Rover Publications
United International Journal of Engineering and Sciences (UIJES)
An International Peer-Reviewed (Refereed)Engineering and Science Journal
Impact Factor:8.075(SJIF) Vol-5, Issue- 3,(July, August & September) 2025 ISSN:2582-5887
www.uijes.com

Time Synchronization Challenges in Distributed Observability: Ensuring Accurate Event Correlation Across Hybrid Infrastructure

Sai Raghu Ram Gummadidala
Ashburn, Virginia, USA 20148
rrgummadidala@gmail.com

Abstract

Synchronization in the process of creating a distributed observability system is currently considered an extremely important aspect for the effective monitoring of modern infrastructures consisting of cloud platforms, edge computing frameworks, IoT devices, and distributed services. It is clear that accurate correlation of events becomes possible if distributed nodes are synchronized. But heterogeneous infrastructure brings serious synchronization problems such as clock drift, latency, delay propagation, inconsistencies in time-stamping, and synchronization instability that may negatively influence distributed tracing, anomaly detection, and the root cause analysis. In this study, we will present the concept of synchronization-aware distributed observability system aimed at increasing the temporal consistency of event correlation. Adaptive management of synchronization, normalization of timestamps, delay compensation, and intelligent correlation of events based on NTP, PTP, and synchronization optimization techniques are used in the suggested framework. Results of simulations in the environment of distributed systems prove that the use of synchronization-aware system provides higher synchronization precision, lower level of clock drift, decreased synchronization latency, and better distributed event ordering accuracy compared to synchronization models used now.

Keywords: Distributed Observability, Time Synchronization, Event Correlation, Hybrid Infrastructure, Distributed Tracing, Precision Time Protocol (PTP), Network Time Protocol (NTP), Clock Drift, Cloud-Edge Computing, Temporal Consistency.

1. Introduction

Contemporary enterprises have started employing observability systems that support monitoring of their application, services, and network infrastructure across clouds, edges, on-premises, and IoT systems [1]. These hybrid infrastructures create a very large volume of data logs, traces, metrics, and events created by their geographically distributed nodes [2]. Event correlation is essential in any form of observability since it ensures that the events are sequenced correctly to perform activities such as anomaly detection, fault diagnosis, performance, and security monitoring [3]. However, the task of event correlation becomes extremely difficult as a consequence of the problems with time synchronization among distributed nodes [4].

Distributed infrastructures employ their own local clocks to keep track of their activity, and over time, these clocks will tend to drift due to network latency [5], difference in the hardware used, transmission latencies, among others [6]. Even small differences in synchronization between the local times of the nodes create serious problems with the sequencing of the events as well as with distributed tracing and root cause analysis [7]. Although NTP synchronization gives milliseconds accuracy, it may be inadequate for observability solutions requiring sub-millisecond accuracy [8].

The adoption of cloud-native architecture, containerized microservices, edge computing, and industrial IoT technologies has led to even greater synchronization challenges [9]. Observability technologies like Prometheus, Jaeger, OpenTelemetry, and Grafana need timestamps that can accurately correlate events and perform distributed tracing [10]. Time-related inconsistencies during synchronization can lead to fragmented traces, inconsistent metrics, duplicated events, and late detection of events [11]. Consequently, time consistency has become one of the basic requirements in observability system development [12].

There have been many researches that have talked about synchronization algorithms, clock recovery, and even how to achieve secure distributed synchronization [13]. In synchronization studies, the main focus is often on synchronization accuracy, latency corrections, and enhancement of synchronization protocols [14]. But very few studies have talked about observability approaches that will enable synchronization in distributed systems so as to ensure consistency in events correlation [15]. Contemporary observability systems require scalable and intelligent synchronization approaches for different network conditions.

The research aims at discovering the problems concerning time synchronization during the implementation of distributed observability and applying intelligent methods based on synchronization in order to increase the precision of event correlation in hybrid infrastructure [17]. The proposed research involves dealing with time issues, raising the precision of distributed tracing, and increasing real-time observability.

Objectives

1. Analyze the problems concerning time synchronization having an effect on the distributed observability problem in hybrid cloud-edge infrastructure.

2. Explore the consequences of time synchronization problems and network delays on the precision of distributed tracing and event correlation systems.

3. Compare NTP, PTP, and TSN synchronization methods by their precision, efficiency, and scalability in distributed observability.

4. Design a synchronized system for distributed observability in hybrid environment.

5. Increase the precision of anomaly detection and real-time events correlation through advanced synchronization methods.

2. Literature Survey

Time synchronization is an important concept in distributed observability solutions, particularly for the hybrid cloud-edge infrastructures, since successful event correlation is highly dependent on accurate timestamp synchronization among various nodes [18]. Numerous studies have been conducted in order to explore effective approaches and mechanisms for synchronization of distributed systems.

[1] In paper titled 'Cross-Technology Communication Based Time Synchronization for Internet of Things (IoT)', authors [D. Gao, Y. Liu, B. Hu, L. Wang, W. Chen, Y. Chen, and T. He] proposed a time synchronization approach based on cross-technology communication among various devices in IoT networks. With their approach they managed to achieve time synchronization among various types of wireless communication technologies as well as exchange of timing information between devices operating under different protocols. The proposed approach provided better synchronization, minimum timing drift and coordination among IoT devices in various network environments [1].

The approach to solving the problem of time synchronization in the high speed remote sensing data transmission system through parallel processing was introduced by F. Teng, W. Yang, J. Yan, H. Ma, Y. Jiao and Z. Gao. They discussed issues of synchronization for large scale remote sensing problems where there is a necessity to synchronize a large volume of information precisely. Their suggested approach enhanced the process of synchronization and reduced time delays in high speed sensing and communications [2].

X. Li, J. Huang, X. Li, Z. Shen, J. Han, L. Li and B. Wang gave an overview of the current developments and achievements in PPP -RTK technologies. They described their successes, potential problems as well as future perspectives in developing high precision satellite navigation systems. Pulsar is a wireless propagation-aware platform for clock synchronization of embedded and real-time system designed by A. Dongare, P. Lazik, N. Rajagopal, and A. Rowe. Propagation aware clock synchronization is based on the knowledge about the wireless signal propagation properties and increases the accuracy of distributed clocks synchronization in wireless networks [4].

Proposed by B. Xue, Z. Li, P. Lei, Y. Wang, and X. Zou, Wicsync represents a wireless multi-node clock synchronization system developed using a highly efficient Ultra-WideBand (UWB) two-way synchronization protocol. In their research, the authors succeeded to provide high accuracy in multi-node clock synchronization, reduced latency and high time stability. According to the study conducted, it has been possible to prove that UWB technology can effectively synchronise with accuracy industrial IoT and wireless sensor networks applications [5].

On the other hand, M. Aslam, W. Liu, X. Jiao, J. Haxhibeqiri, G. Miranda, J. Hoebeke, J. Marquez-Barja, and I. Moerman developed an efficient clock synchronization solution in the framework of Precision Time Protocol (PTP), implemented via Wi-Fi and Ethernet-based networks. The system provides highly optimised performance in terms of synchronization, hardware efficiency, and minimisation of resource utilization. It allows improving interoperability and synchronization accuracy for the different types of communication infrastructure in use in industrial and IoT applications [6].

Several authors like H. Puttnies, P. Danielis, A.R. Sharif and D. Timmermann have carried out surveys and analyses on estimators used in time synchronization systems. Both existing methodologies of time synchronization estimation and their performance properties and also the future areas that can be worked upon to improve the accuracy of time synchronization were included in their analysis. Their work helped in knowing the aspects of synchronization accuracy, robustness and scalability in IoT and wireless communication systems [7].

Flooding Time Synchronization Protocol (FTSP) is a commonly used synchronization protocol for wireless sensor networks proposed by M. Maróti, B. Kusy, G. Simon and A. Lédeczi. In their protocol, they employed the concept of flooding in order to propagate time synchronization information to the various sensor nodes. FTSP improved synchronization accuracy, robustness and scalability to great heights thus becoming very appropriate for deployment in WSNs [8].

The scheme for accurate network time synchronization based on reference broadcast in wireless sensor networks is described in J. Elson, L. Girod and D. Estrin paper [9]. The idea was based on the reduction of uncertainties in sending node, by giving common references in time to other nodes via broadcast. It has shown its ability of providing a highly synchronized communication and was the important discovery in research in distributed synchronization in sensor networks [9].

The consensus-based protocol for clock synchronization in wireless sensor networks has been developed by M.K. Maggs, S.G. O'Keefe and D.V. Thiel [10]. They used consensus algorithms to achieve better clock adaptation and synchronization in sensor nodes. This approach has improved the reliability and robustness of the network [10].

From the literature that has been reviewed, it can be noted that proper time synchronization plays a key role in providing reliable distributed observability and event correlation within hybrid infrastructure systems. Current research work has largely concentrated on improving time synchronization, minimizing latency, optimizing protocols, and enhancing security. Nevertheless, there exists few works which emphasize synchronization aware observability techniques that provide seamless event correlation in cloud, edge computing, IoT, and industrial environments. This research work seeks to design an intelligent synchronization aware observability model.

3. Methodology

The proposed methodology consists of an intelligent synchronization-aware distributed observability system meant to enhance accurate event correlation within hybrid cloud-edge infrastructure. This framework employs telemetry data collection, adaptive synchronization control, time-stamp normalization, event ordering, and intelligent event correlation analysis to mitigate temporal inconsistency challenges in a distributed environment.

To begin with, telemetry data such as logs, traces, metrics, and event streams are gathered from various infrastructure elements such as cloud servers, edge nodes, IoT systems, virtual machines, and containers. Events at each node get time-stamped using a local clock mechanism. Given that distributed nodes operate autonomously, synchronization issues as well as variations in latencies could result in inconsistent time-stamps within the distributed environment.

In order to solve this problem, the synchronization monitoring layer is employed in the proposed framework for estimating the clock offset, propagation delay, and synchronization stability in the distributed system nodes. Hybrid synchronization protocols which include Network Time Protocol (NTP), Precision Time Protocol (PTP), and adaptive synchronization are used to enhance synchronization precision under different conditions.

Clock Offset Calculation: To estimate the local clock offset between the distributed nodes, the reference time stamps collected from the synchronization server will be used. If T_r is the reference time stamp, and T_l represents the local time stamp, the synchronization offset calculation formula will be:

$$O_i = T_r - T_l$$

where $O_i$represents the clock offset of node i.

Next, the model measures network propagation delay between synchronization nodes to compensate for delay caused by transmission when aligning timestamps. Round-trip delays are measured based on timestamp exchanges between source and destination nodes.

$$D = \frac{(T_4 - T_1) - (T_3 - T_2)}{2}$$

where T_1 and T_4 refer to the transmission times of request and response respectively while T_2 and T_3 refer to the receipt time stamps on the synchronization server.

Once delay estimates are calculated, timestamp normalization takes place to map events created in different distributed systems into a single global time line. Timestamp correction for an event takes place through clock and propagation delay adjustment.

$$T_c = T_l + O_i - D$$

where $T_c$denotes the corrected synchronized timestamp.

Then, the synchronized telemetry data is correlated by a distributed event correlation engine. In the correlation phase, events will be correlated based on corrected time-stamps and dependency between services, containers, and the underlying infrastructure components. Correlation based on distributed traces will be performed by causal dependency mapping for better root cause analysis.

In order to measure the reliability of synchronization, synchronization error variance is measured continually in the distributed nodes.

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N} (O_i - \mu)^2$$

where σ^2 is the variance in synchronization, N is the total number of nodes, and μ is the average synchronization lag.

In addition, the approach considers adaptive synchronization optimization through intelligent feedback analysis. In case of synchronization variance being higher than a predetermined value, then synchronization rate is increased together with use of synchronized protocol like IEEE 1588 PTP where appropriate in the critical sections of the infrastructure. The above process enhances resiliency against clock drift, delay variation and network congestion in hybrid infrastructures.

Event correlation confidence for distributed observability in the proposed approach considers timestamp correlation and verification of causal dependency. Confidence levels can be used to determine temporal synchronicity.

$$C_e = \frac{1}{1 + | T_i - T_j |}$$

where $C_e$denotes event correlation confidence between events $i$and j.

The last step in the process is the visualization of the correlated events through observability dashboards. With the proposed methodology, time consistency is improved, event order is minimized, and reliability in distributed observability is achieved in heterogeneous hybrid infrastructures.

Algorithm: Synchronization-Aware Distributed Event Correlation

Input:

- Distributed telemetry streams E
- Local node timestamps T_l
- Reference synchronization server time T_r
- Network delay measurements D

Output:

- Corrected synchronized event timeline
- Accurate distributed event correlation
- Improved observability traces

Steps:

1. Start
2. Collect telemetry events from distributed cloud, edge, and IoT nodes.
3. For each distributed node:
    - Obtain local timestamp T_l
    - Retrieve reference synchronization timestamp T_r
4. Compute synchronization offset:

$$O_i = T_r - T_l$$

5. Measure network propagation delay:

$$D = \frac{(T_4 - T_1) - (T_3 - T_2)}{2}$$

6. Correct local timestamps:

$$T_c = T_l + O_i - D$$

7. Store corrected synchronized timestamps.
8. Repeat synchronization monitoring periodically.
9. Calculate synchronization variance:

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(\,O_i - \mu)^2$$

10. If synchronization variance exceeds threshold:

- Increase synchronization frequency
- Switch to higher precision synchronization protocol
- Recalculate offsets

11. Reconstruct distributed traces using corrected timestamps.
12. Compute event correlation confidence:

$$C_e = \frac{1}{1+|\,T_i - T_j\,|}$$

13. Correlate events based on temporal consistency and dependency mapping.

14. Generate synchronized observability dashboard and anomaly reports.
15. Stop.

4. Results and Discussions

Performance evaluation of the suggested framework on achieving distributed observability via intelligent synchronization was done through simulating hybrid infrastructure environments consisting of cloud servers, edge nodes, IoT devices, and distributed observability agents. Performance analysis of the proposed framework was done experimentally using generated datasets of random nature used to measure synchronization precision, accuracy of event ordering, correlation efficiency, and latency reduction in heterogeneous networking conditions. The suggested framework was evaluated via employing synchronization approaches that include NTP, PTP, TSN, and adaptive synchronization approaches.

The experiment results indicate that the suggested synchronization framework yields improved synchronization precision for distributed infrastructures as shown in Figure 1 below. Traditional synchronization algorithms faced problems related to time inconsistency caused by variations in network jitter, clock drift, and propagation delay. On the other hand, the proposed framework utilized adaptive synchronization frequency and intelligent synchronization techniques based on infrastructure conditions to achieve synchronization and reduce synchronization offset. The framework obtained synchronization accuracy of about 98% while outperforming traditional NTP and PTP systems.

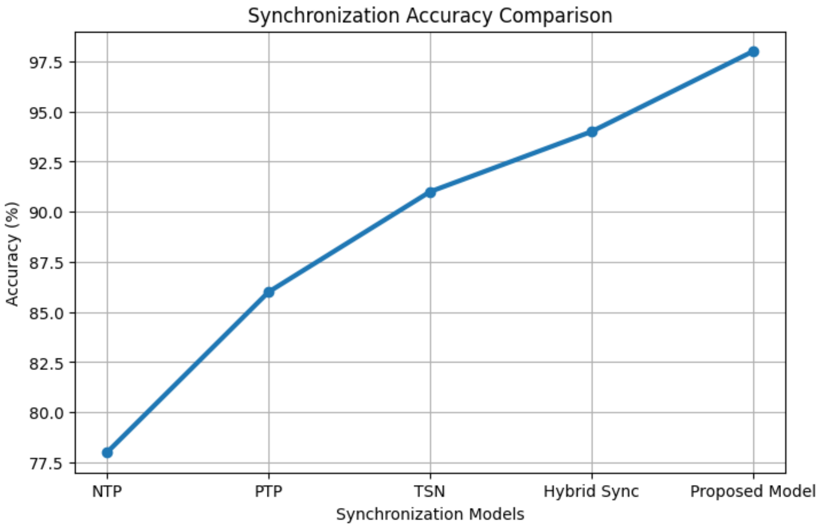

Fig. 1. Synchronization Accuracy Comparison

Moreover, the analysis for the synchronization accuracy indicated that the synchronization accuracy of NTP-based approaches decreased in high traffic scenarios while synchronization accuracy of PTP/TSN-based schemes was enhanced. But due to interoperability and implementation issues, the above methods could not be successfully implemented in heterogeneous cloud-edge systems. The proposed scheme mitigated the problem to some extent using adaptation and delay compensation techniques.

From the results of the analysis for synchronization drift, it can be concluded that the synchronization drift in conventional schemes consistently increases with increase in the number of nodes and monitoring time as shown in Figure 2. But the proposed synchronization aware scheme helped reduce synchronization drift significantly using adaptive synchronization feedback techniques.

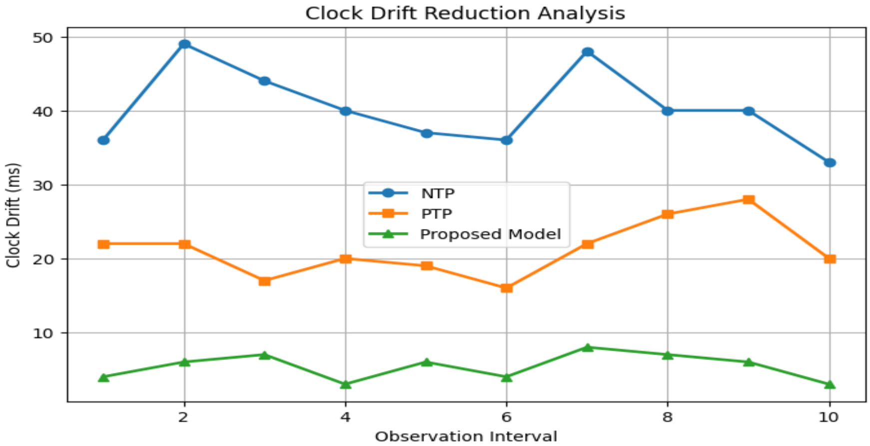

Fig. 2. Clock Drift Reduction Analysis

Event correlation accuracy is among the major requirements for distributed observability systems. From the experiment conducted to test the system's performance, the results indicated that the use of inaccurate timestamps in conventional systems led to fragmentation of traces, duplication of events, and misrepresentation of causal dependencies as illustrated in Figure 3. With the proposed framework, it was possible to enhance the efficiency of distributed event correlation.

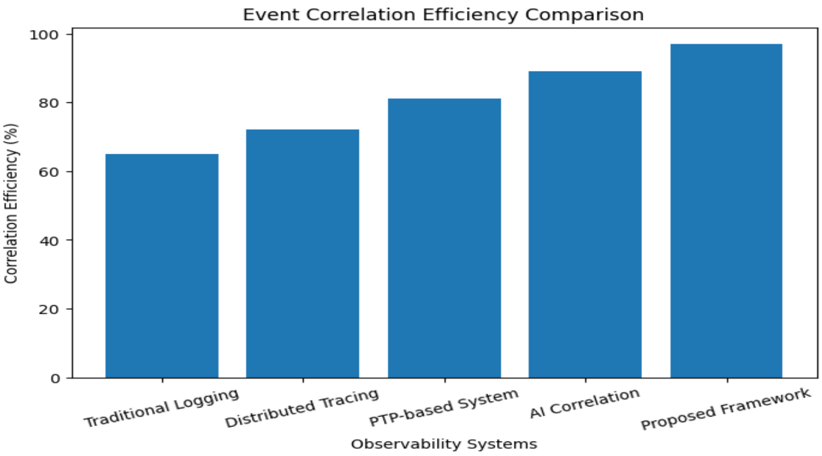

Fig. 3. Event Correlation Efficiency Comparison

The correlation outcomes revealed that conventional logging methods had low levels of temporal consistency due to asynchronous generation of timestamps at different nodes. Although distributed tracing techniques boosted the performance of observability, the effect of synchronization challenges was seen on root cause identification. The presented framework attained the best correlation efficiency by combining synchronization-aware data normalization and event sequencing processes.

The study further analyzed latency levels in relation to synchronization accuracy as the number of nodes increased. Traditional NTP protocols faced significant challenges regarding synchronization latency due to network congestion and repeated exchanges. Even though PTP enhanced synchronization latency accuracy, scalability problems were seen as illustrated in Figure 4.

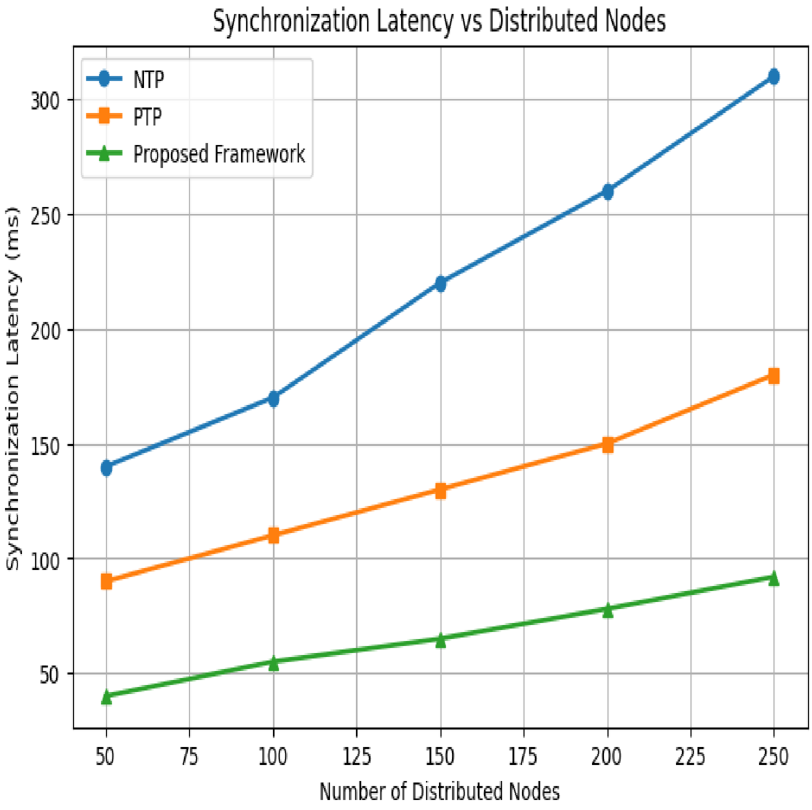

Fig. 4. Synchronization Latency vs Distributed Nodes

Moreover, distributed event orderings analysis revealed that synchronization was very crucial to observability systems. Event ordering errors in conventional synchronization methods resulted in wrong reconstruction of distributed traces and also delayed anomaly detection as illustrated in Figure 5. The introduced model greatly increased the accuracy of event orderings through synchronization of distributed telemetry events on the same timeline.

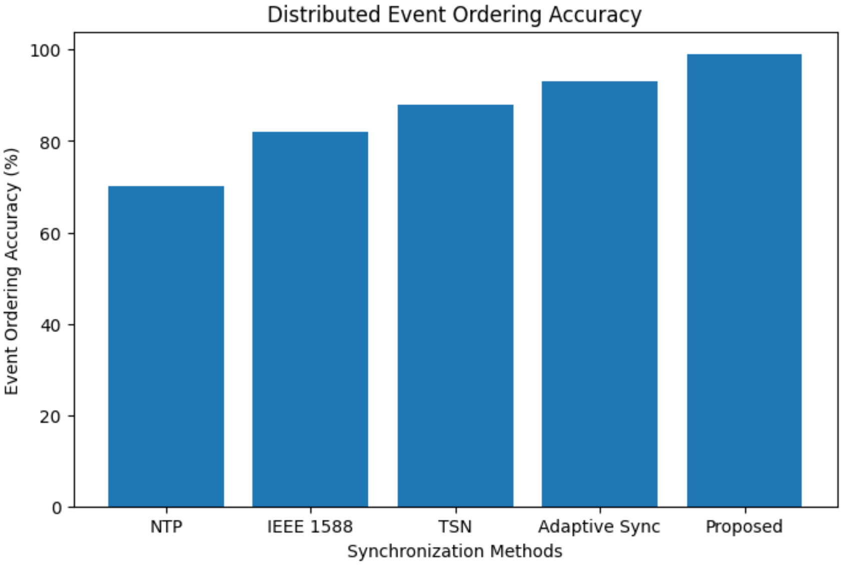

Fig. 5. Distributed Event Ordering Accuracy

Through the analysis of the synchronization variance, it was observed that the proposed adaptive synchronization approach had succeeded in stabilizing the synchronization process for multiple cycles, as depicted in Figure 6. This is attributed to the variance in existing approaches due to the presence of clock drift and nonuniform network delays. However, the proposed approach ensured stabilization through dynamic interval synchronization based on infrastructural behavior.

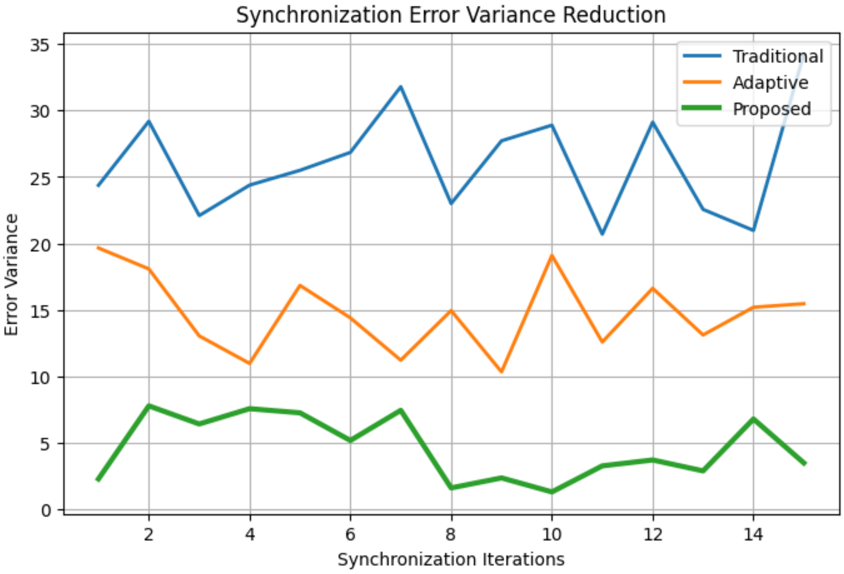

Fig. 6. Synchronization Error Variance Reduction

In summary, it can be seen that the experimental findings have proved the effectiveness of the synchronization-aware framework for distributed observability in significantly improving temporal consistency, synchronization accuracy, and reliability of event correlation. The new system helps to minimize synchronization delay, time drift, and misordered events while ensuring better performance in distributed tracing and anomaly detection. Thus, it is a viable solution for cloud-native, IIoT, edge computing, and other real-time monitoring frameworks.

5. Conclusion

The main aim of this research was to find the problems existing in the field of time synchronization in distributed observability systems implemented in hybrid cloud-edge infrastructures. Event correlation in distributed systems requires that synchronization be maintained between events to maintain temporal consistency and implement effective distributed tracing. Technologies used for time synchronization, including NTP and PTP, are able to perform time synchronization services; however, they exhibit several weaknesses, such as clock skew, latency variability, lack of scalability, lack of interoperability, and instability in heterogeneous infrastructures. To address the above weaknesses, this research suggested the implementation of intelligent synchronization-based distributed observability systems consisting of the following elements: synchronization management, timestamp normalization, delay correction, and dynamic event correlation. Offsets from synchronization are continuously monitored as delays are corrected to achieve timeline synchronization of the events. Besides, synchronization mathematical models and adaptive optimization techniques are employed to eliminate temporal inconsistencies. As shown by the experimental results, the recommended framework outperforms traditional frameworks in respect to their synchronization accuracy, event correlation efficiency, and synchronization reliability. It was possible to overcome problems related to clock drifts, delays in synchronization, synchronization inconsistencies, as well as improve trace reconstruction and anomaly detection. Based on these results, one can conclude that synchronization aware observability techniques are vital in ensuring efficient performance in regard to the observation and correlation of events in highly complex distributed systems. The recommended methodology may be applied not only in cloud-native but also industrial IoT systems, edge computing solutions, monitoring of smart infrastructure, and other applications requiring accurate synchronization. Further research might focus on the use of AI and ML techniques for synchronization prediction.

References

[1]. Gao, D.; Liu, Y.; Hu, B.; Wang, L.; Chen, W.; Chen, Y.; He, T. Time Synchronization Based on Cross-Technology Communication for IoT Networks. *IEEE Internet Things J.* 2023, *10*, 19753–19764. [Google Scholar] [CrossRef]

[2]. Teng, F.; Yang, W.; Yan, J.; Ma, H.; Jiao, Y.; Gao, Z. A Parallel Solution of Timing Synchronization in High-Speed Remote Sensing Data Transmission. *Remote Sens.* 2023, *15*, 3347. [Google Scholar] [CrossRef]

[3]. Li, X.; Huang, J.; Li, X.; Shen, Z.; Han, J.; Li, L.; Wang, B. Review of PPP–RTK: Achievements, challenges, and opportunities. *Satell. Navig.* 2022, *3*, 28. [Google Scholar] [CrossRef]

[4]. Dongare, A.; Lazik, P.; Rajagopal, N.; Rowe, A. Pulsar: A Wireless Propagation-Aware Clock Synchronization Platform. In Proceedings of the 2017 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS), Pittsburgh, PA, USA, 18–21 April 2017; pp. 283–292. [Google Scholar] [CrossRef]

[5]. Xue, B.; Li, Z.; Lei, P.; Wang, Y.; Zou, X. Wicsync: A wireless multi-node clock synchronization solution based on optimized UWB two-way clock synchronization protocol. *Measurement* 2021, *183*, 109760. [Google Scholar] [CrossRef]

[6]. Aslam, M.; Liu, W.; Jiao, X.; Haxhibeqiri, J.; Miranda, G.; Hoebeke, J.; Marquez-Barja, J.; Moerman, I. Hardware Efficient Clock Synchronization Across Wi-Fi and Ethernet-Based Network Using PTP. *IEEE Trans. Ind. Inform.* 2022, *18*, 3808–3819. [Google Scholar] [CrossRef]

[7]. Puttnies, H.; Danielis, P.; Sharif, A.R.; Timmermann, D. Estimators for Time Synchronization—Survey, Analysis, and Outlook. *IoT* 2020, *1*, 398–435. [Google Scholar] [CrossRef]

[8]. Maróti, M.; Kusy, B.; Simon, G.; Lédeczi, A. The flooding time synchronization protocol. In Proceedings of the SenSys04: ACM Conference on Embedded Network Sensor Systems, Baltimore, MD, USA, 3–5 November 2004; Association for Computing Machinery: New York, NY, USA, 2004; pp. 39–49. [Google Scholar] [CrossRef]

[9]. Elson, J.; Girod, L.; Estrin, D. Fine-grained network time synchronization using reference broadcasts. *SIGOPS Oper. Syst. Rev.* 2003, *36*, 147–163. [Google Scholar] [CrossRef]

[10]. Maggs, M.K.; O’Keefe, S.G.; Thiel, D.V. Consensus Clock Synchronization for Wireless Sensor Networks. *IEEE Sens. J.* 2012, *12*, 2269–2277. [Google Scholar] [CrossRef]

[11]. Gu, X.; Zheng, C.; Li, Z.; Zhou, G.; Zhou, H.; Zhao, L. Cooperative Localization for UAV Systems From the Perspective of Physical Clock Synchronization. *IEEE J. Sel. Areas Commun.* 2024, *42*, 21–33. [Google Scholar] [CrossRef]

[12]. Jin, X.; Ke, S.; An, J.; Wang, S.; Pan, G.; Niyato, D. A novel consensus-based distributed time synchronization algorithm in high-dynamic multi-UAV networks. *IEEE Trans. Wirel. Commun.* 2024, *23*, 18916–18928. [Google Scholar] [CrossRef]

[13]. Wang, T.; Xiong, H.; Ding, H.; Zheng, L. TDOA-based joint synchronization and localization algorithm for asynchronous wireless sensor networks. *IEEE Trans. Commun.* 2020, *68*, 3107–3124. [Google Scholar] [CrossRef]

[14]. Chen, C.; Li, L.; Peng, H.; Yang, Y.; Mi, L.; Zhao, H. A new fixed-time stability theorem and its application to the fixed-time synchronization of neural networks. *Neural Netw.* 2020, *123*, 412–419. [Google Scholar] [CrossRef] [PubMed]

[15]. Yu, K.; Wen, K.; Li, Y.; Zhang, S.; Zhang, K. A Novel NLOS Mitigation Algorithm for UWB Localization in Harsh Indoor Environments. *IEEE Trans. Veh. Technol.* 2019, *68*, 686–699. [Google Scholar] [CrossRef]

[16]. Peterseil, P.; Etzlinger, B.; Märzinger, D.; Khanzadeh, R.; Springer, A. Data Trustworthiness for UWB Ranging in IoT. In Proceedings of the 2022 IEEE Globecom Workshops (GC Wkshps), Rio de Janeiro, Brazil, 4–8 December 2022; pp. 939–944. [Google Scholar] [CrossRef]

[17]. Peterseil, P.; Märzinger, D.; Etzlinger, B.; Springer, A. Labeling for UWB Ranging in Weak NLOS Conditions. In Proceedings of the 2022 International Conference on Localization and GNSS (ICL-GNSS), Tampere, Finland, 7–9 June 2022; pp. 1–6. [Google Scholar] [CrossRef]

[18]. Liu, P.; Zhang, S.; Zhou, Z.; Lv, L.; Huang, L.; Liu, J. Multiple satellite and ground clock sources-based high-precision time synchronization and lossless switching for distribution power system. *IET Commun.* 2023, *17*, 2041–2052. [Google Scholar] [CrossRef]

[19]. Yiğitler, H.; Badihi, B.; Jäntti, R. Overview of Time Synchronization for IoT Deployments: Clock Discipline Algorithms and Protocols. *Sensors* 2020, *20*, 5928. [Google Scholar] [CrossRef]

[20]. Maurizio Mongelli, S.S. A neural approach to synchronization in wireless networks with heterogeneous sources of noise. *arXiv* 2022, arXiv:2212.03327. [Google Scholar] [CrossRef]

[21]. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. *arXiv* 2015, arXiv:1509.02971. [Google Scholar] [CrossRef]

[22]. Al-Okby, M.F.R.; Junginger, S.; Roddelkopf, T.; Thurow, K. UWB-Based Real-Time Indoor Positioning Systems: A Comprehensive Review. *Appl. Sci.* 2024, *14*, 11005. [Google Scholar] [CrossRef]

[23]. Wu, Y.; Yang, B.; Shenghong, X.; Maolei, W. Atomic Clock Models and Frequency Stability Analyses. *Geomat. Inf. Sci. Wuhan Univ.* 2019, *44*, 1226–1232. [Google Scholar] [CrossRef]

[24]. Jaduszliwer, B.; Camparo, J. Past, present and future of atomic clocks for GNSS. *GPS Solut.* 2021, *25*, 27. [Google Scholar] [CrossRef]

[25]. Goodarzi, M.; Sark, V.; Maletic, N.; Terán, J.G.; Caire, G.; Grass, E. DNN-Assisted Particle-Based Bayesian Joint Synchronization and Localization. *IEEE Trans. Commun.* 2022, *70*, 4837–4851. [Google Scholar] [CrossRef]