Time Based Synchronization Using PCO-RBSYNC Algorithm in Wireless Sensor Networks

K. Nithya[1] ,Dr. R.Kousalya [2]

[1] *Research Scholar, Department of Computer Science, Dr.NGP Arts and Science College and Assistant Professor, KG College of Arts and Science*

[2] *Professor and Head Department of Computer Applications, Dr.NGP Arts and Science College,*

***Abstract*:-**since synchronization is required for several applications, time (or clock) synchronization is a vast and important area of study. Distributed databases, real-time communication, and distributed data gathering are a few examples of possible uses. Using the Pulse-Coupled Oscillator with Reference Broadcast Synchronization technique (PCO-RBSync), this research presents a new method for time-based synchronization in WSNs. Initializing the clocks, establishing a reference source, and establishing sensor nodes are all part of the first phase of network setup. The Pulse-Coupled Oscillator model enables gradual alignment of internal oscillations through pulse exchanges among neighboring nodes. Reference Broadcast Synchronization periodically broadcasts local timestamps from designated nodes to synchronize other nodes, which calculate time differences and adjust their clocks for synchronization. Continuous error monitoring, compensation techniques, and algorithm fine-tuning mitigate discrepancies, optimizing accuracy and stability. Thorough testing and evaluation validate the PCO-RBSync algorithm's effectiveness in achieving precise time synchronization in WSNs, offering a robust solution for applications such as environmental monitoring, industrial automation, and healthcare.

Keywords: Clock synchronization, Network initialization, Time synchronization, Synchronization approach, Wireless Sensor Networks

1. Introduction

Nodes need to agree on the same time in order for the network to perform a variety of critical functions. Event time stamping, MAC, distributed data aggregation, and localization are all part of this [1]. But, as time passes, the nodes' local clocks drift intrinsically, causing them to get out of sync. All distributed systems rely on time synchronization techniques [2]. With communication occurring at a much coarser time granularity in underwater networks, the accuracy of time synchronization is significantly lower than in terrestrial networks. Nevertheless, accurate time synchronization can reduce the need for frequent re-synchronizations, which in turn saves energy [3-5]. Due to the extreme power constraints of underwater equipment, accurate time-synchronization with little signaling is thereby of paramount relevance for underwater networks [6].

The establishment of a shared understanding of time between nodes is the foundation of all network time synchronization techniques [7]. Nevertheless, synchronization becomes more difficult due to the non-determinism in network dynamics, which includes medium access time, propagation time, and interrupts handling time [8]. Underwater sensor networks cannot employ conventional time synchronization techniques due to the unknown and very lengthy propagation delays of sound waves [9, 10]. Uncertainty in time between two synchronization nodes results from not estimating the propagation delay. When nodes are movable, this problem gets even worse [11]. Nodes can't be properly synchronized using measurements taken from time-stamped messages anymore due to the fact that the propagation latency increases with time. A small number of time-sync protocols have been developed with underwater networks in mind; however these are better suited to networks where nodes can move around very much [12–13].

From industrial automation to environmental monitoring, Wireless Sensor Networks (WSNs) have become essential components in many applications. Precise temporal synchronization among the dispersed sensor nodes is crucial to the performance of these networks [14]. Time-based synchronization guarantees dependable communication, coordinated data collecting, and effective energy consumption. When considering how to achieve accurate time synchronization in WSNs, the PCO-RBSYNC algorithm stands out as a potential answer [15]. One method for ensuring that resource-constrained sensor nodes maintain a consistent time is known as Phase-Coupled Oscillator-Reference Broadcast Synchronization, or PCO-RBSYNC [16]. Utilizing the ideas of phase-coupled oscillators and reference broadcasts, this method synchronizes the local clocks of sensor nodes without requiring high computational resources [17]. An essential aspect of wireless sensor networks is time synchronization, which is thoroughly examined in this study as it pertains to the PCO-RBSYNC approach. We want to investigate the unique features and benefits of this synchronization approach in WSNs in order to give a comprehensive description of its application [18]. By delving into the complexities of time-based synchronization with the PCO-RBSYNC algorithm, we hope to shed light on synchronization techniques in WSNs and provide a foundation for more efficient and reliable applications in many domains [19-22].

The main contribution of the paper is:

- Network Initialization
- Pulse-Coupled Oscillator (PCO) Model
- Reference Broadcast Synchronization
- Time Synchronization
- Error Compensation and Fine-tuning

The following is the outline for the rest of the paper. Different time-based synchronization solutions are discussed by several writers in Section 2. On display in Section 3 is the suggested model. The findings of the study are briefly reviewed in Section 4. Section 5 wraps up by talking about the outcome and what's next.

1.1. Motivation of the paper

Our research presents a ground-breaking method for accomplishing exact time synchronization in WSNs, acknowledging the critical role of synchronization as an essential prerequisite for these applications. The inherent problems in WSNs are addressed by the proposed solution, which is the Pulse-Coupled Oscillator with Reference Broadcast Synchronization algorithm (PCO-RBSync). The motivation for this novel approach lies in its ability to seamlessly initialize networks, synchronize sensor nodes, and maintain accurate and stable time across the entire system.

2. Background study

Barolli, L., et al. [1] To gain a thorough understanding of the book's content, one should start by examining the overview provided in the table of contents, preface, and introduction.Gu, X., et al [5] To comprehensively grasp the content, initiating the study with a thorough review of the abstract, introduction, and conclusion is essential. These sections provide insights into the research problem and the primary outcomes of the study.Pan, X., et al. [8] these authors' research problem and significant findings are outlined in the abstract, introduction, and conclusion. The study can be better understood by delving into the authors' backgrounds, techniques, results, and references.Sun, X., et al [13]these authors research problem and significant findings are outlined in the abstract, introduction, and conclusion.Wang, H., et al. [15] Outlining the research topic and findings are key sections—the abstract, introduction, and conclusion. To improve knowledge, one should look into the writers' backgrounds, learn about the methodology, and analyze the outcomes.Wu, T.-Y., et al. [17] an enhancement method for wireless body sensor networks' MAC synchronization driven by heartbeats is suggested in the work of Wu and Wang. The research challenge and contributions are outlined in the abstract, introduction, and conclusion, three key elements of a research paper. Yang, Y., et al. [20] Emphasizing the research challenge and contributions are key sections—the abstract, introduction, and conclusion. To improve knowledge, one should look into the writers' backgrounds, learn about the methodology, and analyze the outcomes.

2.1. Problem definition

The Pulse-Coupled Oscillator (PCO) algorithm, while innovative, can face limitations in achieving precise synchronization in certain scenarios. One potential disadvantage is its reliance on pulse exchanges among neighboring nodes, which can introduce synchronization errors and latency, particularly in large-scale networks or environments with high levels of interference or node mobility.

III. Materials and methods

Here, we detail the methods that went into developing and testing PCO-RBSync, an algorithm for time synchronization in WSNs, Pulse-Coupled Oscillator with Reference Broadcast Synchronization. The materials cover the parts that need to be there, like sensor nodes, and the methods explain how to set up the network, how to use the Pulse-Coupled Oscillator model, how to synchronize the broadcasts, how to monitor errors continuously, how to fix them, and how to fine-tune the algorithms.

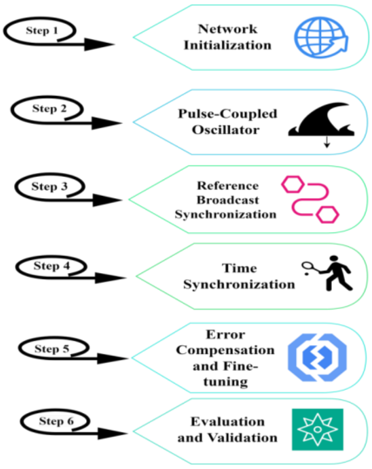


Figure 1: Proposed workflow architecture

3.1. Network model

The Pulse-Coupled Oscillator (PCO) model is employed to facilitate gradual alignment of internal oscillations among neighboring nodes in a wireless sensor network (WSN). The model is represented by the following equation:

$$\theta_i(t+1) = \theta_i(t) + w_i + \sum_{j=1}^{N} A_{ij} \, . sin\big(\theta_j(t) - \theta_i(t) + \emptyset_{ij}\big) \text{------------ (1)}$$

Here,

- $\theta_i(t)$ is the phase of oscillator ii at time t,
- w_i is the natural frequency of oscillator i,
- A_{ij} represents the coupling strength between oscillators ii and j,
- $\emptyset_{ij}$ is the phase difference between oscillators ii and j,
- N is the total number of oscillators in the network.

Reference Broadcast Synchronization is used for periodic broadcasting of local timestamps from designated nodes to synchronize other nodes. The algorithm adjusts the clocks of receiving nodes based on the time differences calculated from these broadcasts.

$S_i(t) = T_{local}(t) - T_{broadcast}(t)$ ------------ (2)

Where,

- $S_i(t)$ is the time difference for node ii at time t,
- $T_{local}(t)$ is the local timestamp of node i at time t,
- $T_{broadcast}(t)$ is the broadcasted timestamp.

B. Clock model

A linear clock model is introduced briefly in this section. Think of a WSN with N sensors. A hardware clock is included on every sensor node i, and its current reading is provided by

$T_i(t) = a_i t + b_i$------------ (3)

Where a_i is the clock speed-determining hardware clock skew and $b_i$is the clock offset-determining hardware clock. Due to factors such as ambient temperature, defective crystal oscillators, battery power, and oscillator aging, the hardware clock skews can vary slightly from node to node, as mentioned in. Additionally, we presume that the skew $a_i$of each hardware clock meets

$1 - p \leq a_i \leq 1 + p$-------------- (4)

Other hardware components in WSNs can rely on a constantly operating hardware clock, thus it's known that its value cannot be manually modified. As a result, the synchronized time is typically represented by a logical clock. The value of the logical clock, in particular, is directly proportional to the hardware clock.

$L_i(t) = a_i t_i(t) + b_i = a_i . a_i t + a_i b_i + b_i$ ------------- (5)

It is usual practice to initialize each node with a constant value, allowing them to broadcast their messages at regular intervals with a period determined by their hardware clock.

C. Pulse-Coupled Oscillator (PCO)

Here we provide a concise description of PCO for a number of classical model oscillators that have been published referred by Chen Z et al. (2023). The famous integrate-and-fire paradigm is the first example:

$\frac{dv}{db} = -V + I$---------------- (6)

Where $V(t^+) = 0$ and$V(t^-) = 1$. According to our presumption, this has a period of oscillation since$I > 1$.

$T = -1n \frac{I-1}{I}$ ---------------- (7)

We query when the oscillator will fire next after adding an amount a to V at $t = t_s$. The perturbation a will cause V to rise beyond 1, and the oscillator will fire instantly, resulting in$\hat{T}(t_s) = t_s$, if ts is near enough to $t = T$. On the other hand, a simple calculation reveals that

$\hat{T}(t_s) = -1n \frac{I-1}{I-ae^{t_s}}$-------------- (8)

The Phase-Response Curve(PRC) for the integrate-and-fire model for $I = 1.05$ and different values. When$a > 0$, the PRC violates the requirement that$\Delta(0) = 0$, and when a< 0, it violates both the constraint that $\Delta(0) = 0$ and the condition that $\Delta(1) = 0$. There is also an iteration of the integrate-and-fire paradigm known as the "quadratic" model:

$\frac{dx}{dt} = I + x^2$------------- (9)

Where firing is defined $as x(t) \to infinity$, and x is reset to infinity thereafter. Time is just

$T = \int_{-\infty}^{\infty} \frac{dx}{I+x^2} = \frac{\pi}{\sqrt{I}}$----------- (10)

The usual shape of a limit cycle close to a saddle-node gives birth to this model. As mentioned earlier, we will suppose that a perturbation of magnitude an provided at time $t = t_s$. Next, a simple computation reveals that

$\hat{T}(t_s) = t_s + \frac{1}{\sqrt{I}}(\frac{\pi}{2} - \arctan[\frac{a}{\sqrt{I}} - \cot(\sqrt{I}(t_s))])$-------------- (11)

The PCO disappears at both 0 and T in this model, in contrast to the integrate-and-fire model, and this is true independent of the perturbation strength. Because of this, it serves as a "better" substitute for a genuine neuron.

The classic radial isochron clock is a two-dimensional system that, like all systems approaching a Hopf bifurcation, has a simplified normal form for our last example:

$\frac{dr}{dt} = \wedge r(1 - r^2),\ \frac{d\theta}{dt} = 1$------------- (12)

Since θ is an element of S1, it represents the oscillator's polar form. Disruptions are characterized as changes in the x-coordinate of length a, whereas firing is defined as $\theta = 0$. In order to quickly return perturbations to the limit cycle along radii, it is assumed that the parameter Γ is quite big. For the term "phase" to be defined, the absolute value of a must be less than 1. For the same reason as before, $(x, y) = (cost_s, sint_s)$ in the plane, where t_s is the time after firing when the pulse is provided. The following coordinates are after the pulse: $(a + cost_s, sints)$. The trigonometric method yields the alternative angle, θ:

Thus

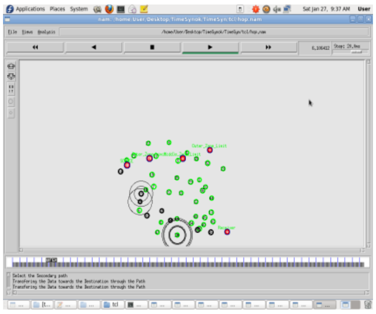
$\hat{T}(t_s) = t_s + 2\pi - \arctan\left(\frac{\sin t_s}{a + \cos t_s}\right)$----------- (13)

For each given value of a, the PRC can advance or postpone the firing, unlike the other two types. The PCO becomes a solitary value as $a \rightarrow 1$. The PRC disappears at $t = 0$, regardless of the value of a, so keep that in mind. A very basic variant of the radial isochron clock PCO exists for tiny values of a.

3.2 Reference Broadcast Synchronization

Regularly, nodes in an RBS scheme will use the physical-layer broadcast of the network to communicate with their neighbors via beacon messages referred by H. Kim et al. (2018). The time of arrival of the message serves as a benchmark for the receivers to compare their own clocks. The precise time of transmission is irrelevant, and the message does not include a timestamp. Time required for reference packet reception and processing by either node is the primary determinant of RBS accuracy.

Based on RBS's analysis, the key routes send and Access times are no longer problematic. We do not include LANs or ad hoc networks with a range of tens of feet in our micro-scale error budget because their propagation times are only tens of nanoseconds at most. Hence, RBS basically thinks the propagation time is zero. By utilizing the broadcast channel to synchronize receivers, the most significant sources of nondeterministic delay on the essential route can be eliminated.



Select the Secondary path
Transfering the Data towards the Destination through the Path
Transfering the Data towards the Destination through the Path

Figure 2: Reference Broadcast Synchronization

The sender is part of the conventional system's critical route, as shown in the top figure. Since RBS is a method of synchronization between receivers, it eliminates the need for the sender to be on the critical route. With the use of an event timeline, the general synchronization technique for dispersed camera networks employing reference broadcast is presented in Figure 2.

3.3. Time Synchronization

Time synchronization in mesh topologies presents challenges due to the dynamic nature of the network topology, multi-hop communication, and variable propagation delays. Achieving precise synchronization in mesh networks requires efficient routing protocols, adaptive synchronization algorithms, and robust error correction mechanisms to compensate for network dynamics and environmental factors affecting communication.

In flat grid topologies, time synchronization can be relatively simpler compared to mesh topologies due to the regular and predictable communication patterns. All of the camera nodes' frame synchronization was examined by measuring the event time of reference signal reception and picture capture. At each j^{th} cycle, the distribution of t_{rj} and t_{cj}in determines frame synchronization. Here are several descriptive statistics numbers in Eq. (14) that show the synchronization accuracy.

$$\mathrm{d}_{\mathrm{cj}} = \{\mathrm{d}_{\mathrm{cj}} | \mathrm{d}_{\mathrm{cj}} = \mathrm{t} - \mathrm{t}_{\mathrm{sj}}, \forall \mathrm{t}_{\mathrm{cj}}\}\text{-------------- (14)}$$

$$\sigma_{\mathrm{cj}} = \sqrt{\mathrm{E}\left(\mathrm{d}_{\mathrm{cj}} - \mathrm{E}(\mathrm{d}_{\mathrm{cj}})\right)^2}\text{-------------- (15)}$$

All of the camera nodes' frame synchronization was examined by measuring the event time of reference signal reception and picture capture. At each j^{th} cycle, the distribution of t_{rj} and t_{cj}. Here are several descriptive statistics numbers in Eq. (15) that show the synchronization accuracy:

$$\sigma_{\mathrm{c}} = \{\sigma_{\mathrm{c1}}, \dots, \sigma_{\mathrm{cM}}\} = \cup_{\mathrm{j}=1}^{\mathrm{M}} \sigma_{\mathrm{cj}}\text{------------- (16)}$$

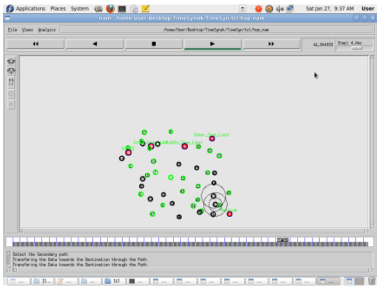
Where M is the number of cycles that have passed since the measurement of time began. We can understand the frame synchronization by looking at the distribution $\mathrm{D}(\sigma_{\mathrm{c}})$

The distribution$\mathrm{D}(\sigma_{\mathrm{r}})$, which describes the reference receiving at each camera node in terms of time, is similarly illustrated in Eq. (17-19).

$$\mathrm{d}_{\mathrm{rj}} = \{\mathrm{d}_{\mathrm{rj}} | \mathrm{d}_{\mathrm{rj}} = \mathrm{t} - \mathrm{t}_{\mathrm{sj}},\} \forall \mathrm{t}_{\mathrm{rj}}\text{--------- (17)}$$

$$\sigma_{\mathrm{rj}} = \sqrt{\mathrm{E}\left(\mathrm{d}_{\mathrm{rj}} - \mathrm{E}(\mathrm{d}_{\mathrm{rj}})\right)^2}\text{--------------- (18)}$$

$$\sigma_{\mathrm{r}} = \{\sigma_{\mathrm{r1}}, \dots, \sigma_{\mathrm{rM}}\} = \cup_{\mathrm{j}=1}^{\mathrm{M}} \sigma_{\mathrm{rj}}\text{---------------- (19)}$$



```
Select the Secondary path
Transfering the Data towards the Destination through the Path
Transfering the Data towards the Destination through the Path
```

Figure 3: Time Synchronization

3.4. Error Compensation and Fine-tuning

The simple concept is that node A should communicate with node B by sending a message including its time stamp in order to bring their clocks into sync. Upon receiving this communication from node A, node B promptly changes its own clock to reflect the time stamp included in the message. One of the main causes of synchronization errors, however, is the propagation delay. Since information molecules travel at a somewhat slower rate than electromagnetic waves, this becomes a far more significant issue in Main cross-Connection MC compared to wireless transmission.

$$T_{update} = T_A + T_{del} \quad \text{-------------------- (20)}$$

At that point, nodes A and B will be in sync with one another.

So, let's pretend that we're just concerned with one event within a system that has a stable limit cycle solution. As an example, it's possible that we're just concerned with the intervals between certain neuronal action potential firings. The PRC, which is dependent on both the size and the timing of the disturbance, is the time-dependent variation. For this particular oscillation, let's call the natural period T and the timings of its subsequent occurrences $t = 0, 2T, \ldots$ Assume that we interrupt the trajectory at $t = t_s \in [0, T]$.

3.5. Evaluation and Validation of PCO-RBSync

The PCORBSync algorithm is defined to address the limitations of existing time synchronization methods in WSNs by combining the benefits of both PCO and RBSync algorithms. By utilizing the gradual alignment of internal oscillations facilitated by PCO and the periodic broadcast synchronization mechanism of RBSync, PCORBSync offers improved accuracy, stability, and scalability in time synchronization for WSNs. Additionally, PCORBSync incorporates error monitoring, compensation techniques, and algorithm fine-tuning to mitigate synchronization discrepancies and optimize performance. The PCO-RBSync algorithm has been represented at algorithm 1.

The subsequent firing time is denoted as$\hat{T}\,(t_s)$. The events that take place after $t = \hat{T}\,(t_s)$ are at $t = \hat{T}\,(t_s) + T, \hat{T}\,(t_s) + 2T, \ldots$, according to a key postulate of PRC theory. What this means is that the disturbance only has an impact for a single cycle and is then forgotten. Although this is a stringent limitation, it is usually valid in reality. As a result of a sluggish potassium current, however, the following spike timing is somewhat different in. This means that the second spike after the stimulus occurs at $< \hat{T}\,(t_s) + T$, but the d difference between this and the initial phase-shift caused by the perturbation is negligible. Here, the PRC is defined as

$$\Delta(\emptyset) = 1 - \frac{\hat{T}(T\emptyset)}{T} \text{--------------- (21)}$$

The phase, denoted as $\emptyset$, is defined as the integral of t_s divided by T, where $\emptyset$ is an integer between 0 and 1. So, $\Delta(\emptyset)$ is positive (negative) if the stimulus causes the following event to occur sooner (later). On the whole, neurons include PRCs that fulfill.

$$\Delta(0) = \lim_{\emptyset \to 1^-} \Delta(\emptyset) = 0 \quad \text{---------------- (22)}$$

If the disturbance comes at the beginning of the spike, the spike time remains the same. The PRCs calculated for inhibitory cortical neurons and excitatory pyramidal neurons both exhibit this quality.

Algorithm 1: PCO-RBSync

```
Input:
1.    Network topology and node configurations.
2.    Initial clock synchronization parameters.
Steps:
Initialization:
  Configure sensor nodes with PCORBSync algorithm parameters.
$\theta_i(t+1) = \theta_i(t) + w_i + \sum_{j=1}^{N} A_{ij} . \sin(\theta_j(t) - \theta_i(t) + \emptyset_{ij})$
```

Establish reference nodes for broadcast synchronization.

Initialize clock values for each node.

$$L_i(t) = a_i t_i(t) + b_i = a_i . a_i t + a_i b_i + b_i$$

Pulse-Coupled Oscillator (PCO) Phase Alignment:

Nodes exchange pulses with neighbors to gradually align internal oscillations.

$$\hat{T}(t_s) = t_s + \frac{1}{\sqrt{I}}\left(\frac{\pi}{2} - \arctan\left[\frac{a}{\sqrt{I}} - \cot\left(\sqrt{I}(t_s)\right)\right]\right)$$

Adjust internal clocks based on received pulse timings.

Reference Broadcast Synchronization (RBSync):

Designated reference nodes periodically broadcast local timestamps.

$$d_{cj} = \{d_{cj} | d_{cj} = t - t_{sj}, \forall t_{cj}\}$$

Nodes receive broadcasted timestamps and adjust their clocks accordingly.

$$\sigma_{cj} = \sqrt{E\left(d_{cj} - E(d_{cj})\right)^2}$$

Error Monitoring and Compensation:

Continuously monitor synchronization errors and discrepancies.

$$T_{update} = T_A + T_{del}$$

Apply compensation techniques to mitigate errors and optimize synchronization performance.

Output:

1. Synchronized clock values for each sensor node.
2. Performance metrics such as accuracy, stability, and scalability.

4. Results and Discussion

Here, we offer the results of our research on the Pulse-Coupled Oscillator with Reference Broadcast Synchronization algorithm (PCO-RBSync) for WSN time synchronization. The outcomes demonstrate that the suggested method is capable of attaining exact synchronization, as confirmed by thorough testing and assessment.

Table 1: Simulation Parameters

Parameters	**Value**
SimulationTime	500(s)
NumberofNodes	0to127
Mobility	10-50m/s
RoutingProtocol	DSDV
Channel Type	flatgrid
SimulationArea	550 x480 m
TransmissionRange	250m
MAC type	Mac/802_11

4.1. Communication overhead

The communication overhead formula typically involves calculating the additional data or processing required for communication beyond the actual payload being transmitted.

$\text{Communication Overhead} = \text{Total Communication Data} - \text{Payload}$ ---------- (23)

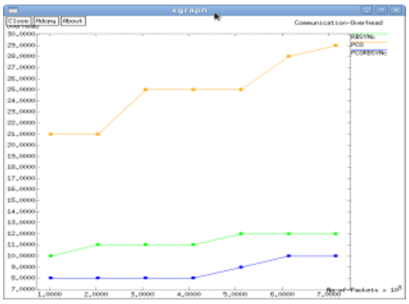


Figure 4: Visual representation of communication overhead costs

A comparison chart of transmission overhead is shown in figure 4. On one side, we can see the total packet count, and on the other, we can see the value of the communication overhead.

4.2. Broadcasts

The formula for calculating the number of broadcasts required in a network depends on several factors such as the network topology, the broadcast method used, and the number of nodes in the network.

$\text{Broadcasts} = \text{N} - 1$ --------------- (24)

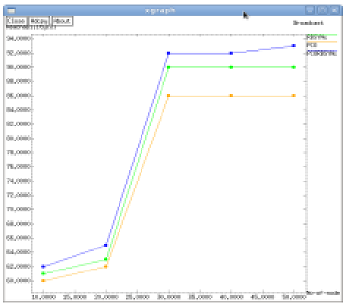


Figure 5: Visual comparison of broadcasts

Figure 5 displays a chart comparing broadcasts. Values for broadcasts are shown on the y-axis, while the number of nodes is shown on the x-axis.

C. Throughput

$\text{Throughput}= \frac{\text{Number of Packet Size}}{\text{Time duration} * \text{Successful average Packet size}}$ ----------- (25)

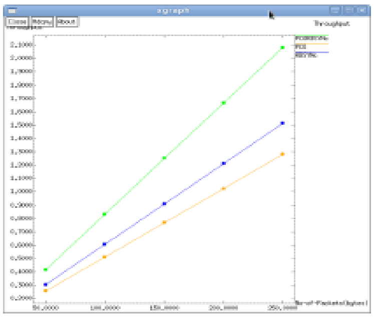


Figure 6: A graph comparing throughput

The throughput comparison chart is shown in figure 6. Number of packets is shown on the x-axis while throughput figures are shown on the y-axis.

4.3 Packet Delivery ratio

$$\text{PDR}=\frac{\text{Number of Packets Receive}}{\text{Total Packets}} * 100 \quad \text{-----------} \ (26)$$

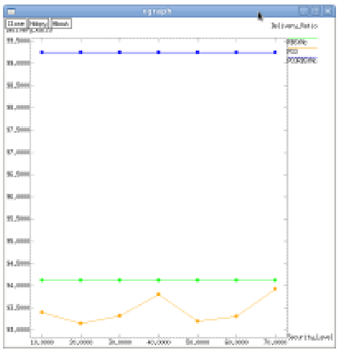


Figure 7: Table comparing delivery rations

A comparison table showing delivery rations is shown in figure 7. On one side, we have the security level, and on the other, we have the delivery ratio.

4.4 Throughput zones

The concept of "throughput zones" is not a standardized term in networking or communication theory. However, if we are to interpret it as different ranges of throughput within a network, the formula for determining throughput zones would depend on the specific criteria or thresholds set to define each zone.

Zone_1: Low Throughput ---------- (27)

Zone_2: Medium Throughput --------- (28)

Zone_3: High Throughput ------------- (29)

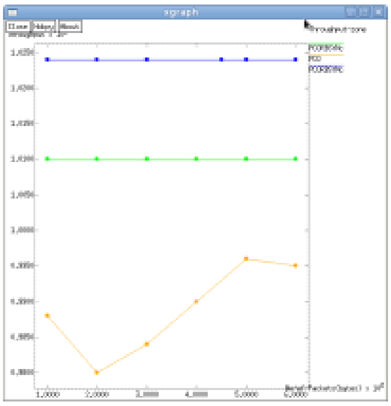


Figure 8: Graph comparing throughput zones

The throughput-zone comparison chart is shown in figure 8. The number of packets is shown on the x-axis, while the throughput-zone values are displayed on the y-axis.

4.5. Time Delay

$$\text{Time Delay}=\frac{\text{Number of Sensor nodes}}{\text{energy consumption for sending packets at a times x forwarding time in ms}} \quad \text{-----------} \ (30)$$

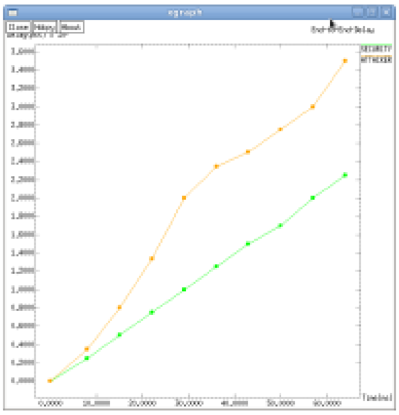


Figure 9: Time compared from beginning to end graphic

The end-to-end delay comparison chart is shown in figure 9. Both the time (in milliseconds) and the values of delays (on the y-axis) are plotted.

4.6 Density

The formula for calculating the density of a network depends on the definition of density being used. In the context of network communication, density typically refers to the number of communication links present within the network. Here's a simple formula for calculating network density:

$$\text{Density} = \frac{2\times \text{E}}{\text{N}\times(\text{N}-1)} \text{------------ (31)}$$

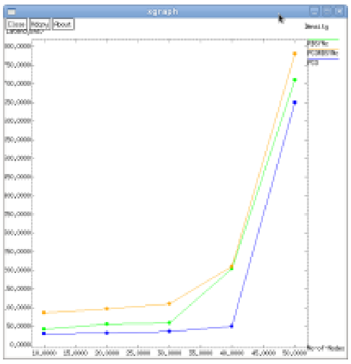


Figure 10: Visual representation of density

A density comparison table is shown in figure 10. Number of nodes is shown on the x-axis, while latency in milliseconds is shown on the y-axis.

5. Conclusion

The Pulse-Coupled Oscillator with Reference Broadcast Synchronization algorithm (PCO-RBSync) is a game-changing time-based synchronization method developed specifically for the complex requirements of WSNs. This area of study is crucial because of the need of time synchronization in many different applications, such as distributed databases, real-time communication, and distributed data gathering. The suggested synchronization architecture develops methodically, starting with the critical steps of network startup. Setting up a reference source, deploying sensor nodes, and initializing clocks are all part of this process. A dynamic mechanism for the gradual alignment of internal oscillations is introduced by the Pulse-Coupled Oscillator model, which, when integrated, fosters a cohesive network environment through pulse exchanges among surrounding nodes. The algorithm's accuracy is enhanced by its compatibility with Reference Broadcast Synchronization (RBSync). In order to maintain a synchronized temporal framework throughout the whole WSN, specific nodes periodically broadcast their local timestamps. This prompts the rest of the network to work together to determine time disparities and change their clocks. The PCO-RBSync method is made more resilient by strategies including continuous error monitoring, compensation, and algorithm fine-tuning. By using these steps, inconsistencies are reduced, and the synchronization process is optimized for increased precision and consistency. The method is very suitable for resource-constrained and dynamic wireless sensor networks because to its adaptability and robustness.

6. Reference

[1] Barolli, L., Hellinckx, P., &Natwichai, J. (Eds.). (2020). Advances on P2P, Parallel, Grid, Cloud and Internet Computing. Lecture Notes in Networks and Systems. doi:10.1007/978-3-030-33509-0

[2] Chalapathi, G. S. S., Chamola, V., Gurunarayanan, S., &Sikdar, B. (2019). E-SATS: An Efficient and Simple Time Synchronization Protocol for Cluster-based Wireless Sensor Networks. IEEE Sensors Journal, 1–1. doi:10.1109/jsen.2019.2922366

[3] Fernandez-Madrigal, J.-A., Navarro, A., Asenjo, R., & Cruz-Martin, A. (2020). Efficient Geometrical Clock Synchronization for Pairwise Sensor Systems. IEEE Sensors Journal, 1–1. doi:10.1109/jsen.2020.3014525

[4] Godor, I., Luvisotto, M., Ruffini, S., Wang, K., Patel, D., Sachs, J., … Farkas, J. (2020). A Look Inside 5G Standards to Support Time Synchronization for Smart Manufacturing. IEEE Communications Standards Magazine, 4(3), 14–21. doi:10.1109/mcomstd.001.2000010

[5] Gu, X., Zhou, G., Li, J., &Xie, S. (2020). Joint Time Synchronization and Ranging for a Mobile Wireless Network. IEEE Communications Letters, 1–1. doi:10.1109/lcomm.2020.3001138

[6] Liu, G., Yan, S., Mao, L., & Sui, Z. (2020). Time Synchronization and Clock Parameter Estimation for Wireless Sensor Networks with Unequal Propagation Delays. 2020 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC). doi:10.1109/icspcc50002.2020.9259527 Measurement, 183, 109760. doi:10.1016/j.measurement.2021.109760

[7] Y. Chen, H. Dai and B. Wang, "Long-Haul Fiber-Optic Time and Frequency Synchronization," 2023 Joint Conference of the European Frequency and Time Forum and IEEE International Frequency Control Symposium (EFTF/IFCS), Toyama, Japan, 2023, pp. 1-4, doi: 10.1109/EFTF/IFCS57587.2023.10272096.

[8] Pan, X., Shen, Y., & Zhang, J. (2021). IoUT Based Underwater Target Localization in the Presence of Time Synchronization Attacks. IEEE Transactions on Wireless Communications, 20(6), 3958–3973. doi:10.1109/twc.2021.3054745

[9] Prager, S., Haynes, M. S., &Moghaddam, M. (2020). Wireless Subnanosecond RF Synchronization for Distributed Ultrawideband Software-Defined Radar Networks. IEEE Transactions on Microwave Theory and Techniques, 68(11), 4787–4804. doi:10.1109/tmtt.2020.3014876

[10] Shams, R., Khan, F. H., Amir, M., Otero, P., &Poncela, J. (2020). Critical Analysis of Localization and Time Synchronization Algorithms in Underwater Wireless Sensor Networks: Issues and Challenges. Wireless Personal Communications. doi:10.1007/s11277-020-07233-1

[11] Shams, R., Otero, P., Aamir, M., & Khan, F. H. (2021). Joint Algorithm for Multi-Hop Localization and Time Synchronization in Underwater Sensors Networks Using Single Anchor. IEEE Access, 9, 27945–27958. doi:10.1109/access.2021.3058160

[12] Song, H., & Jiang, D. (Eds.). (2021). Simulation Tools and Techniques. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. doi:10.1007/978-3-030-72792-5

[13] Sun, X., Su, Y., Huang, Y., Tan, J., Yi, J., Hu, T., & Zhu, L. (2020). Design and development of a wireless sensor network time synchronization system for photovoltaic module monitoring. International Journal of Distributed Sensor Networks, 16(5), 155014772092734. doi:10.1177/1550147720927341

[14] Tang, H., Wang, J., Tang, Z., & Song, J. (2020). Scheduling to Minimize Age of Synchronization in Wireless Broadcast Networks with Random Updates. IEEE Transactions on Wireless Communications, 1–1. doi:10.1109/twc.2020.2979436

[15] Wang, H., Gong, P., Yu, F., & Li, M. (2020). Clock Offset and Skew Estimation Using Hybrid One-Way Message Dissemination and Two-Way Timestamp Free Synchronization in Wireless Sensor Networks. IEEE Communications Letters, 1–1. doi:10.1109/lcomm.2020.3019521

[16] Wang, Z., & Wang, Y. (2020). Global Synchronization of Pulse-Coupled Oscillator Networks Under Byzantine Attacks. IEEE Transactions on Signal Processing, 1–1. doi:10.1109/tsp.2020.2993643

[17] Wu, T.-Y., & Wang, W. (2019). An improvement mechanism for heartbeat-driven MAC synchronization in wireless body sensor network. International Journal of Communication Systems, e4224. doi:10.1002/dac.4224

[18] Wu, Y., & He, X. (2020). Finite-Time Consensus-Based Clock Synchronization Under Deception Attacks. IEEE Access, 8, 110748–110758. doi:10.1109/access.2020.3002577

[19] Xue, B., Li, Z., Lei, P., Wang, Y., &Zou, X. (2021). Wicsync: A wireless multi-node clock synchronization solution based on optimized UWB two-way clock synchronization protocol.

[20] Yang, Y., Yang, Y., Hu, X., Tang, C., Guo, R., Zhou, Z., … Su, M. (2021). BeiDou-3 broadcast clock estimation by integration of observations of regional tracking stations and inter-satellite links. GPS Solutions, 25(2). doi:10.1007/s10291-020-01067-x

[21] Chen Z, Anglea T, Zhang Y, Wang Y. Optimal synchronization in pulse-coupled oscillator networks using reinforcement learning. PNAS nexus. 2023 Apr 1;2(4):pgad102.

[22] H. Kim, M. Ishikawa and Y. Yamakawa, "Reference broadcast frame synchronization for distributed high-speed camera network," 2018 IEEE Sensors Applications Symposium (SAS), Seoul, Korea (South), 2018, pp. 1-5, doi: 10.1109/SAS.2018.8336781.