three-dimensional simulation setups.

Table 1 lists the parameters used in the simulations of our scenario. We have executed 100 simulations for each initial position of the nodes. Thus, we deploy 30 underwater nodes randomly distributed not exceeding a volume of 100 × 100 × 100 m^3. Also, we put a buoy node at the center of the area placed at the surface (zero depth). During our simulations, each of the ordinary nodes is able to directly communicate with the buoy through a previous message exchange to synchronize, while the rest wait their turn to do the same operation

Table 1 Simulation parameters

Parameter	**Value**
Area	100 × 100 m
Depth	100 m
Transmission power	0.75 W
Receiving power	0.2 W
Propagation speed	1500 m/s
Routing protocol	VBF (Vector-based forwarding)
Simulation time	2000 s
Number of nodes	31
Maximum data packet size	40 bytes
Minimum data packet size	5 bytes
Coverage range	130 m

During the simulation, the propagation speed is kept constant. Each simulation campaign lasts 2000 seconds. After sending data, each node is able to pass to a waiting state saving energy in this way. The simulation scenario is illustrated in Figure 2.

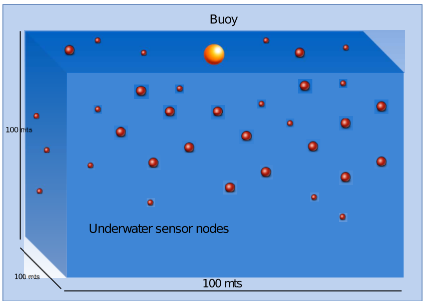

Fig. 2 Simulation setup for our Light-Sync evaluation.

4 Results

We execute campaigns of 200 simulations in order to get accurate results. The parameter of interest is the synchronization time of each of the 30 nodes distributed along the configurated region with respect to the buoy node at the surface. The synchronization procedure in Light-Sync starts when one of the nodes, mainly the closest to the surface, broadcasts a synchronization request message to the buoy, which contains a timestamp, $T_A(t_{i,1})$. Then, at time t_{i2}, has received the message. Thus, after a time ε, the buoy responds to the node with a message carrying timestamps t_{i2} and $t_{i,3}$ of reception and transmission. To end this first part, the node receives such a message at time $T_A(t_{i,4})$. Such message exchange between the node and the buoy is repeated I times until an accurate estimation of the clock skew and drift is obtained respect to the buoy.

Later on, the nodes that are furthest away from the buoy but inside its coverage range will repeat the same process. Then, the second phase of Light-Sync operates on the nodes outside the buoy coverage range but inside the coverage range of a node that has already estimated its clock values. Thus, we show in Figure 3 the linear regression we compute to confirm that the further the node is from the buoy, the largest its synchronization delay respect to it.

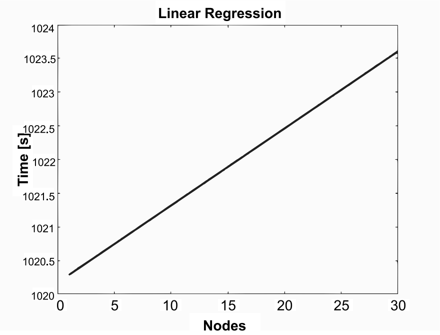

Fig. 3 Linear regression for clock synchronization in our scenario.

Finally, we compute the mean squared error (MSE) to analyze the exchange of messages between nodes and the buoy. Clock estimation is independent of the number of iterations, I, and the number of nodes in the network because each node is considered separately. Thus, Figure 4 shows the MSE computed from the obtained values during the communication between the buoy (P) and the closest node (A). We find that the estimators for clock skew and drift will be very efficient.

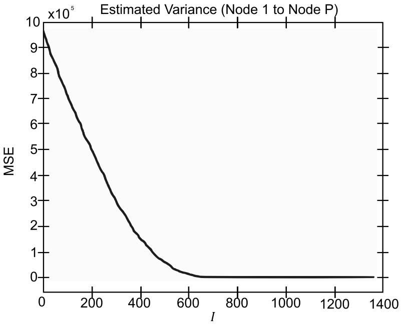

Fig. 4 MSE estimation versus the number of iterations, *I*.

5 Conclusions

We have evaluated, by means of simulation with NS-2 and Aqua-Sim the performance of Light-Sync, a well-known clock synchronization protocol for UWSNs. Most of the existing literature does not consider a simulation approach for evaluation. This is quite important, since such an approach does not take into account over simplifying assumptions often made by closed-form models. Moreover, providing a simulation-based evaluation offers a more complete and robust approach. Thus, we have validated the results presented in [5], which are consistent with those found with the analytical model provided. Our results match accurately those of Light-Sync about the synchronization accuracy, regardless of the number of nodes in the UWSN. The importance of validating synchronization algorithms by means of simulation is clearly important, as also shown in [7].

References

1. Fattah, S.; Gani, A.; Ahmedy, I.; Idris, M.Y.I.; Targio Hashem, I.A. A Survey on Underwater Wireless Sensor Networks: Requirements, Taxonomy, Recent Advances, and Open Research Challenges. Sensors (2020), 20, 5393. https://doi.org/10.3390/s20185393.
2. Chirdchoo, N., Soh, W.-S., & Chua, K. C. (2008). MU-Sync: A Time Synchronization Protocol for Underwater Mobile Networks. ACM International Workshop on Underwater Networks, 35–42. https://doi.org/10.1145/1410107.1410115
3. Lu, F., Mirza, D., & Schurgers, C. (2010). D-Sync: Doppler-Based Time Synchronization for Mobile Underwater Sensor Networks. ACM International Workshop on UnderWater Networks. https://doi.org/10.1145/1410107.141011510.1145/1868812.1868815
4. Liu, J., Zhou, Z., Peng, Z., Cui, J.-H., Zuba, M., & Fiondella, L. (2013). Mobi-Sync: Efficient Time Synchronization for Mobile Underwater Sensor Networks. IEEE ToPDS, 24(2), 406–416. https://doi.org/10.1145/1410107.141011510.1109/TPDS.2012.71
5. Zennaro, D., Tomasi, B., Vangelista, L., & Zorzi, M. (2012). Light-Sync: A low overhead synchronization algorithm for underwater acoustic networks, Oceans. https://doi.org/10.1109/OCEANS-Yeosu.2012.6263491
6. Syed, A. A., & Heidemann, J. (2006). Time Synchronization for High Latency Acoustic Networks. Proc. of the IEEE Infocom, 1–12. https://doi.org/10.1145/1410107.141011510.1109/INFOCOM.2006.161
7. Karim, S., Shaikh, F.K., Aurangzeb, K., Chowdry, B.S., Alhussein, M. Anchor nodes assisted cluster-based routing protocol for reliable data transfer in underwater sensor networks. IEEE Access (2021). https://doi.org/10.1109/ACCESS.2021.3063295