SOUTHEAST ASIAN JOURNAL OF SCIENCE AND TECHNOLOGY

Volume 6, Issue 2, (Special Issues) 2021
P-ISSN: 2672-2984
E-ISSN: 2672-2992
www.sajst.org

A Comparative Study of Distributed Clock Synchronization Algorithms

Richard D. Hortizuela
Ilocos Sur Polytechnic State College

Abstract – *This paper explores the concept of distributed clock synchronization and its use cases. It examines the algorithms that were implemented in various researches over the years. This paper highlights the different factors which affect the performance of clock synchronization in a distributed system and compares the performance of existing synchronization methods to analyze their impairments and shortcomings and tries to present possible solutions to address the issues currently faced in distributed clock synchronization. In order to gather literature, Manu Bhatia's method in qualitative research was implemented. For the comparison of the algorithms, specific parameters of comparison have been defined such as fault tolerance, scalability, energy efficiency, approach, accuracy and purpose of implementation based on the standards set by the ISO/IEC 27002:2013. Several factors that affect the performance of distributed clock synchronization have been uncovered such as failures and errors, scalability, sleeping schedule and finally, security attacks. Seven algorithms were examined and compared which are implemented from the year 2015 to 2019. The results of the comparative study show that among the seven recently implemented clock synchronization algorithms including the most commons protocols, the Network Time Protocol (NTP) and Precision Time Protocol (PTP), the Xie's Algorithm which uses a fast algorithm for consensus and allow solutions for truly scalable networks exemplary met the standards and stands out having the highest level of accuracy.*

Keywords – *Distributed clock synchronization, synchronization algorithms, distributed system*

Introduction

Imagine asking a group of audience what time is it, and you will always get different answers. Similar to this scenario, in centralized systems, time is unambiguous as each process can make a system call to get the current time. However, processes may get or produce different timestamps. [1] On the other hand, distributed systems (DS), there exist difficulties in agreement on time. In normal operations, it doesn't affect much. However, there exist systems and programs that require time to be precise.

In a distributed system, computers are connected via a high-speed communication network. These computers needed to work and operate in unison through coordination of events. This process is called clock synchronization [2]. The requirement in precision in clock synchronization (CS) or time synchronization (TS) has become more and more pertinent as communication technologies have moved from inherently synchronous networks. Eventually, this has led to the use of Ethernet as the basis for communication between distributed devices in automation networks. However, ethernet has its drawback specifically the inability to support real-time communication [3]. Since there are usually errors in initial setting of the clock, and the clock will drift itself, clocks in distributed systems must be corrected periodically to keep consistent with the standard clock.

As part of the advancement in this venture, protocols have been implemented. Generally, there are two main clock synchronization protocols for distributed networks, which are the client-server-based Network Time Protocol (NTP) and master-slave-based Precision Time Protocol (PTP). [4] These protocols have been used and implemented throughout the years, even now. However, accuracy and reliability remain an

SOUTHEAST ASIAN JOURNAL OF SCIENCE AND TECHNOLOGY

Volume 6, Issue 2, (Special Issues) 2021
P-ISSN: 2672-2984
E-ISSN: 2672-2992
www.sajst.org

issue and may cause problems in various DS applications like the following sample scenarios, as mentioned by Dalwadi [5]:

1. In a distributed banking system, if the timing and ordering of financial transactions are not tracked, it may raise inconsistent state in the system
2. A distributed online reservation system in which the last available seat may get booked from multiple nodes if their local clocks are not synchronized

These are just pieces of the hundreds of problems arising because of unsynchronized clocks in various systems. Different researchers and organizations tried to bridge the gap in precise clock synchronization.

In 2010, Chen et al. [6] used a feedback-based clock synchronization, a lightweight synchronization protocol to solve the problems associated with clock model perturbation and external disturbances, which widely exist in most low-frequency sleep clocks in wireless sensor networks. However, it was not made for relatively large networks, making it very difficult to use it in resource-constrained networks.

Further researches have been conducted to improve existing protocols and develop new algorithms to achieve the common goal of a synchronized clock. In 2012, Hao and his team [7] developed *WizSync*, a novel time synchronization approach by exploiting the existing Wi-Fi infrastructure at that time. It employs advanced signal processing techniques to detect periodic Wi-Fi beacons and use them to calibrate the frequency of native clocks. The following year, Lenzen et al [8] proposed PulseSync, an efficient and scalable clock synchronization protocol. It offers asymptotically optimal trade-offs between various design goals. The study has shown positive results. However, it does not support a wider range of topologies that distributed systems usually have, and especially in conjunction with dynamics and do not offer robustness.

The venture to a precise and synchronized clock had developed over the years, even up to this day. One of the most recent studies on different algorithms was conducted by Dalwadi [5] but it covers old algorithms which were implemented from 1984 to 2010. This paper attempts to continue to examine the recent algorithms in distributed clock synchronization which were implemented in various researches from 2015 through 2019. Moreover, this paper primarily pursues to answer the following questions:

1. What are the different use cases of cock synchronization and the factors that affect its performance in a distributed system?
2. In different distributed systems, what are the existing problems and issues organizations and researchers face in clock synchronization?
3. What clock synchronization algorithms were implemented in various researches in recent years, the purpose of its implementation, and the drawbacks or limitations?
4. What are the parameters for comparison of the different algorithms?

Through analysis and interpretation of the results of the comparative study, the best of which is recommended for use based on the set parameters for comparison. Furthermore, this paper provides a brief overview of the different emerging for clock synchronization and their features that could be required by applications. The different processes and procedures in conducting the study have been presented in Section 2. The findings were presented in Section 3 respectively.

MATERIALS AND METHODS

2.1 Browsing research databases

In order to achieve the aforementioned research goals, in this section, the methodology and procedures applied were discussed.

Online research databases such as Google Scholar, ResearchGate, IEEE Xplore Digital Library, ScienceDirect, and Springers were utilized to discover and access journal articles, conference proceedings, technical standards, and related materials using the

SOUTHEAST ASIAN JOURNAL OF SCIENCE AND TECHNOLOGY

Volume 6, Issue 2, (Special Issues) 2021
P-ISSN: 2672-2984
E-ISSN: 2672-2992
www.sajst.org

keywords *clock synchronization, time synchronization, distributed clock synchronization, clock synchronization algorithms, and synchronization algorithms.* The search terms were used for all fields including title, abstract, keywords, and full text.

In browsing for research articles online, the following procedure, as crafted by Manu Bhatia [9] and stated in the Humans of Data website was applied:

2.1.1 ***Getting familiar with the data****:* Since most qualitative data are just words, the researcher started by reading the data several times to get familiar with it and start looking for basic observations or patterns. The choosing of articles was based through the following criteria:

1. published from January 2015 to March 2020
2. relevant or has the same line of interest as of this review
3. peer-reviewed or a proceeding of a research symposium
4. availability of the full-text copy

However, some articles published earlier than 2015 were utilized in order to establish the development of research throughout the years and to establish a need for further research.

2.1.2 ***Revisiting research objectives***: Here, the researcher revisits the aforementioned research objectives and identifies the questions that can be answered through the collected data

2.1.3 ***Identifying patterns and connections****:* In this process, the research looked for the most common responses to questions, identifying data or patterns that can answer research questions, and finding areas that can be explored further. Furthermore, the references section of each research article has been looked upon for any relevant researches that could help in a more comprehensive review.

2.2 Identifying clock synchronization use cases and performance factors

In identifying the various use cases of clock synchronization in a distributed system, several papers were examined with the following criteria:

1. paper identifies clearly the use case
2. includes a comprehensive description of the features, requirements, and the motivation of implementation each use case
3. the use case has been mentioned and/or implemented in a research or in a real-world environment

Then, in order to identify the different factors affecting clock synchronization performance, an academic online resource browsing was conducted. After familiarization, a thorough investigation of peer-reviewed and up-to-date researches was conducted with a focus on pointing out the specific points of interest or motivations in conducting the research.

2.3 Identifying issues on clock synchronization in distributed systems

In identifying issues regarding clock synchronization in distributed systems, various papers were examined with the basis of the following metrics:

1. the paper identifies a specific gap relevant to clock synchronization
2. he paper comprehensively explains the drawbacks
3. the paper includes probable or proposed solutions to address the issues
4. the paper clearly identifies the future directions of the research stating that further research is recommended or required

The different issues identified in the various papers which were examined are aggregated with the consideration of the different use cases uncovered by following the procedures on Section 2.2. The issues identified are either predominant or common in the various clock synchronization use cases.

2.4 Identifying and examining recently implemented clock synchronization algorithms

To be able to identify the clock synchronization algorithms that are implemented recently, a focused search on online repositories was conducted by the means of using the keywords *clock synchronization algorithms* and *time synchronization algorithms*.

SOUTHEAST ASIAN JOURNAL OF SCIENCE AND TECHNOLOGY

Volume 6, Issue 2, (Special Issues) 2021
P-ISSN: 2672-2984
E-ISSN: 2672-2992
www.sajst.org

Moreover, in the focused search, the papers to be examined should include the following:

1. the algorithm was implemented in a research or by an organization from the year 2015 to 2020.
2. clear identification and description of the algorithm, its motivation/s of creation and implementation
3. proof of concept, computational methodology of the algorithm
4. the algorithm has been tested, simulated or implemented in research laboratory or a real-world environment

In order to give fairness, the algorithms selected were implemented for varying purposes (i.e telecommunication, finance, estimation, improvement of current research) and by different researches.

2.5 Setting and defining the parameters of comparison

Part of the comparative study is to set the parameters of comparison, thus, in order to attain this, similar studies (i.e comparative analysis, performance analysis, survey of methodologies) were examined and the metrics and parameters used in those peer-reviewed researches were adapted in this study. The metrics of choosing the articles as basis for adapting the parameters are the following:

1. the paper clearly stated, defined, and justified the parameter/s
2. the paper specifies the motivation of using such parameter/s
3. the parameters used in the paper should be in compliance with the standards set by ISO on operations as stated in ISO/IEC 27002:2013.

RESULTS

3.1 Filtered researches for review

In the initial process of searching for articles in the different research databases, presented in Table 1. is the average number of results appeared in the different online repositories of researches exclusive of the article citations.

Table 1. Results from searched online databases

Research Database	Total Number of Results	Peer-Reviewed
Google Scholar	22,400	N/A
ScienceDirect	389	299
IEEE	2,256	2,173
ResearchGate	128	38
Springer	1865	5

The total number of results (Column 2) signifies the popularity of the subject of the research throughout the years 2015 up to date which contains all results, including non-scientific writings such as magazine articles, trade publications etc.

During the process of browsing for articles, a total of 63 research articles and journals were aggregated as prospects for utilization. These articles were filtered according to their publishing date in which articles published from the year 2015 to 2020 were the only ones considered to be utilized in this review. The research papers were also scanned by the researcher through the aforementioned methods. Articles gathered came from different authors, institutions, organizations, and countries. As a result of filtering, out of the 63 gathered researches, only 23 of these were used as references for this paper as shown in Table 2.

Table 2**.** Tally of Utilized Researches

Research Database	Articles
Google Scholar	[2] [5] [9] [14] [16] [24]
ScienceDirect	[15] [21] [23]
IEEE	[3] [6] [7] [8] [11] [12] [17] [18] [19] [20]
ResearchGate	[1] [22]
Springer	[4] [10] [13]

SOUTHEAST ASIAN JOURNAL OF SCIENCE AND TECHNOLOGY

Volume 6, Issue 2, (Special Issues) 2021
P-ISSN: 2672-2984
E-ISSN: 2672-2992
www.sajst.org

3.2 Clock synchronization use cases and performance factors

Clock synchronization plays an important role to different applications and systems like in the Internet of Things (IoT), Wireless Sensor Network (WSN), and in a system of distributed components or objects and some of them are as follows:

1. In smart grid, different parts of electrical grid to connect and disconnect without disrupting customers and it allows networks to impose order on streams of data from scattered sensors.
2. In mobile communications, the sequencing of calls is important when two or more callers trying to communicate with the same receiver [10].
3. In industrial automation and data acquisition applications, synchronization is also critical to improve precision, productivity, and quality.
4. In law enforcement, military and crisis applications, accurate time is a key element as a variety of audio, video, and measurement data are collected in challenging environments, the processing of this information is significantly affected if timestamps are not accurate [11].
5. In the financial industry, clock synchronization is also critical where Precision Time Protocol (PTP) is currently the only solution meeting the regulatory requirements. For example, NYSE Euronext, one of the world's leading exchange operators and market access providers, requires sub-100 microsecond to 10 microsecond synchronization range. [12]

With clock synchronization, these applications work more effectively. However, there are factors that affect the performance of a clock. These factors are discussed below the presented Table 3.

Table 3. Factors Affecting the Clock Synchronization Algorithm Performance

Factors	**Related Works**
Failures and Errors	[1] [3] [5] [11] [12] [13] [15]
Scalability	[3] [5] [8] [11] [14] [15]
Sleep Schedule	[3] [11] [15] [22]
Attacks (Security)	[1] [3] [11] [5]

3.2.1 **Failures and Errors.** In any kind of environment or system, the room for failures and errors must be small or non at all. A single error in a node will result to an inaccurate clock which will cause a disaster where machines will not operate in unison. In the study of Jie et al. [13], they proposed distributed second-order consensus time synchronization algorithm in which results reveal that faulty nodes are the predominant issue in clock synchronization. Moreover, an algorithm should be robust or tolerant to failure from small to large-scale projects which brings up another factor, protocol's scalability.

3.2.2 **Scalability.** The scalability of a clock synchronization protocol in WSNs is a relative notion, and it depends on the objective to be achieved by the clock synchronization protocol, and the associated cost per sensor node per unit time. If this cost increases as the total number of nodes increases, then the protocol is not scalable [15]. Depending upon the objective, a protocol may or may not be considered to be scalable. Furthermore, as the project scales up, the more energy it consumes. Thus, a mechanism is needed to minimize the consumption of energy.

3.2.3 **Sleep Schedule.** The most important technique to reduce energy consumption of sensor nodes is to put at least the transceiver to sleep mode whenever a sensor node does not need to communicate with others. When it is considered that a sensor node is put to sleep mode, it means that at least the transceiver is put off [15]. For instance, the mesh network IEEE 802.15.4e, it targets industrial applications and its main feature is the time synchronized channel hopping to increase the reliability of transmissions and reduce the energy consumption of nodes [3]. However, reducing the energy consumption of nodes is as important as making them prone to attacks.

3.2.4 **Attacks.** Security is of increasing importance for systems where time is critical, since malicious attacks to the clock synchronization service could cause damage and issues to the equipment and service reliability. In [11], results show that threats such as manipulation, spoofing and Denial of Service (DoS) attacks are the causes of false time and accuracy degradation. Thus, synchronization services should use few computing resources and provide high accuracy in a secure manner.

3.3 Distributed clock synchronization issues

In distributed systems, there are many reasons for a communication delay which needs to be addressed to get a nearby accurate time to achieve synchronization and some of which are the following:

1. **Faulty Nodes.** In state of the art clock synchronization, faulty agents or nodes with non-cooperative behaviors are inevitable to interfere system coordination, such as Internet ghost- writers to influence public opinions, enemy aircrafts to disturb unmanned aerial vehicle formation control, malfunctioning clocks to break clock synchronization, etc. [16] This issue is predominant in all of the aforementioned use cases.
2. **Clock Drift.** In a distributed system, each computer consists of quartz crystal that oscillates at a predictable frequency when squeezed. In practice, a register in a clock I is chosen to be 60 tick per second. There might be differences in crystal oscillation which leads to the clock running at different rates. Hence, computer clock drifts from the real-time clock. [5] The occurrence of clock drift issue highly impacts all the use cases mentioned above but is more serious in IoT applications.
3. **Clock Skew**. It is a phenomenon, in which a clock does not run at the exactly same speed as the reference clock. It is caused by the clock accuracy as well as ambient effects such as temperature, pressure and aqueous medium which may consume more energy [17].

3.4 Recently implemented clock synchronization algorithms

Through the development of the research on the improvement of clock synchronization techniques to achieve near-perfect sync, algorithms have been crafted, tested and implemented by various tech giants, organizations and researchers in the academe. In this section, algorithms selected were implemented for a specific purpose and published consecutively from 2015 to present to establish a perspective focusing on the development of research. However, common algorithms implemented by majority of organizations were included to expand the comparison. In Section 3.4 and 3.5 algorithms presented with no specified name are named after the main author.

At first, existing protocol like the Network Time Protocol (NTP), invented on 1981, is designed to synchronize the clocks of computers over a network. This is a fault-tolerant protocol that will automatically select the best of several available time sources to synchronize to. Multiple candidates can be combined to minimize the accumulated error. Temporarily or permanently insane time sources will be detected and avoided [18]. However, it has drawbacks in terms of performance and accuracy. With its latest version, NTP 4, released on March 2019, it has a redesigned clock discipline algorithm that improves accuracy, handling of network jitter, and polling intervals. This now has a support for the nanokernel kernel implementation that provides nanosecond precision as well as improved algorithms. Furthermore, up to this day, similar to NTP, the IEEE 1588 Precision Time Protocol (PTP), has become a de facto standard for CS in various applications. [19]

In 2015, Lin et.al [17] proposes an algorithm on a two-way message exchange mechanism with the molecular propagation delay based on the inverse

SOUTHEAST ASIAN JOURNAL OF SCIENCE AND TECHNOLOGY

Volume 6, Issue 2, (Special Issues) 2021
P-ISSN: 2672-2984
E-ISSN: 2672-2992
www.sajst.org

Gaussian distribution. The main purpose of their work was to estimate the clock offset and the clock skew for the clock synchronization, therefore enabling the nanomachines to work in an identical timeframe using diffusion. However, this model operates in nanoscale, thus, it may not be considered to be scalable. In relevance, as the network size increases, the accuracy becomes worse. On the brighter side, with its specified purpose on nanomachines, in comparison of the proposed scheme with a clock synchronization method based on symmetrical propagation delay, it demonstrates that Lin's Algorithm proposed scheme can achieve a better performance in terms of accuracy. Moreover, this method is energy efficient and has low heat dispersion.

Within the same timeline, a group of Turkish computer engineers presented a novel distributed synchronization algorithm based upon a Proportional-Integral (PI) controller for Wireless Sensor Networks (WSNs), PISync as they call it [20]. The algorithm was inspired by an existing optimal clock synchronization algorithm. It compensates the clock offsets and the differences in clock speeds based on a Proportional-Integral (PI) controller. The paper provides a theoretical analysis to highlight the benefits of the algorithm in terms of improved steady state error and scalability as compared to least-squares based time synchronization. Since *PISync* is linear, simple and easy to implement, it is perfectly suitable for WSNs. This approach, referred as flooding-based time synchronization, allows each receiver node to calculate its relative offset and frequency differences with respect to the received time information, and to estimate the reference time. In the paper, highlighted are its benefits: a) improved steady state error, b) scalability, c) energy efficiency, and d) minimum requirement.

The following year, Luo et.al [21] studies the global clock synchronization problems for wireless sensor networks under unknown exponential delays in the two-way message exchange mechanism. They proposed a fully-distributed synchronization algorithm which has low communication overhead and computation cost. Simulation results show that the proposed algorithm achieves better accuracy than consensus algorithm and the distributed least squares algorithm, and can always converge to the centralized optimal solution.

An improved algorithm was proposed by Sun et al. [22] in the succeeding year. In their study, they summarized the synchronization error and energy consumption factors, and designed an improved time synchronization algorithm, that is, clustering-based low energy consumption time synchronization (CLECTS). The algorithm uses the LEACH protocol divides the whole wireless sensor network into several clusters, and the cluster head uses cycle random mode which enable low energy consumption. As compared to classical algorithms, CLECTS offer better performance, higher accuracy and lower network cost. With same object as the CLECTS plus the added solution for scalability, Xie et al [23] proposed new fully distributed clock synchronization algorithm. The algorithm is developed based on a finite average consensus concept. Many nice properties of the algorithm have been studied, including low complexities, robustness against transmission adversaries and asynchronous implement ability.

With these existing mechanisms, it is indeed a developing research aiming to provide a better, or not best algorithm for clock synchronization in distributed systems. These algorithms differ in specified purpose. However, they all aim at the same mark. The summary of the comparison is presented in Section 4. The succeeding section discusses the parameters used for comparing the algorithms.

3.5 Parameters of comparison

Through examining similar studies (i.e comparative analysis, performance analysis, survey of methodologies) in clock synchronization algorithms, Table 4. presents the parameters or metrics be used in comparing the different algorithms included are the parameters based on the different factors and issues presented in Sections 3.2 and 3.3, respectively. The

SOUTHEAST ASIAN JOURNAL OF SCIENCE AND TECHNOLOGY

Volume 6, Issue 2, (Special Issues) 2021
P-ISSN: 2672-2984
E-ISSN: 2672-2992
www.sajst.org

parameters used are in compliance with the standards set by ISO on operations as stated in ISO/IEC 27002:2013 [24], the code of practice for information security controls in specification on logging and monitoring services.

Table 4. Summary of Parameters of Comparison

Parameter	**Reference Paper**
Fault Tolerance	[1] [5] [11] [12] [15]
Scalability	[5] [8] [11] [14] [15]
Energy Efficiency	[3] [11] [15] [22]
Accuracy	[1] [5] [11] [15]
Approach	[1] [15]
Purpose of Implementation	N/A

In reference to Section 3.2.1, fault tolerance is used a parameter of comparison. It indicates if the algorithm is tolerant to errors and failure of nodes. Secondly, scalability pertains to property of the algorithm to adapt from small to large scale networks (see Section 3.2.2). Third, energy efficiency pertains to the capability of using less network cost and energy in general. In addition to these parameters is the accuracy which denotes the level of capability of the algorithm. The accuracy is evaluated as Low (nanoseconds), Average (microseconds), and High (milliseconds). Moreover, since algorithms were designed for different purpose, this paper made use of the approach as a parameter of comparison.

With the recently implemented clock synchronization algorithms, seven of which are discussed in Section 3.4. In this section, Table 5 presents the summary of the comparative study of the algorithms.

Algorithm	**Year**	**Approach**	**Purpose of Implementation**	**Fault Tolerance**	**Scalability**	**Energy Efficiency**	**Accuracy**	**Related Work**
Lin's Algorithm	2015	Diffusion	Clock offset and Skew Estimation	Yes	No	High *(low heat dispersion)*	High *(nanosecond)*	[17]
PISync	2015	Flooding and Fully-Distributed	Compensate both clock offset and frequency differences	Yes	Yes	High *(lightweight CPU usage)*	Average *(140 micro-seconds)*	[20]
Luo's Algorithm	2016	Fully-distributed	Global clock synchronization for wireless sensor networks	Yes	Yes	High *(low computational cost)*	Low *(stated as the future work)*	[21]
CLECTS Algorithm	2017	Clustering	Improve accuracy and lower network cost compared classical algorithms	Yes	No	High *(LEACH Protocol)*	Average *(50-100 microseconds)*	[22]
Xie's Algorithm	2018	Fully-distributed, finite average consensus	Fast algorithm for consensus and allow solutions for truly scalable networks	Yes	Yes	High	High *(50 nanoseconds)*	[23]

(continued on the next page)

Table 5 (continued)

Network Time Protocol (NTP)	2019	Intersection	Coordinated Universal Time (UTC)	Yes	Yes	Average	High *(nanosecond)*	[18]
IEEE 1588 Precision Time Protocol (PTP)	2019	Ethernet / IP-based	Financial transaction, telecommunications , sub-sea acoustic arrays	Yes	Yes	High	Average to High *(micro-second to nanosecond range)*	[19]

As shown in Table 5, throughout the years, various clock synchronization algorithms are designed for a specific purpose. The most significant thing about its development is that all of the examined algorithms are fault tolerant despite its year of implementation. This addresses the aforementioned predominant issue in clock synchronization which is failure and error. Since algorithms are designed for various purposes, researchers used different approaches. 3 out of the 7 examined algorithms implemented a fully-distributed approach which provides low complexities and low communication overhead [13]. The approach that stands out is the diffusion-based concept used by the Lin's Algorithm and the intersection by the NTP which achieves a more precise level of accuracy however, it is not scalable as compared to different approaches used in the rest of the algorithm.

In terms of energy efficiency, the NTP consumes more energy and network cost as compared to the 7 other algorithms as they used different protocols to lessen energy consumption as shown in Column 7 (Energy Efficiency).

Finally, despite the tolerance to failure of the Luo's Algorithm, the algorithm does not focus on achieving a higher level of accuracy. PTP has the capability of hitting a nano-level accuracy however, experimentation results reveal that it can go down to a micro-level accuracy [18].

CONCLUSION

In this paper, the recently implemented clock synchronization (CS) have been discussed and evaluated. Moreover, the results of the study show that clock synchronization is playing a significant role to different systems and applications such as smart grid, Internet of Things (IoT), mobile communications, industrial automation and data acquisition, law enforcement, and the financial industry. In these various applications, clock synchronization performance is directly inclined to its faulty nodes, algorithm scalability, sleep schedule or idling, and security threats and attacks. Thus, a clock synchronization service should use fewer computing resources to save energy, and should provide the highest accuracy possible in a secure manner.

The study uncovered that in distributed clock synchronization, the predominant issue is the faulty and non-cooperative behavior of nodes. With the differences in crystal oscillation, clocks may run at different rates causing a phenomenon called clock drift which highly impact all the CS use cases

Throughout the years, from 2015 to 2019, researchers have been attempting to provide the best algorithm for clock synchronization. In this paper, 7 recently implement algorithms have been examined and evaluated. The most common protocols, Network Time Protocol (NTP) and the Precision Time Protocol (PTP), were included in this study. These protocols are widely used by organizations during the present time however, they have their drawbacks in accuracy and energy efficiency. The rest of the algorithms examined are the Lin's Algorithm and PISync, which were both implemented in 2015, Luo's, CLECTS, and Xie's algorithm are implemented from 2016 to 2018, respectively.

In order to compare these algorithms, parameters of comparison have been set in compliance with the ISO/IEC 27002:2013:

1. Fault-tolerance
2. Scalability
3. Energy Efficiency
4. Accuracy
5. Approach
6. Purpose of Implementation

As a result of the comparative study, it was uncovered that all of the algorithms evaluated are fault-tolerant. However, Lin's and CLECTS algorithms do not support large scale of network topologies. Moreover, all of the algorithms met the standard in energy consumption. The seven algorithms were designed for various purposes and utilized different approaches. The most common approach used is the fully-distributed based concept algorithm. In terms of accuracy, The Lin's, Xie's, NTP and PTP are capable to achieve a nano-level accuracy. Through analysis and interpretation of the comparative study, the Xie's Algorithm stands over the rest of the algorithm as it excellently met the parameters, highlighting its accuracy of 50 nanoseconds. Moreover, the Xie's algorithm supports a fast and truly-scalable network. Therefore, Xie's algorithm is recommended to use as a clock synchronization algorithm for distributed networks as it is fast, accurate, adaptable and consumes energy efficiently without sacrificing significant nodes.

REFERENCES

[1] Charles K. Gitonga, 2015. Clock Synchronization in Distributed Distant Objects. *International Journal of Advance Research in Computer Science and Management Studies* (Dec, 2015), pg. 143-146.

[2] Tirado Andrés & Araujo Am. (2019). Performance of clock sources and their influence on time synchronization in wireless sensor networks. *International Journal of Distributed Sensor Networks*. https://doi.org/10.1177/1550147719879372

[3] Aneeq Mahmood, Reinhard Exel, Henning Trsek and Thilo Sauter. (2017). Clock Synchronization Over IEEE 802.11—A Survey of Methodologies and Protocols. *IEEE Transactions on Industrial Informatics.*

[4] Hongliang Li, Xuan Feng, Song Shi, Fang Zheng, & Xianghui Xie. (2015). A High-Accuracy Clock Synchronization Method in Distributed Real-Time System.

[5] Neha N. Dalwadi1 & Mamta C. Padole. (2015). Comparative Study of Clock Synchronization Algorithms in Distributed Systems. *Advances in Computational Sciences and Technology.* ISSN 0973-6107 Volume 10, Number 6 (2017) pp. 1941-1952

[6] Jiming Chen, Qing Yu, Yan Zhang, Hsiao-Hwa Chen, & Youxian Sun (2010). Feedback-Based Clock Synchronization in Wireless Sensor Networks: A Control Theoretic Approach.

[7] Tian Hao, Ruogu Zhou., Guoliang Xing, Matt Mutka, and Jiming Chen (2013). WizSync: Exploiting Wi-Fi Infrastructure for Clock Synchronization in Wireless Sensor Networks. *IEEE Transactions on Mobile Computing*

[8] Christoph L., Philipp S. & Roger W. (2015) PulseSync: An Efficient and Scalable Clock Synchronization Protocol.

[9] Manu Bhata. (2018). Your Guide to Qualitative and Quantitative Data Analysis Methods. Retrieved from https://humansofdata.atlan.com /2018/09/qualitativequantitative-data-analysismethods/#Data_Preparation_and_Basic_Data_Analysis

[10] Neha Dalwadi & Mamta Padole. (2019) An Insight into Time Synchronization Algorithms in IoT. Data, Engineering and Applications. https://doi.org/10.1007/978-981-13-6351-1_23

[11] Martin Lévesque & David Tipper (2016). A Survey of Clock Synchronization Over Packet-Switched Networks. *IEEE Communications Surveys & Tutorials, Vol. 18, No. 4, Fourth Quarter*

[12] Pedro V. Estrela, Sebastian Neusüß, & Wojciech Owczarek (2015). Using a multi-source NTP watchdog to increase the robustness of PTPv2 in financial industry networks. *IEEE Symp. Precis. Clock Synchronization Meas. Control Commun. (ISPCS*), Austin, TX, USA, 2015, pp. 87–92.

[13] Wu Jie, Bai Yu, & Zhang Lyi. (2015). Distributed Time Synchronization in Wireless Sensor Networks via Second-Order Consensus Algorithms. *DOI 10.1007/s12209-015-2579-5*
[14] Yilong Geng, Shiyu Liu, & Zi Yin, Ashish Naik, Balaji Prabhakar, Mendel Rosenblum, & Amin Vahdat (2018). Exploiting a Natural Network Effect for Scalable, Fine-grained Clock Synchronization. *Proceedings of the 15th USENIX Symposium on Networked Systems Design and Implementation.*
[15] Amulya Ratna Swain & R.C. Hansdah (2015). A model for the classification and survey of clock synchronization protocols in WSNs. *Ad Hoc Networks* 27 (2015) 219–241
[16] Jian Hou, Zhiyong Chen, Zhiyun Lin, & Mengfan Xiang (2020). Resilient Consensus via Weight Learning and Its Application in Fault-Tolerant Clock Synchronization.
[17] Lin Lin, Chengfeng Yang, Maode Ma, & Shiwei Ma (2015). Diffusion-Based Clock Synchronization for Molecular Communication under Inverse Gaussian Distribution. *IEEE Sensors Journal* DOI 10.1109/JSEN.2015.2429673,
[18] Walt Kelly (2014). The Network Time Protocol (NTP) Distribution
[19] Irfran Allahi, Bilai Khan, Aamir Soail Nagra, Rabbia Idrees, & Shahid Masud (2018). Performance Evaluation of IEEE 1588 Protocol Using Raspberry Pi over WLAN. 2018 IEEE International Conference on Communication Systems (ICCS). doi:10.1109/iccs.2018.868922
[20] Kasım Sinan Yıldırım, Ruggero Carli, & Luca Schenato (2015). Adaptive Control-Based Clock Synchronization in Wireless Sensor Networks.
[21 Bin Luo, Lei Cheng, & Yik-Chung Wu (2016). Fully distributed clock synchronization in wireless sensor networks under exponential delays. Signal Process. 125, C (August 2016), 261–273. DOI: https://doi.org/10.1016/j.sigpro.2016. 02.007
[22] Ji-hong Sun & Huanzheng Shao (2017). An Improved Time Synchronization Algorithm for Wireless Sensor Networks.
[23] Kan Xie, Qianqian Cai, & Minyue Fu (2018). A fast clock synchronization algorithm for wireless sensor networks. Automatica 92 (2018) 133–142. iJOE – Vol. 13, No. 5, 2017. DOI: https://doi.org/10.3991/ijoe.v13i05.7049
[24] ISO/IEC 27002:2013. Information technology — Security techniques — Code of practice for information security controls.