12-15-2020 2:00 PM

Intelligent and Low Overhead Network Synchronization over Large-Scale Industrial Internet of Things Systems

Pengyi Jia, *The University of Western Ontario*

Supervisor: Wang, Xianbin, *The University of Western Ontario*

A thesis submitted in partial fulfillment of the requirements for the Doctor of Philosophy degree in Electrical and Computer Engineering

Abstract

With the extensive development of information and communication technologies and vertical industry applications, industrial IoT (IIoT) systems are expected to enable a wide variety of applications, including advanced manufacturing, networked control, and smart supply chain, which all exclusively hinge on the efficient cooperation and coordination among the involved IIoT machines and infrastructures. The ubiquitous connection among IIoT entities and the associated exchange of collaborative information necessitate the achievement of accurate network synchronization, which can guarantee the temporal alignment of the critical information. To enhance the temporal correlation of heterogeneous devices in large-scale IIoT systems, this thesis aims at designing industry-oriented network synchronization protocols in terms of accuracy improvement, resource-saving, and security enhancement with the assistance of learning-based methods.

Initially, the real-time timestamps and historical information of each IIoT devices are collected and analyzed to explore the varying rate of the skew (VRS) at each enclosed clock. K-means clustering algorithm is adopted to organize the distributed devices into a few groups, and each of them is assigned with an optimized synchronization frequency to avoid potential resource waste while ensuring synchronization accuracy. Historical VRS values are further utilized as the identification of each clock for providing verification information so that the security against message manipulation attacks during network synchronization can be enhanced.

Moreover, a digital twin-enabled clock model is established by comprehensively investigating the characteristics of each clock with diversified operating environments. A cloud-edge-collaborative system architecture is orchestrated to enhance the efficiency of data gathering and processing. With the assistance of the accurate estimation generated by the digital twin model for each clock, the situation-awareness of network synchronization is enhanced in terms of a better understanding of the clock feature and necessary synchronization frequency. Mean-

while, since temporal information generated at each local IIoT devices are efficiently gathered at the edge devices, the effect of packet delay variation is significantly reduced while the synchronization performance under various network conditions can be guaranteed.

To further reduce the network resource consumption and improvement the performance under abnormal behaviors during network synchronization, a passive network synchronization protocol based on concurrent observations is proposed, where timestamps are exchanged without occupying dedicated network resources during synchronization. The proposed scheme is established based on the fact that a group of IIoT devices close to each other can observe the same physical phenomena, e.g., electromagnetic signal radiation, almost simultaneously. Moreover, multiple relay nodes are coordinated by the cloud center to disseminate the reference time information throughout the IIoT system in accomplishing global network synchronization. Additionally, a principal component analysis-assisted outlier detection mechanism is designed to tackle untrustworthy timestamps in the network according to the historical observation instants recorded in the cloud center. Simulation results indicate that accurate network synchronization can be achieved with significantly reduced explicit interactions.

Keywords: Network synchronization, low overhead, intelligent clustering, digital twin, edge-cloud collaborative computing, Industrial Internet of things systems, concurrent observation, reliability enhancement

Lay Summary

Industrial Internet of things (IIoT) plays a critical role in achieving the next industrial revolution, i.e., industry 4.0. A wide variety of applications ranging from advanced manufacturing to wirelessly networked control enabled by IIoT systems exclusively rely on the seamless collaboration among the IIoT devices. Clock synchronization, as one of the enablers to improve the temporal correlation among the distributed devices, should be exclusively designed for the industrial environment considering its intrinsic features. Although multiple synchronization protocols are already designed for residential IoT systems, the incurred high network overhead, insufficient achievable accuracy, and lacking security mechanisms will inevitably deteriorate the overall performance of the IIoT systems. Consequently, to further enhance the overall performance of the synchronization in IIoT systems, industry-oriented clock calibration protocols are designed in the thesis from three aspects. Initially, the varying rate of skew is thoroughly explored to find out the ideal synchronization strategy for each IIoT device in terms of synchronization frequency and security enhancement. Moreover, digital twin-assisted comprehensive clock models are established for the distributed clocks according to their timestamps generated under diversified operating environments. The synchronization accuracy and cost-efficiency can fully improved benefit from the better understanding of the clock behavior. Finally, a passive clock synchronization protocol based on common observations with enhanced security is designed to eliminate the dedicated network resource allocated for clock synchronization without sacrificing the synchronization accuracy.

To those whom I love

Acknowledgments

I would like to express my deepest appreciation to my supervisor, Dr. Xianbin Wang, for his guidance, patience, and encouragement in developing my research. It was his enlightening supervision that inspired me to explore novel research areas and broadened my views in the research area.

Sincere thanks to Dr. Cheng Li, Dr. Konstantinos Kontogiannis, Dr. Gerry Moschopoulos, and Dr. Jayshri Sabarinathan for serving as my thesis examiners and a critical reading of the dissertation. Their insightful advice and comments improve the quality of this dissertation.

Thanks go to all present and former members of our research group for the time that we spent both at work and after work. I would give my best regards for their success in both study and life. I would like to extend my thanks to all my friends at Western University. The last four years have been full of fun and warmth because of your accompany and friendship. As always, I wish to thank my parents, my girlfriend, and my family for their love, support, and encouragement throughout these years.

Contents

List of Figures

List of Tables

List of Abbreviations

5G	fifth-generation
AAL	advanced assembly line
AI	artificial intelligence
AMS	advanced manufacturing system
ARMA	autoregressive and moving average
BAN	body area network
CCH	chief cluster head
CH	cluster head
CM	cluster member
CN	coordinator node
CPS	cyber-physical systems
CS	clock synchronization
D2D	device-to-device
DaaS	digital as a service
DCS	distributed clock synchronization
DCSP	distributed clock synchronization protocol
DoS	deny of service
DSP	distributed synchronization phase
DT	digital twin
ICT	information and communication technologies
IIoT	industrial Internet of Things
IoT	Internet of Things
IN	interactive node
FOM	free-space outdoor model
FTSP	flooding time synchronization protocol
GPS	global positioning system
KF	Kalman filter

List of Abbreviations (Cont'd)

LOS	line-of-sight
LTE	long term evolution
M2M	machine-to-machine
MAC	medium access control
MSE	mean squared error
NMSE	normalized mean square error
NTP	network time protocol
OCP	offset compensation process
PC	principal component
PCA	principal component analysis
PDV	packet delay variation
PIN	potential interactive node
PTP	precision time protocol
QoS	quality of service
RSS	received signal strength
RSSI	received signal strength indicator
SIP	synchronization initialization phase
SPE	squared prediction error
SSC	smart supply chain
TDMA	time-division multiple access
UAV	unmanned aerial vehicle
URLLC	ultra-reliable low-latency communication
VRS	varying rate of skew
Wi-Fi	wireless fidelity
WSAN	wireless sensor and actuator network
WSN	wireless sensor network

Chapter 1

Introduction

1.1 Overview of Industrial Internet of Things Systems

The fast convergence of information and communication technologies (ICT) and vertical industrial applications is promoting the advent of the fourth industrial revolution and industrial Internet of things systems (IIoT), aiming at enhancing the efficiency, productivity, and reliability of the industrial applications [1] [2]. In virtue of the advanced enabling technologies in IIoT and cyber-physical systems (CPS), the ubiquitous connectivity, seamless interactions, and distributed collaborations among IIoT infrastructures lay the foundation of the fruitful evolution for traditional industrial systems. Figure 1.1 shows the typical architecture of a large-scale IIoT system, where the heterogeneous IIoT elements cooperate locally to constitute various distinctive subsystems with diverse collaborative applications. Some commonly adopted local subsystems in a large-scale IIoT system are summarized as follows in details:

- **Wireless sensor and actuator network (WSAN)** is an integrated system combining traditional control techniques, wireless sensor networks (WSN), and advanced computing technologies such as cloud and edge computing. The involvement of heterogeneous devices and time-sensitive control commands pose more stringent requirement on the seamless interaction and cooperation among the distributed devices. The ubiquitous sensing and omnidirectional data acquisition provided by the inexpensive and small-sized sensors that are massively deployed in WSNs can realize the real-time status feedback for each

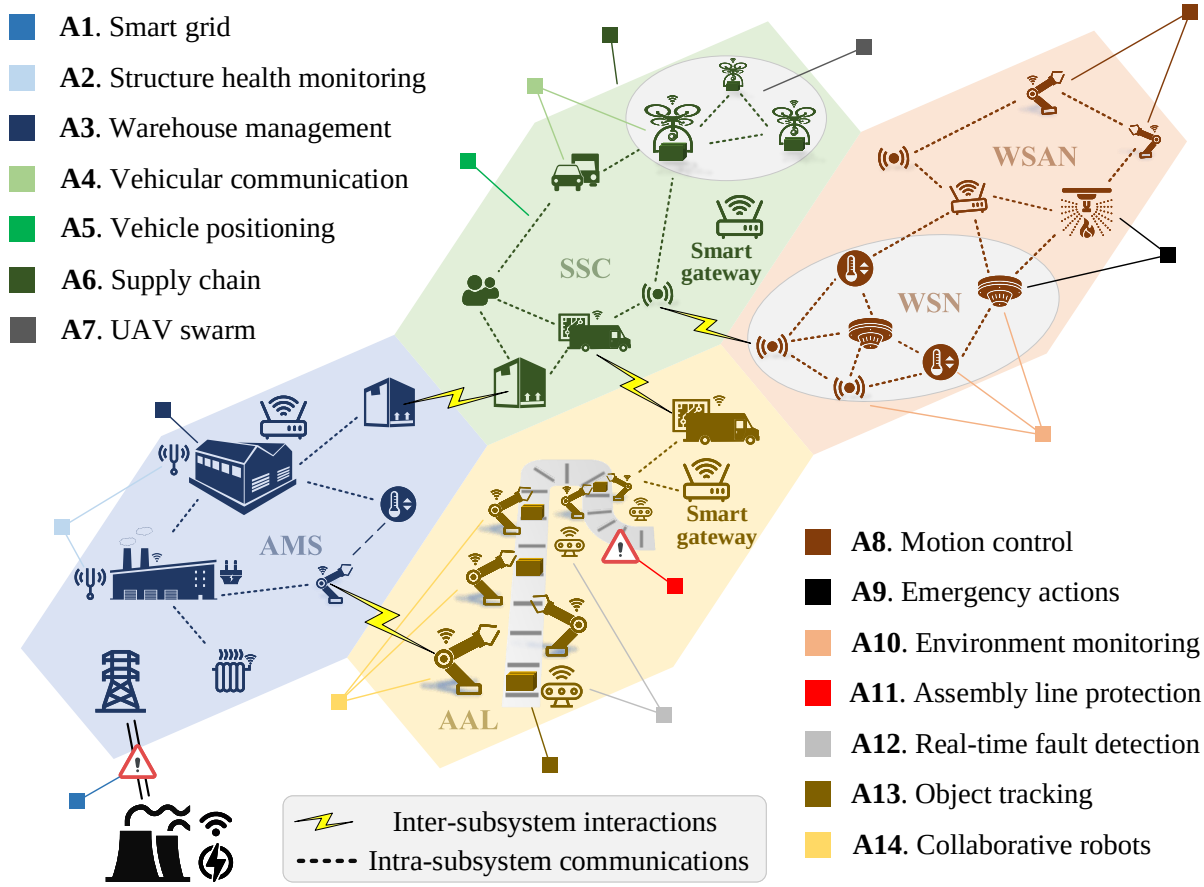


Figure 1.1: The overall structure of an IIoT system. The heterogeneous IIoT devices constitute diverse applications, which are enabled by different advanced techniques.

IIoT device as well as the thorough monitoring of its operating environment. With the involvement of controllers and actuators, WSANs are evolved from conventional WSNs with the expectation to execute more complicated tasks, including industrial automation, networked control, and emergency response. These tasks require accurate interactions between the various actuators and their operating environment, which necessitate a more robust system design, including the cohesive coordination among heterogeneous IIoT elements, efficient collection and transmission of the massive sensing data through limited network bandwidth, and reliable collaboration under uncertainties incurred from cheap devices [3].

- **Advanced manufacturing systems (AMS)** are more sophisticated applications involving the cooperation among devices and plants of distant locations as well as and dis-

tributed coordination for diversified industrial processes. In virtue of the innovative ICT and the advanced simulation technologies, including augmented reality and digital twin, several intelligent advancements are realized in traditional industrial manufacturing processes in the form of data-driven product design, optimized production plan, real-time manufacturing monitoring, and active preventive maintenance. The coordination among these processes and their associated heterogeneous IIoT entities exclusively hinge on the accurate and timely analysis of the data that spread through the IIoT system. Therefore, information sharing via cross-standard communication in diverse operating environments and the various uncertainties incurred thereby will inevitably lead to an increasingly challenging design of the IIoT system [4].

- **Smart supply chain (SSC)**, also referred to as supply chain 4.0, acts as one of the essential enablers in fully realizing industry 4.0. With the application of smart supply chains in industrial systems, the involved advanced technologies, e.g., pervasive sensing, big data analytics [5], and unmanned vehicles, are generating extensive benefits, including reduced costs, more responsive production, increased satisfaction, and enhanced time-saving [6]. The establishment of a complete SSC consists of two main components, namely, intelligently maintaining demand-supply balance and smart transportation for appropriate product delivery. As a consequence, designing an efficient data sharing scheme between the supplier and consumer, as well as the management of collaborative networking, which consists of vehicle-to-vehicle (V2V) communication and human-to-vehicle interactions among the involved conveyances and industrial entities, are two of the critical essentials during the SSC design [7].

- **Advanced assembly line (AAL)** differentiates from the traditional assembly line in terms of the efficient coordination among the remotely connected robotic arms in accomplishing highly productive assembly and the preventive intervention for potential assembling failures. Empowered by the advancement of sensing and communication abilities provided by edge devices, timely and reliable data sharing among the distributed actuators are feasible, while the usefulness of the data exchanged will depend on the consistency of the involved devices and the precise alignment of the temporal data, leading to

the necessity of extremely accurate clock synchronization. Meanwhile, appropriate precautionary intervention schemes enabled by recent technical advancements, e.g., neural networks (NN) [8] and artificial intelligence (AI) [9], are necessary to tackle against unreliability and potential attacks in the industrial systems for the AAL design so that huge expenditure and profit losses incurred can be avoided [10].

Meanwhile, efficient inter-subsystem interactions are critical to accomplish increasingly complicated industrial tasks, which require network-wide coordination among the involved infrastructures. As one of the prerequisites in cohering the distributed entities, accurate network synchronization serves as the backbone for various types of collaboration in large-scale IIoT systems. The extensive technological development will boost the machine processing precision and resolution of the associated data, leading to the ever-demanding requirements on the attainable synchronization accuracy for the cutting-edge industrial applications. For instance, centimeter-level positioning in advanced manufacturing systems (AMS) is necessary to efficiently support automated object tracking [11], which hinges on extremely accurate processing of the temporal data generated at distributed IIoT devices. In this case, network synchronization with sub-microsecond accuracy among the involved devices will be essential to support the tracking of critical targets and avoid potential manufacturing malfunctions. Considering the heterogeneity within IIoT systems in terms of the industrial constituents and communication standards, massive devices with stringent and individualized synchronization demands will interact through the sophisticated IIoT networks. As a consequence, accurate network synchronization will play an increasingly important role with the expansion of the network scale, which is indispensable in bridging the gap among different local networks to improve mutual comprehension and interoperability.

Conventional accurate clock synchronization techniques (e.g., Precision Time Protocol) exclusively hinge on the frequent and inflexible exchange of timestamps among the involved entities, which severely hinders the performance of the critical industrial applications due to overwhelming resource consumption and lack of intelligence during synchronization processes. The frequent synchronization-related data exchange will excessively occupy limited network resources reserved for industrial applications, leading to the deteriorated and even unattainable collaboration among the IIoT devices [12]. Moreover, adopting conventional rigid synchro-

nization methods without intelligently investigating the industrial circumstances will cause more challenging issues, manifesting as the increased synchronization frequency and uncontrollable synchronization error. Therefore, it necessitates novel industry-oriented strategies with reinforced efficiency and situation-awareness to accommodate the stringent requirements of network synchronization in challenging and dynamic industrial environments.

1.2 Challenges of Network Synchronization in Industrial IoT Systems

Accurate network synchronization is one of the critical prerequisites to ensure the temporal coherence among the distributed IIoT devices in achieving the collaborative performance of the diverse industrial applications. With the expansion of the network scale and the increasingly heterogeneous industrial infrastructures, many technical challenges are inevitably hindering the anticipated accomplishment of network synchronization in the large-scale IIoT systems, as demonstrated in Fig. 1.2. In this section, the critical challenges of conventional clock synchronization techniques for IIoT systems are summarized into three categories in detail.

- **Low Efficiency of the Offset-Driven Compensation Mechanisms:** The embedded clock driven by a crystal oscillator at each IIoT device will continuously generate a series of timestamps with its unique drift and offset compared to the standard time (e.g., UTC [13]), which is the principal reason behind periodic clock calibration. Conventional clock synchronization techniques fulfill the offset calibration demands through frequently exchanging timestamps between each target device and the time reference to guarantee the synchronization accuracy throughout the entire network, instead of analyzing the reason behind the ever-growing clock inaccuracy. This kind of offset-driven compensation scheme is achieved by passively responding to the observed clock inaccuracy, while the necessity of frequent clock calibration processes is barely alleviated. Consequently, severe issues will be caused during network synchronization due to the lack of proactive calibration mechanisms, mainly manifesting as the significantly reduced efficiency in large-scale IIoT systems.

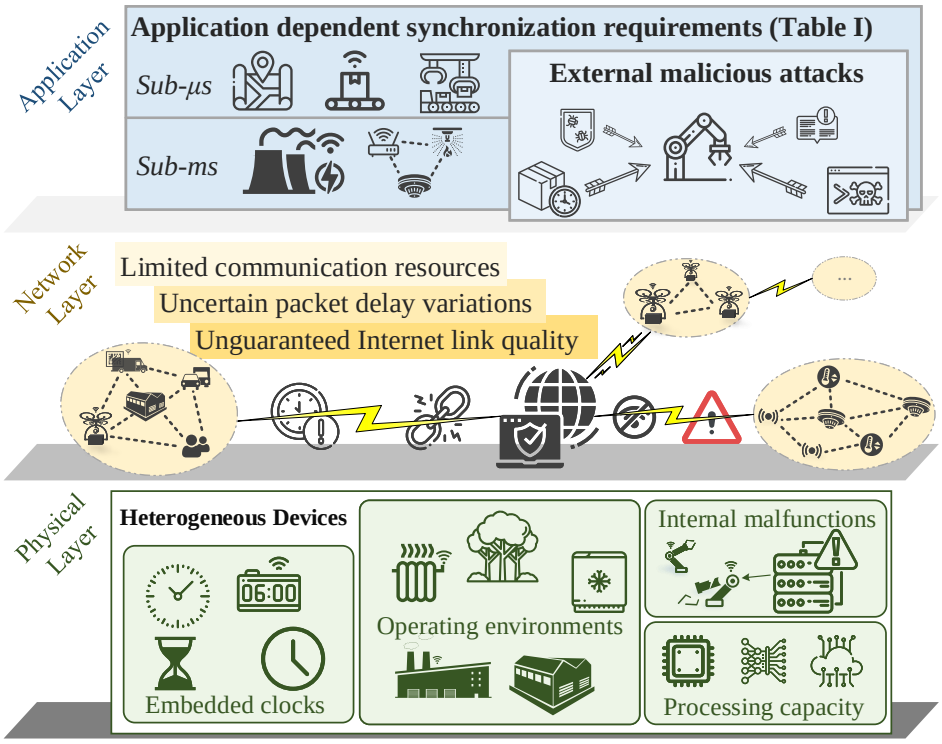


Figure 1.2: The Challenges encountered by traditional clock synchronization techniques in large-scale IIoT systems with three hierarchical layers.

Moreover, the observation of clock inaccuracy relies heavily on the explicit interactions and timestamps exchange among the involved IIoT devices. Due to the incomprehension of the characteristics in the distributed clocks, the unknown and diverse offsets throughout the network cannot be predicted by the conventional clock synchronization techniques, leading to the difficulties of selecting an opportune time for initiating the clock calibration process. The unoptimized network-wide exchange of timestamps will inevitably cause inappropriate synchronization frequencies for the distributed industrial devices. Some of the challenges may lie in the overwhelming occupation of the limited network resources in the case of excessive clock calibration [14], as well as the inaccurate and misaligned local data generation caused by insufficient synchronization. Additionally, the time-consuming interactions during clock offset estimation will occupy an increasing amount of limited communication resources with the expansion of the network scale, while the available resources for other critical industrial applications will be further suspicious.

- **Uncontrollable Timestamps Accuracy in Sophisticated Networks:** The increasingly sophisticated networks in large-scale IIoT systems pose more challenges in provisioning timely and trustworthy timestamps during network synchronization due to the highly complicated end-to-end communication process and the unsecured timestamps delivery [15]. On the one hand, conventional point-to-point clock calibration techniques are challenging to accommodate the stringent network synchronization requirements for industrial applications in terms of the anticipated accuracy. As a result of the extensive development of ICT technologies, large-scale IIoT networks generally comprise a wide variety of non-unified communication protocols, leading to the necessity of cross-standard interactions among the involved distributed entities. The best-effort service provided by Internet-based communications will induce unguaranteed communication conditions in large-scale networks, including the stochastic packet dropouts, increased access contentions, and dynamic routing of mesh networks. These critical issues will severely hinder the timeliness of the timestamps transmitted. Meanwhile, the increasing backhauls and multiple handshakes incurred by end-to-end timestamps exchange in the cross-network interconnections will accumulate excessively long latency, which will further deteriorate the timeliness and usefulness of the timestamps during network synchronization.

  On the other hand, the security threats at the vulnerable industrial devices incurred by the ubiquitous and collaborative interconnections in large-scale IIoT systems will influence the confidential provisioning of accurate timestamps for clock calibration, which becomes one of the fundamental considerations during network synchronization design [16]. The timestamps generated by the master clock and relayed by adjacent network nodes would be suspicious due to the susceptibility of resource-constrained IIoT devices to external interference and malicious attacks, such as spoofing, replaying, and faulty data injection [17]. The involvement of untrustworthy timestamps and malicious reference nodes will lead to uncontrollable and catastrophic results subsequent to the network synchronization, including unattainable collaborations for time-sensitive applications and the potential shutdown of industrial factories.

Table 1.1: Application-dependent synchronization requirements in a large-scale IIoT system

Functionality	Synchronization Accuracy	Applications shown in Fig. 1.1
Industrial automation [18] [19]	Sub-microsecond	A8
Localization & tracking [20]	Sub-microsecond	A4, A13
Multi-agent cooperation [21] [22]	Microsecond	A7, A14
	Tens of microseconds	A5
Data fusion [23] [24]	Tens of microseconds	A9, A12
	Sub-millisecond	A10
Fault protection [22]	Sub-millisecond	A1, A11
Remote monitoring [25]	Millisecond	A2
Smart logistics	Flexible	A3, A6

- **Complexity and Heterogeneity within Large-Scale IIoT Systems** A wide variety of collaborative applications involving intelligent machines and infrastructures forms the backbone of the large-scale IIoT systems. The highly complicated and heterogeneous features of IIoT systems will further degrade the network synchronization performance from several perspectives.

  Initially, the diversified industrial applications, which are enabled by the timely interaction within the local intra-subsystem communication and efficient interactions among the distributed subsystems, pose differential expectations on the synchronization accuracy, as summarized in Table 1.1. This requirement will vary extensively from sub-microsecond in critical applications (e.g., industrial automation) to a few milliseconds for less time-sensitive scenarios (e.g., remote environment monitoring). Moreover, due to the heterogeneous quality of the enclosed crystal oscillator at each IIoT device in terms of the initial manufacturing and stability regarding environmental variations, the distributed clocks will vary at different rates compared to the time reference. For example, the Simple Packaged Crystal Oscillators (SPXO) densely deployed in IIoT systems will drift hundreds of times faster than Oven Controlled Crystal Oscillator (OCXO) due to their low-cost and lacking appropriate techniques for eliminating temperature impacts [26]. Furthermore, the diverse and dynamic industrial operating environment of an IIoT system will further deteriorate the clock inconsistency throughout the network. As we previously studied [12], clocks in complicated circumstances are more suscepti-

ble to drift due to the temperature variation, which may be more challenging for devices assigned collaborative applications in dynamic environments, such as vehicular ad-hoc networks [27].

However, the conventional inflexible synchronization mechanisms always ignore these heterogeneous characteristics of the distributed IIoT devices while conducting unified clock calibration strategies with nonselective priority assignment and resource allocation. Therefore, the incurred inefficiency and resource-wasting inherent to the design deficiencies will inevitably become the technical bottleneck of accomplishing efficient network synchronization in large-scale IIoT systems with extensive heterogeneities.

1.3 Research Objectives of the Thesis

In this section, promising research directions are elaborated in three directions to design network synchronization schemes in tackling the existing challenges in large-scale IIoT systems.

- **Comprehensive Modeling-Enabled Proactive Network Synchronization:** Conventional clock synchronization techniques achieve clock calibration by directly adopting immutable empirical clock models during offset estimation, which will inevitably induce accumulated errors due to the modeling inaccuracy, especially for network synchronization with massive IIoT devices and multi-hop timestamp delivery. Meanwhile, as previously elaborated, the offset estimation process will occupy excessive network resources, which will be increasingly intolerable for large-scale IIoT systems.

  To address these deficiencies, we propose to comprehensively investigate the features of heterogeneous clocks during network synchronization in industrial environments instead of directly using inaccurate empirical models for offset estimation. As validated in the literature, the crystal oscillator enclosed at each IIoT device is uniquely affected by its intrinsic characteristics, among which the issues remained unsolved in low-cost IIoT devices are temperature sensitivity, manufacturing defect, and aging issues. Moreover, the extreme and dynamic temperatures inherent to inhospitable operating environments in distributed industrial systems will further exacerbate the instability of the crystal os-

cillator. Based on these observations, it would be critical to gather all available local information at each IIoT device during the comprehensive modeling phase. The collected inherent attributes, including the sequentially generated timestamps and the correlated environmental factors, can help to investigate the intrinsic features of each clock thoroughly. In virtue of the regression analysis assisted by model-based methods and machine learning techniques, the informative attributes of the distributed devices can be converted into useful clock models to guide the subsequent network synchronization.

The establishment of comprehensive clock models can be advantageous in enhancing the efficiency of network synchronization from two aspects. The intelligently formed clock models can be utilized to predict the real-time behavior of each distributed clock, which can assist in estimating the expected clock offsets throughout the network. Proactive network synchronization can be thereby designed by *predictively* eliminating the distributed clock offset to alleviate the necessary explicit interactions during clock calibration and dramatically reduce the incurred network overhead. Furthermore, the enhanced comprehension of the intrinsic clock characteristics at distinctive IIoT devices is indispensable in achieving the situation-awareness during network synchronization. for large-scale heterogeneous IIoT systems, this superior understanding can enable more efficient synchronization mechanisms in terms of the optimized synchronization-related network resources allocation and appropriate scheduling among complicated industrial applications.

- **Clustering-based Efficient Network Synchronization:** The main reason behind inefficient network synchronization lies in the inflexible exchange of timestamps without considering the heterogeneity within the large-scale IIoT systems in terms of the application-dependent synchronization requirements and clock characteristics. Additionally, lacking coordinators among concurrent calibration processes will further aggravate the efficiency of the overall network synchronization.

  Therefore, we propose to intelligently and adaptively assign synchronization frequencies for distributed IIoT devices by jointly exploiting the multi-dimensional intrinsic attributes. Typical synchronization-related attributes of an IIoT device include the initial

clock offset, clock skew, sequentially generated timestamps, assigned industrial applications, potential collaboration partners, operating environment parameters, and so on. By efficiently extracting meaningful attributes from the uploaded local information, comprehensive analysis can be conducted at devices with outstanding processing capacity (e.g., edge devices and cloud platform). Proper machine learning techniques can be adopted to enhance the processing efficiency, for instance, k-means clustering algorithms are powerful to accurately organize the distributed devices in a large network according to their associated multi-dimensional attributes. In virtue of the enabled intelligence, more rational decision-making can be timely generated to guide the optimized resource consumption for large-scale network synchronization. Specifically, devices with highly prioritized attributes (e.g., sub-microsecond accuracy requirement, unstable crystal oscillator, and inhospitable operating environments) should be assigned with much higher synchronization frequency, while other IIoT devices can be synchronized less recurrently to reduce the overall network overhead. Meanwhile, network resources can be preferentially allocated to industrial devices with more stringent synchronization requirements, which will be beneficial in enhancing holistic synchronization performance. Consequently, an intelligent scheme with an individualized synchronization frequency assignment and well-coordinated message exchange schedule can be established to enhance the efficiency of network-wide synchronization.

- **Enhanced Timestamps Provisioning with Uncertainty Control:** As previously discussed, the reliable provisioning of timestamps during synchronization is severely hindered by various uncertain issues induced from the intermediate communication process, including varying channel conditions, packet delay variations (PDV), and external malicious attacks. Different controlling and filtering approaches can be adopted to address specific issues.

  The network uncertainties during packet exchange will degrade the timeliness of the associated timestamps, leading to undesired synchronization performance. However, this kind of effect can be mitigated or eliminated by designing dedicated communication protocols for network synchronization. For example, an entirely controlled communication

channel is achieved in [28] by reserving intermediate nodes for synchronization data relaying, which can minimize the redundant processing during timestamps transmission. Meanwhile, a prioritized data transfer protocol [29] is designed to alleviate the effect of communication randomness by assigning higher priority to synchronization-related data and guarantee corresponding channel access. These methodologies are effective in proactively reducing the influence of network uncertainties prior to synchronization in dedicated communication environments.

Furthermore, the unreliable issues due to the vulnerability of resource-constrained IIoT devices in terms of inner malfunctions and external malicious attacks can be further addressed by filtering untrustworthy timestamps according to the timely analysis of historical information. Model-based filtering mechanism can be designed to detect unreliable timestamps and malicious devices according to the local clock information collected at each device. With increasingly massive timestamps involved in large-scale IIoT systems, other machine learning techniques (e.g., principal component analysis) can be considered to enhance the detection accuracy and efficiency during network synchronization.

1.4 Technical Contributions of the Thesis

The main contributions of this thesis are summarized as follows:

- By periodically collecting local timestamps from distributed devices, the varying rate of skew (VRS) of each clock is analyzed based on the historical time information for two different purposes. On the one hand, the distributed devices are clustered into several groups with the assistance of a chief cluster head based on their different VRS by adopting the K-Means clustering algorithm, while each group is assigned with a cluster head for providing time reference and a unique synchronization frequency where a group of devices with larger VRS will be synchronized more frequently. By adopting the proposed intelligent clustering-based scheme, the number of required packets used during network synchronization is significantly reduced under the permission that the synchronization accuracy can be guaranteed. On the other hand, a VRS information-

based fault detection algorithm is proposed to enhance the security of the network during network synchronization against potential malicious attacks. Historical VRS values are recorded and processes at the cluster heads to detect if any node is potentially malicious. A second-order autoregressive model is successively used to further identify the untrustworthy device to ensure the detection accuracy in the case that multiple devices are malicious. A distributed network-wide synchronization is finally achieved for IIoT systems with enhanced reliability and reduced network overhead on the promise that the synchronization accuracy is acceptable.

- To comprehensively investigate the characteristics of each clock under diverse operating environments, a digital twin model is established for each local clock with the assistance of the edge devices and the cloud center, where a cloud-edge-collaborate system is organized to boost the efficiency and accuracy of the data processing. To be more specific, the timestamps generated at each local device are efficiently recorded and processed at each assigned edge devices to minimize the modeling error induced from the network uncertainties, e.g., packet delay variations. The effect of asymmetrical network delay can be mostly eliminated to ensure the accuracy of clock modeling as well as the further synchronization actions. Moreover, the features of each clock, including the initial clock skew, temperature sensitivity factor, and ideal operating temperature, are thoroughly investigated and modeled based on the series of timestamps generated, while a digital twin of each clock is established and updated continuously. Based on the model formed, the real-time clock skew for each device can be estimated efficiently, so that accurate network synchronization can be achieved with little network resource consumption under diverse network conditions.

- To further reduce the network overhead induced from explicit interactions among distributed devices during network synchronization, a passive network synchronization scheme based on concurrent observations is proposed for industrial IoT systems to achieve network synchronization without dedicated network resources consumption for delivering timestamps. More specifically, due to the fact that a group of devices close to each other can observe the same physical phenomena nearby with a slight time difference,

i.e., observation bias, these devices can be synchronized with the same time reference by recording and analyzing the commonly observed phenomena and the associated time instants. The received signal strength value of each device on observing the common signal is recorded and analyzed to compensate for the observation bias so that the synchronization accuracy is further improved. Finally, a principal component analysis algorithm is adopted to detect untrustworthy timestamps and enhance the reliability of devices in the network according to the historical observation instants recorded in the cloud center. Explicit interaction-free network synchronization is finally accomplished for large-scale IIoT networks with the assistants of reliable relay nodes and trustworthy time references.

1.5 Thesis Outline

The remainder of this thesis is organized as follows:

In Chapter 2, a comprehensive study of clock synchronization in industrial IoT systems is conducted. An overview of the technical background of clock synchronization is given first, followed by the current challenges of clock synchronization in IIoT systems, including the cost-efficient and secure synchronization protocol design as well as the clock uncertainty induced from the operating environment. As one of the most important enablers to achieve accurate and intelligent clock synchronization in IIoT systems, a survey on digital twin related to our research is given at the end.

In Chapter 3, a distributed clock synchronization scheme enabled by intelligent clustering is proposed for local area industrial IoT systems. The overall synchronization protocol is comprised of two phases, namely, the synchronization initialization phase (SIP) and the distributed synchronization phase (DSP). Specifically, clocks are firstly synchronized with the reference node in SIP to eliminate the initial clock drift. With the assistance of the K-means clustering algorithm, devices in the network are intelligently clustered based on the varying rate of skew, and different synchronization frequencies are assigned to each cluster to achieve distributed clock synchronization. Moreover, a two-tier fault detection mechanism is proposed to determine malicious devices. Finally, simulations are conducted in MATLAB to demonstrate the effectiveness of the proposed protocol in terms of packets delivery reduction and malicious

nodes detection, as compared to the traditional simultaneous synchronization protocol.

A digital twin enabled distributed synchronization scheme is proposed in Chapter 4 to achieve an intelligent clock synchronization for reducing resource consumption associated with distributed synchronization in fast-changing IIoT environments. To be more specific, a digital-twin-enabled clock model is initially established at the remote location to characterize each involved clock based on the timestamps generated at different operating environments. By utilizing the formed model, each local device can estimate its clock skew under dynamic operating environments in a real-time manner so that its clock skew can be calibrated accordingly to achieve an accurate clock synchronization without much resource consumption. The simulation results demonstrated that the proposed scheme can create an accurate model for each clock while the clock accuracy and cost-efficiency during clock synchronization are significantly enhanced in different communication networks, as compared to the existing studies.

In Chapter 5, a passive clock synchronization scheme based on concurrent observations is proposed for industrial IoT systems. Based on the fact that a group of devices near the same antenna can receive the broadcasting signal at almost the same time, passive clock synchronization is achieved by processing the commonly observed signals with the assistance of the selected coordinator node. Received signal strength values are analyzed to reduce the effect of propagation latency so that the synchronization accuracy is further improved. Moreover, the principal component analysis algorithm is adopted to enhance the overall security of the proposed scheme by analyzing the historical timestamps uploaded to the cloud center. Finally, simulations are conducted to evaluate the performance of the proposed protocol in terms of clock accuracy and security during the network synchronization.

Finally, all the contributions are summarized in Chapter 6, with the identification of future research directions.

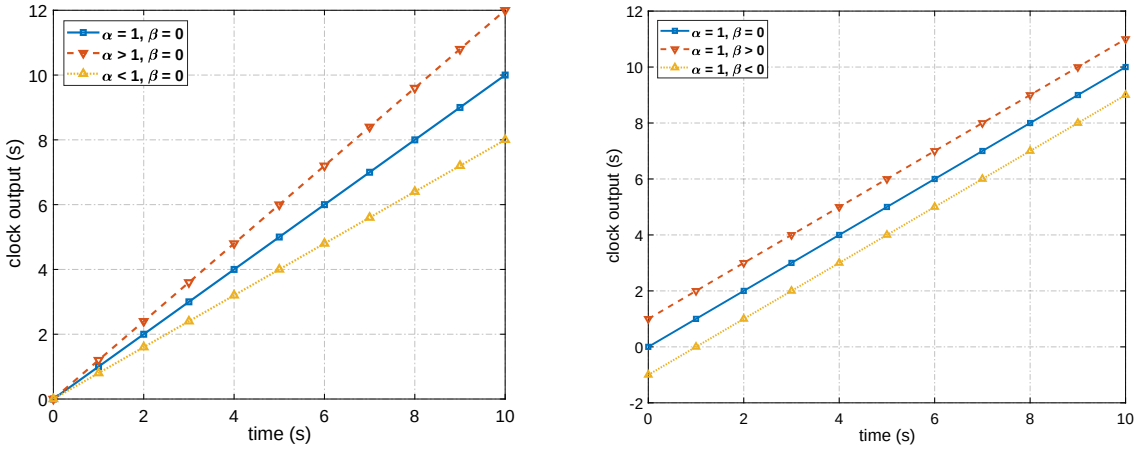
Chapter 2

Challenges and Existing Solutions of Clock Synchronization in IIoT Systems

2.1 Background of Clock Synchronization

Accurate clock synchronization is one of the critical prerequisites for cohesive collaboration in an IIoT system, in which a large number of IIoT devices are assigned to accomplish time-sensitive applications that hinges on the precise time alignment of the packets exchanged. Without coherent time information provided by these widely distributed IIoT devices, the efficiency and productivity of these industrial applications will be significantly degraded. Typically, due to the fact that the clock output of each device at the same time is not identical, the observed clock error will lead to misalignment of the transmitted packets. This time difference of each device is caused by the uniqueness of its embedded quartz-crystal oscillator, which can continuously generate time information with a unique frequency. Due to the uncertainty and dynamic of the frequency of each device, frequent clock synchronization is necessary to maintain time consistency throughout the network.

Ideally, the clock output generated by any device, denoted by $C(t)$, should be identical to the real time t, i.e., $C(t) = t$, so that each device will provide a consistent time index that associate with the critical data. However, due to the uniqueness of the oscillation frequency of each device as well as its manufacturing defect [30], the clock output generated at the device i will be fluctuating with time in reality. The clock behavior is typically modeled by a simplified



(a) The clock output is affected by the clock skew, where clock skew greater than 1 will make the clock run faster than the ideal case while smaller clock skew will lead to clock lag.

(b) The initial offset will lead to clock shift compared to the ideal clocks. Together with the clock skew, clock output will be more in accurate in terms of the slope and y intercept.

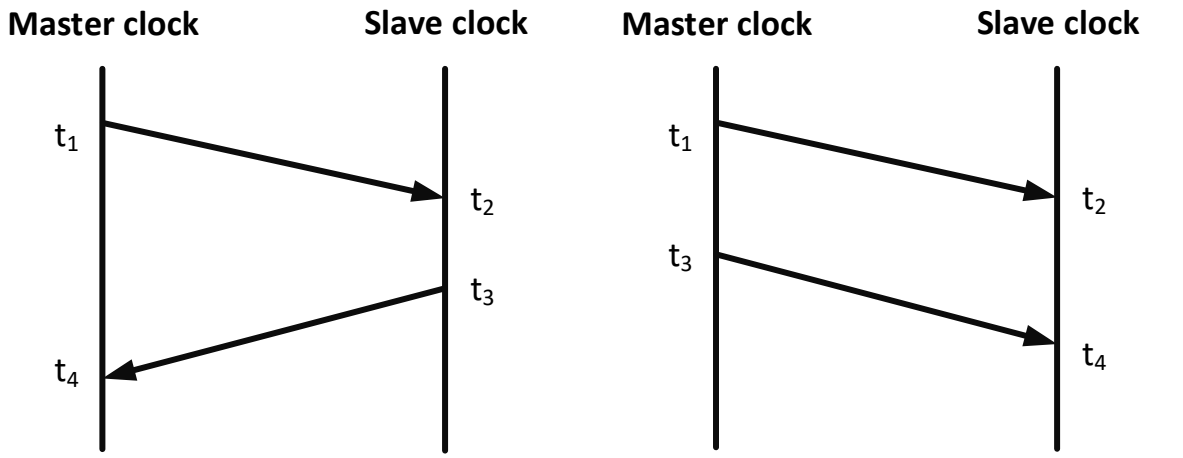
Figure 2.1: The evolution of clock information under the effect of initial clock skew and offset compared to the ideal clock.

linear function of its time-varying clock skew α_i and constant clock offset β_i [31], given by

$$C_i(t) = \alpha_i t + \beta_i \tag{2.1}$$

where the clock skew α_i will lead to the different increasing speed of the clock output while clock offset β_i will cause initial clock inaccuracy. The effect of clock skew and offset are shown in Fig. 2.1a and Fig. 2.1b, respectively. It can be observed that the clock skew, i.e., the slope of clock output, will lead to a faster clock output in the case that $\alpha_i > 1$. As a consequence, the corresponding clock information generated will be larger than the ideal value, especially in long-term operation. The accumulated clock error will eventually cause significant temporal misalignment and degraded overall performance of the IIoT system. By contrast, the initial clock offset β_i will lead to constant clock error compared to the ideal case, shown as Fig. 2.1b, where the clocks are set with a unit clock skew and different clock offsets. Although this offset will not grow with time, the initial offset is typically much greater than the clock skew, meaning that a more observable clock error can be induced if this initial offset is not eliminated by clock correction processes.

According to the degree of clock correction, the calibration of each clock consists of three



(a) The clock offset is corrected by exchanging a pair of bidirectional packets to eliminate the effect of propagation delay.

(b) The relative clock skew can be estimated by transmitting a pair of packets in the same direction between two devices.

Figure 2.2: Exchange of packets during clock synchronization, where the skew and offset are calibrated with two successive pairs of packets.

categories, namely, clock skew compensation only for clock skew cancellation, clock offset calibration responsible for eliminating the initial offset, and a combination of the former two, which can generate a more stable clock performance with higher synchronization cost. Moreover, the process of clock correction can be achieved by two approaches in terms of their synchronization purposes. On the one hand, each clock in the network can be synchronized with a selected reference clock by exchanging packets that containing successive timestamps. In this case, all devices will finally generate identical time information as compared to the reference clock. On the other hand, instead of selecting a reference clock as the standard, each device will exchange its local time information with its neighbors so that a consensual virtual clock can be formed at each node. This approach is commonly adopted in wireless sensor networks (WSN), where a large number of sensors are densely deployed in a network for information sensing and sharing. Since our study is accomplished by the packet-switch-based synchronization approach, details of the consensus-based synchronization will not be introduced in this thesis.

Two-way packet exchange is a commonly adopted approach for a pair of devices to efficiently share their local timestamps during point-to-point clock synchronization. Generally, a thorough clock synchronization can be achieved by exchanging two different pairs of packets

between every two devices for skew compensation and offset calibration, respectively. The processes of packets exchange between two devices during clock synchronization are shown in Fig. 2.2, where the master clock i serves as the time reference while the slave clock j is embedded at the device that requires to be synchronized. The purpose of the first pair of round-trip packet exchange, which is shown in Fig. 2.2a, is aimed at achieving offset correction with the elimination of network latency. To be more specific, the propagation delay between the two devices can be estimated by calculating the round-trip latency based on the timestamps recorded at node i and j. The relationship between the propagation latency and the timestamps recorded can be written as

$$d_{ij} = t_2 - t_1 - \beta_{ij} \tag{2.2}$$

and

$$d_{ji} = t_4 - t_3 + \beta_{ij} \tag{2.3}$$

where d_{ji} is the propagation latency when a packet is transmitted from the opposite direction of d_{ij}. Consequently, by assuming the propagation delay is symmetric and averaging the propagation delay shown in 2.2 and 2.3, it can be obtained that

$$d_{ij} = \frac{d_{ij} + d_{ji}}{2} = \frac{(t_2 + t_4) - (t_1 + t_3)}{2} \tag{2.4}$$

Then, the relative clock offset between the two devices can be estimated based on the estimated propagation delay, given by

$$\beta_{ij} = t_2 - t_1 - d_{ij} = t_3 - t_4 + d_{ij} \tag{2.5}$$

Meanwhile, the second pair of packets exchanged shown in Fig. 2.2b is responsible for estimating the relative clock skew between the two nodes. It is straightforward to calculate the relative skew according to the local clock variation of the two devices, given by

$$\alpha_{ij} = \frac{\alpha_i}{\alpha_j} = \frac{t_3 - t_1}{t_4 - t_2} \tag{2.6}$$

Therefore, the simplest way to achieve clock synchronization that includes skew compensa-

tion and offset correction can be accomplished by adopting 2.5 and 2.6 successively. However, the complicated IIoT system in terms of harsh operating environments and sophisticated networks will hinge the synchronization performance and post more stringent requirements on the synchronization protocol design.

It can be observed that the previous assumption, i.e., the propagation latency between two nodes is symmetric, is not always true, especially in a complicated network with a large number of devices to be synchronized. In such a system, numerous issues will lead to packet delay variations (PDV) [32], ranging from the physical transmission medium (e.g., the fiber asymmetry and fluctuating communication environments) to non-deterministic network parameters (e.g., the unbalanced access contention and the varying latencies caused by the intermediate buffers). Consequently, PDV will inevitably induce calculation error and synchronization inaccuracy, which should be further considered during the synchronization protocol design.

2.2 Current Challenges and Solutions of Clock Synchronization

Traditional point-to-point clock synchronization exclusively relies on the exchange of packets containing timestamps used for clock calibration. For a large-scale IIoT system, the clock synchronization design will be more challenging due to the existence of a large number of devices and the sophisticated operating environments. Meanwhile, the unique features of industrial applications also pose novel design challenge as compared to the conventional techniques. The increment of the challenge in designing the synchronization protocols for an IIoT system arises from three aspects. Initially, the performance of clock synchronization in an industrial environment should be enhanced in terms of the achievable synchronization accuracy and network resource consumption during synchronization. These two criteria determine the fundamental performance of the synchronization protocol and its compatibility with the industrial environment. Moreover, the robustness and security of the protocol designed should be considered. As IIoT devices are typically vulnerable to external attacks, proper security mechanisms should be designed to ensure not only the synchronization accuracy but also the overall perfor-

mance of the IIoT system. Finally, the operating environments for industrial applications are much harsher than office and residential circumstances, leading to the necessity of designing a situation-aware clock synchronization protocol that is effective in industrial environments. State-of-art studies aimed at addressing the previously mentioned challenges are introduced in the following three subsections.

2.2.1 Accurate and Cost-Efficient Clock Synchronization

Accuracy is the most basic requirement and one of the most fundamental criteria when designing a clock synchronization protocol. With the increment of the achievable synchronization accuracy, the cohesion of the overall IIoT system can be increasingly enhanced from a wide range of perspectives. On the one hand, the collaboration among the distributed IIoT devices relies on accurate clock synchronization. For example, automation in IIoT systems requires accuracy of $1\mu s$ as the prerequisite. On the other hand, a higher synchronization accuracy can enhance the performance of traditional applications. Taking the data acquisition during manufacturing as an example, a clock synchronization with at least $1ms$ is required to achieve the collection tasks. However, with higher accuracy, the data correlation gathered from distributed devices can be further improved so that the performance of the manufacturing will be potentially enhanced.

As a consequence, many works aimed at achieving accurate clock synchronization had been proposed in the literature for different systems ranging from wireless sensor networks (WSN) to IoT systems. As traditional precision time protocol (PTP) is only designed for estimating the clock offset between the slave clock and master clock, some protocols are designed to enhance the performance of PTP through different approaches. The author in [33] considered improving PTP by incorporating clock drift factors based on the timestamps received, while a mechanism for detecting the start of the frame is designed to further enhance the decoding during clock synchronization. Similarly, Kalman filters are adopted to estimate and correct the clock drift and offset in PTP-based industrial networks [34]. The performance of the networks with long linear topology is further improved. The clock accuracy can be increased at the cost of a low convergence rate during clock synchronization. Moreover, many synchronization mechanisms

are designed by considering different estimation algorithms. An adaptive time window linear regression method is designed in [35] for WSNs, where the synchronization period and the window size of the linear regression algorithm are adjustable. The accuracy and computational cost are decreased since the proposed scheme can adaptive to the environmental variations. Meanwhile, authors in [36] designed an accurate and scalable synchronization approach in 802.11-based networks. A high synchronization accuracy is achieved by adopting an adaptive protocol that can adjust the priority and frequency of beacon transmission during clock synchronization. Furthermore, due to the reason that network uncertainties, e.g., PDV, have a dramatic effect on the synchronization accuracy, maximum consensus and bounded theory are designed in [37] to achieve highly accurate clock synchronization in WSN under bounded noise. The performance of the clock synchronization is claimed to be more accurate and faster as compared to the existing methods.

However, most of the previously mentioned synchronization methods typically achieve accurate synchronization at the cost of higher network overhead during clock synchronization. As mentioned previously, the communication and computation resources in an IIoT system are typically very limited, leading to the necessity of designing a cost-efficient synchronization protocol. The cost-efficiency of a clock synchronization protocol depends on the overall resource consumption during the clock synchronization, while a smaller amount of resource consumption can enhance the overall performance of the IIoT system since more resources can be allocated to the critical industrial processes. Consequently, some works have been proposed in the literature by designing cost-efficient or energy-efficient synchronization methods. Authors in [38] proposed a recursive time synchronization protocol to achieve both accurate and energy-efficient clock synchronization for WSNs. The MAC-layer timestamping techniques and propagation delay compensation are adopted to enhance the accuracy of the timestamps, while infrequent broadcasts and synchronization requirement aggregations are further utilized to decrease the resource consumption during the clock synchronization. Similarly, a spanning tree is formed based on the selected backbone sensor nodes so that sender-to-receiver and receiver-to-receiver protocols can be combined to reduce the necessary resource used for clock synchronization. Different from the previously mentioned methods, some works focus on the approaches about estimating the clock behaviors. Authors in [39] proposed a lightweight syn-

chronization method by designing a maximum likelihood estimator that can estimate the clock features based on timestamps collected. The overhearing mechanism is further designed to avoid the increment of the network resource for synchronizing devices outside the range of the master clock. Meanwhile, a generalized maximum likelihood estimator (GMLE) method is adopted to estimate the clock parameters, and an easily implementable synchronization protocol is obtained [40]. The network overhead can be reduced once a good estimation is available. Furthermore, the synchronization problem is formulated as a closed-loop control problem in [41], while a proportional-integral controller is designed to compensate for the clock skew. The overall protocol can save more resources and achieve higher robustness to external disturbances.

Some of these synchronization protocols can achieve good performance in terms of achievable clock accuracy and energy-efficiency. However, almost all of these methods are designed without considering the reason behind clock errors. Instead, all devices are synchronized periodically or in an error-driven manner so that the synchronization would be inefficient and resource-wasting. Moreover, some of the works are not suitable for industrial environments, which would be harsher and more complicated compared to hospitable environments, e.g., residential spaces. Therefore, novel synchronization methods with enhanced intelligence are required for sophisticated IIoT systems.

2.2.2 Malicious Attacks against Clock Synchronization

In an Industrial IoT system, devices are vulnerable to a wide variety of attacks from external devices aiming at different purposes ranging from stealing information to crashing the overall IIoT system. These attacks can be classified based on different criteria, while a generally recognized classification method is to sort the malicious behaviors according to the communication layers that the attacks are against. For example, malicious behaviors in the physical layer may include eavesdropping, external interference, malicious data injection, Sybil attack, etc. These attacks can affect the behavior of the IIoT devices as well as their associated IIoT data. The eavesdropping or injection of the data will inevitably cause information leakage or confusion, leading to a fatal effect on the overall IIoT system. By contrast, unfairness attacks, spoofing

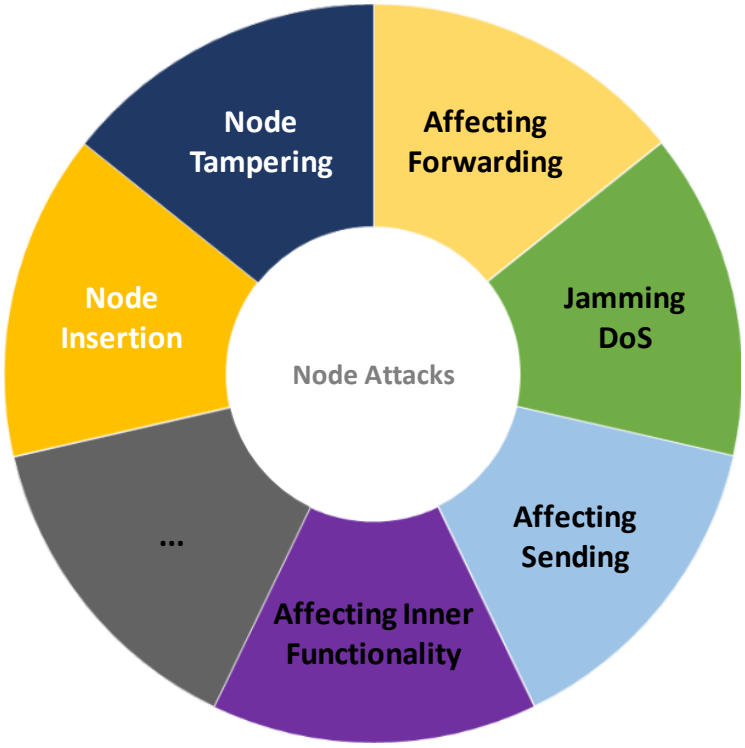


Figure 2.3: Different kinds of attacks against the nodes involved in the process of clock synchronization.

attacks, and message replay attacks can be classified as attacks against the medium access control (MAC) layer. These attacks are mainly focused on interrupting the access mechanism of the communication link so that the timeliness of the delivery of critical information cannot be guaranteed. Different attacks will lead to nonidentical consequences, and corresponding security schemes are typically adopted during the network operation phase to protect the delivery of IIoT data. However, clock synchronization, which is generally conducted during the network initialization phase, is still required to be secured to avoid potential damage or deterioration of the IIoT system.

Lacking an appropriate security scheme during clock synchronization will lead to unexpected degradation of the synchronization performance from two perspectives. On the one hand, the malicious attack against the IIoT devices involved in clock synchronization and the relevant synchronization timestamps will cause a direct effect on the synchronization performance in terms of achievable accuracy and efficiency. Any interference of the timestamps will lead to a potential increment of the synchronization frequency to guarantee the clock accuracy

Figure 2.4: Different types of attacks toward the communication link for transmitting timestamps for clock synchronization.

of the IIoT system. On the other hand, the influence on the clock synchronization performance will cause an indirect effect on the overall performance of the system. Degraded synchronization accuracy and enlarged synchronization frequency will potentially result in temporal misalignment of the critical data shared among the devices under the situation that fewer resources are available. Consequently, security issues should be carefully considered during the synchronization protocol design. Due to the uniqueness of clock synchronization, only some of the malicious behaviors in an IIoT system will affect the clock synchronization process. Other attacks, e.g., eavesdropping of timestamps, will not affect the synchronization performance directly since the delivery of correct timestamps during clock synchronization is mostly prioritized. Based on the target of malicious behaviors, these attacks can be classified into two categories, namely, node attacks and communication link attacks.

- **Node attacks** are abnormal behaviors that target at affecting the functionality of an IIoT device or the accuracy of its associated timestamps, while typical node attacks during clock synchronization are listed in Fig. 2.3. Node tampering, as one of the most com-

Table 2.1: The summary of the common-observed security threats during clock synchronization including different kinds of attacks and their classified impact on the synchronization performance.

Attack	**Impact** [42]		
	Fake timestamps	Reduced accuracy	Deny of service
Manipulation	★		
Spoofing	★		
Replay attack	★		
Rogue master attack	★		
Interception and removal		★	★
Delay attack	★		
L2/L3 DoS attack			★
Cryptography performance attack			★
Time protocol DoS attack			★
Master time source attack	★		

monly observed malicious attacks, can lead to severe consequences during clock synchronization. The information shared between the master clock and the slave clock can be untrustworthy, while any synchronization actions based on fake timestamps can make the overall clock error even larger. Moreover, the inner functionality of a node in the IIoT system can be regarded as its identification during clock synchronization. By changing the feature of any device, its clock generation and long-term behavior will be affected, leading to unpredictable performance during synchronization. Therefore, to deal with the node attacks, appropriate detection, recognition, and validation methods are necessary.

- **Communication link attacks**, by contrast, are attacks aimed at disturbing the transmission of timestamps among IIoT devices that are involved in the clock synchronization, and some of the attacks are summarized in Fig. 2.4. Those attacks can lead to potential failure and unexpected large latency of timestamp transmission, which will degrade the

accuracy and efficiency of the clock synchronization. Different from node attacks, communication link attack will not affect the behavior of the IIoT devices, so that the feature of each device will remain normal during attacks. As a consequence, the identification-based method could be less powerful in distinguishing abnormal timestamps. Since clock synchronization is exclusively conducted based on the timestamps shared in the network, ensuring the reliability of the timestamps at the destination would be one of the necessities for designing synchronization protocols.

A summary of the mentioned security threats are given in Table 2.1, where different kinds of external attacks are classified into three types in terms of their effect on the clock synchronization, namely, fake timestamps, accuracy reduction, and denial of service (DoS). Different security mechanisms should be designed against these issues correspondingly.

2.2.3 Clock Synchronization with Enhanced Security

Based on these observations, security issues should be well considered during the design of the clock synchronization protocol in IIoT systems. Since IIoT devices are typically vulnerable to external interference, lacking the security scheme during clock synchronization is identical to the behavior that exposing critical information and weakness to potential hostility. As a consequence, a mature synchronization technique should be associated with security-enhancing mechanisms or at least be able to cooperate with the existing methodologies. In the literature, different security-related schemes are proposed for clock synchronization from two main perspectives.

On the one hand, security can be enhanced by filtering untrustworthy timestamps. Authors in [43] proposed a resilient consensus-type algorithm where the unreliable data are filtered, and only trustworthy information is used for clock synchronization. In the case without reliable information, historical timestamps will be used to avoid loss under malicious attacks and uncertain network conditions. Moreover, a robust and secure time synchronization method is designed for sensor networks to tackle Sybil attacks [44]. The graph theory is utilized at the message level so that the performance of security-enhancement is further improved against Sybil attack and message manipulation attack. Furthermore, the blockchain-based method

is designed in [45] to improve security during clock synchronization for IoT systems. The timestamps are recorded and broadcasted by the blockchain so that the external attacks can be reduced. This kind of approach can achieve highly secure synchronization at the potential cost of high resource consumption.

On the other hand, different schemes for distinguishing unreliable devices and untrustworthy time references are proposed for different clock synchronization protocols. A complex synchronization protocol comprises level-based and diffusion-based methods is designed to ensure that only reliable time reference will be used during clock synchronization [46]. Consequently, the overall efficiency will be reduced as a drawback. By contrast, authors in [47] proposed a synchronization method on estimating the upper bound of the clock skew using a drift model so that the timestamps beyond this upper bound will be regarded as malicious. The synchronization performance is enhanced under malicious attacks, with the potential issue that the estimation accuracy cannot be guaranteed. Different from the other methods, a node-identification-based scheme is designed for industrial environments based on the fact that the drift of each clock is always unique [36]. The correlation between the clock skew and its operating environment is utilized to defend against Sybil attack and message manipulation attack.

2.2.4 Environment-Induced Clock Uncertainty

In an IIoT system consists of various applications ranging from advanced manufacturing to adaptive storage, the operating environments of the distributed devices can be very diversified. The operating temperatures can vary from very high (e.g., 60^oC in manufacturing spaces) to very low (below 0^oC in storage or outdoor winter environment) during network operation, leading to huge uncertainty of the clock behavior for the distributed devices. As every clock in the system is driven by the embedded crystal oscillator that is affected by the operating temperature, the diverse and harsh industrial environments will inevitably induce synchronization errors.

The operating environment has a gradual effect on the oscillation frequency of the device, which will lead to clock drift and synchronization inaccuracy. For a typical crystal oscillator, it is known that several issues will lead to the variation of oscillation frequency, including

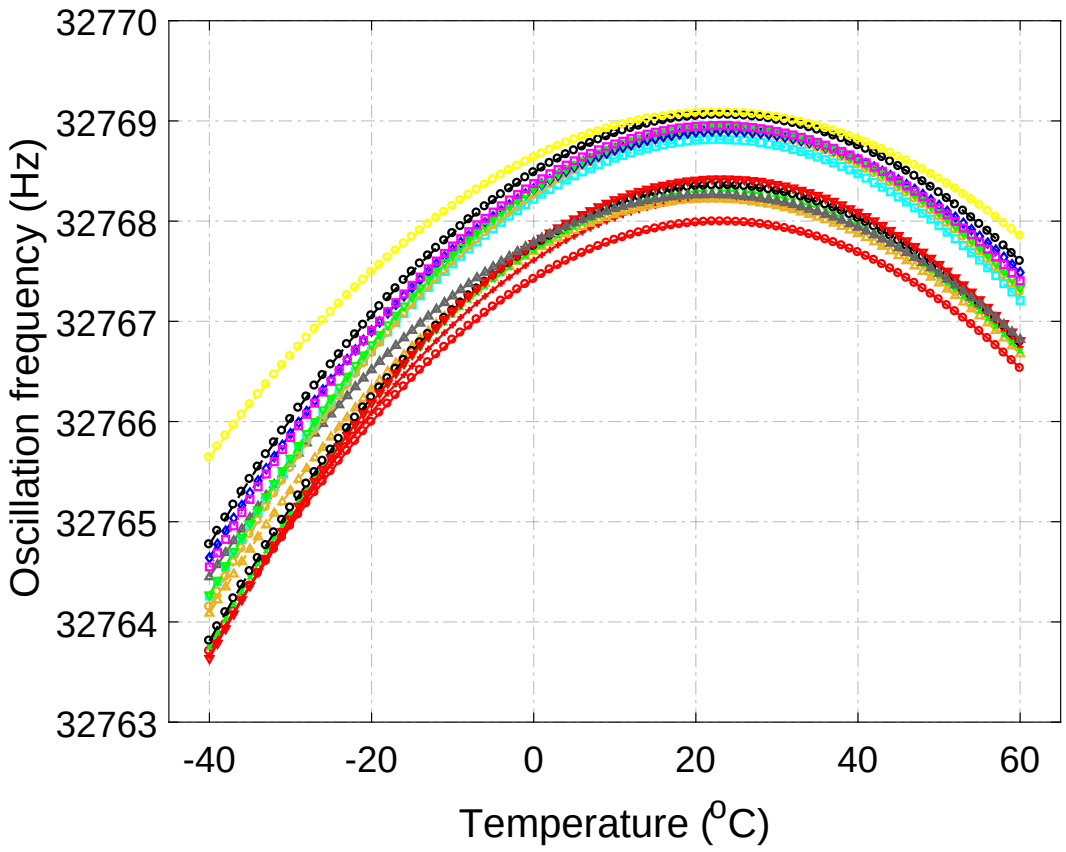


Figure 2.5: The experimental relationship between temperature and the oscillation frequency.

the operating temperature, external humidity, voltage, and so on. Temperature, among these environmental issues, has the most significant and unavoidable effect on clock inaccuracy. As evaluated in [48] based on experimental data, the relation between the clock oscillation frequency and its operating temperature follows a quadratic trend, which can be modeled by

$$f(T) = f_0(1 + \beta(T - T_0)^2) \tag{2.7}$$

where β is the temperature sensitivity factor and T_0 is the ideal operating temperature.

Based on (2.7), the oscillation frequencies of a series of clocks under temperatures in typical industrial environments, which are ranging from -40 to 60, are illustrated in Fig. 2.5. It can be observed that with a temperature variation of 30^oC, the oscillation frequency will drift by around $2Hz$. Although this drift is not huge compared to the nominal frequency, the clock drift will be high after long-term operation. Moreover, it is also observed in [49] that extreme

change of temperature will lead to a significant clock error ($60^{o}Cresultedin2\mu s$). Therefore, considering the effect of temperature during clock synchronization design can enhance the synchronization performance substantially.

To address the clock uncertainty induced by the external temperature, the most straightforward approach is to utilize temperature-compensated crystal oscillators (TCXO) to drive the local clocks. There are different kinds of TCXO manufactured, which are analyzed and compared in [50]. By adopting TCXO, the temperature issue can be substantially addressed. However, the cost of the device will be increased significantly, especially for dense sensors that are commonly deployed in IIoT systems. Authors in [51] proposed a temperature-compensated Kalman filter based synchronization method to model the variation of clock skew according to the features of each clock. The theory of Kalman filter is adopted to combine the estimation from neighbors to enhance the estimation accuracy. Moreover, different estimation methods are considered in the literature to investigate the relationship between the clock drift and its operating temperature. A constrained least square problem is solved in [52] to explore the behavior of each local clock, while the results showed that the clock drift can be saved with longer re-synchronization frequency. Based on the embedded temperature sensor in each device, the oscillators are automatically calibrated to enhance the synchronization performance, as shown in [53]. Meanwhile, a neural-network-based approach is considered [54] to derive a more comprehensive expression of each clock based on the experimental data, while the computation resource consumption may be higher as a drawback. Furthermore, a temperature-assisted clock synchronization approach is proposed in [55] with the ability to achieve self-calibration under different operating temperatures. Accurate estimation is assumed to be achieved based on the experimental data, while the effect of the clock skew can be removed as a benefit. However, the estimation error will inevitably induce synchronization inaccuracy that will be accumulated in the long-term operation.

Although there are many works proposed to analyze the relationship between clock behavior and the operating environment, almost none of the previously mentioned studies considered the effect of network uncertainties on the clock inaccuracy. As a consequence, a more comprehensive study is required to ensure a comprehensive understanding of the clock behaviors to be achievable.

2.3 Digital Twin in Industrial IoT Systems

2.3.1 Concept of Digital Twin

The situation of each constituent in an IIoT system is usually diverse and ever-changing in terms of their process condition, operating environment, work plan, and so on. The availability of analyzing and predicting the real-time situation of each component in the IIoT system is very critical in achieving complete intelligence for any critical applications, e.g., advanced manufacturing. Motivated by this consideration, digital twin is proposed exclusively for industrial systems in recent years. As one of the most promising techniques to enable industrial 4.0, digital twin is considered indispensable to bridge the gap between the ever-changing IIoT constituents and the associated real-time data analysis. Digital twin, by definition, is the digital counterpart of any physical process/product established through comprehensive observation, modeling, and simulation to completely demonstrate the real-time behavior of the physical entity. The established digital twin models in the remote locations, e.g., cloud center, can provide the real-time situation of the physical entities without necessary communication. Based on the situation estimated, accurate decisions can be made in advance so that potential loss or failure can be avoided.

Digital twin plays an important role in achieving seamless interaction and collaboration between the cyber domain and the physical world. By adopting the digital twin theory, a comprehensive understanding and thorough modeling will be conducted for each device of the IIoT system. Generally, historical and physical data will be collected from the local devices, and a virtual equivalent will be established accordingly. Therefore, the characteristics of each device, the situation of each process, and the potential abnormalities of each stage can be conveniently estimated and predicted by the digital twin established. The establishment and utilization of a digital twin system can be organized into three phases, namely, physical-virtual collaboration, virtual-virtual collaboration, and virtual-physical collaboration, representing three different kinds of novel interactions between the physical world and the digital domain. To be more specific, the physical-virtual collaboration comprises the establishment and update of the virtual model according to the physical data collected from the local devices. A wide variety of issues will affect the efficiency of this collaboration, including the data analysis techniques

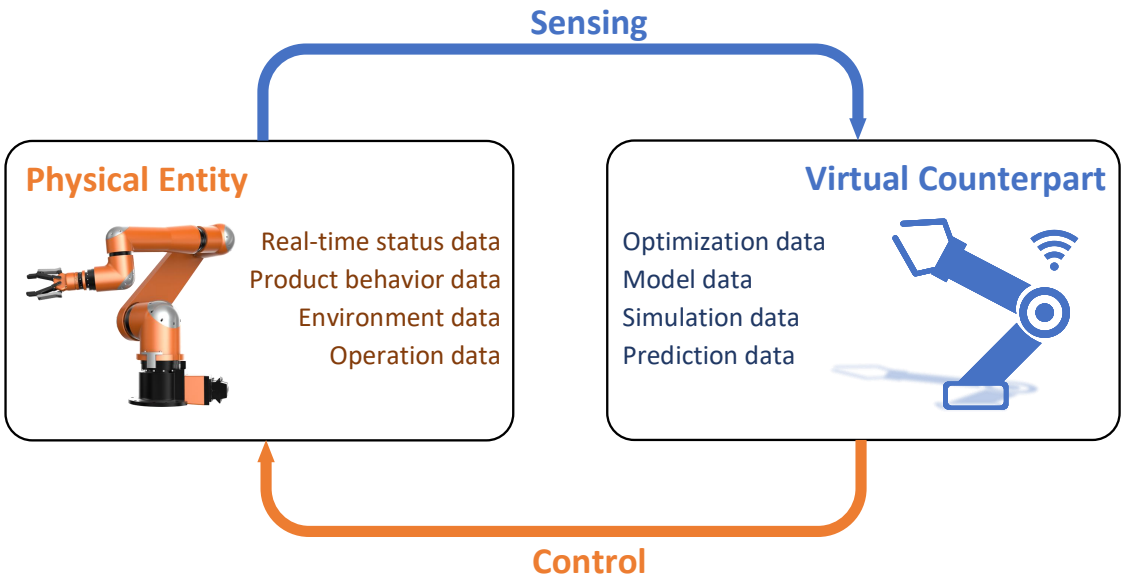


Figure 2.6: Digital twin of a robot arm and the associated data exchange.

adopted, simulation methods, data collection speed, etc. By contrast, virtual-physical collaboration mainly focuses on the feedback information from the digital twin to the physical entity. Real-time digital twin data will be generated so that timely decisions can be made to ensure the physical product/process will operate in a normal and optimized manner. The more accurate digital twins are established, the more timely and important decisions the physical infrastructures can receive. Finally, after establishing a digital twin for each device in a large system, the virtual models will interact with each other in a similar way that the physical devices are connected. As a consequence, a large digital twin system will be established with virtual-virtual collaborations so that the overall behavior of the physical IIoT system can be analyzed and modeled. Due to the virtual data and physical information associated with these three kinds of collaborations, a larger amount of integrated data will be inevitable, leading to the necessity of designing an efficient digital-twin-assisted system in enhancing the overall performance.

Taking intelligent manufacturing (IM) as a typical example, digital twin can improve the efficiency and intelligence of the overall process. The virtualization of the entire manufacturing process consists of the digitization of all the constituents of the system as well as their interactions. For an IM system, physical entities include the heterogeneous devices (sensors, actuators), plants, and processes, while those entities can be thoroughly modeled based on the massive relevant data collected. Generally, diversified sensors are responsible for collecting

various kinds of parameters from the physical entities to establish virtual models including the geometrical and physical information for each infrastructure. The robot arm, which is one of the essential components in the IM system, can be modeled through a comprehensive analysis of the data collected, as shown in Fig. 2.6. The most important data gathered from the physical entity can be categorized into real-time status data, product behavior data, environmental data, operation data, and so on. Each category of the data can reflect different characteristics of the physical robot arm, while they will be mapped to the digital twin domain for different purposes. The established virtual counterparts will be used to guide the manufacturing process, while real-time monitoring and feedback will be conducted to adjust the virtual models efficiently. Corresponding analysis according to the collected data will be performed from different perspectives, e.g., the optimization of the overall manufacturing process, the modeling of the robot arm, as well as the prediction and diagnosis of any potential abnormal behaviors. Moreover, there are two kinds of interactions in such a digital twin system, namely, sensing and control. These two interactions are responsible for establishing the real-time digital twin model and providing timely decision making, respectively. The accuracy and effectiveness of the overall digital twin-assisted IM system are enabled by the efficiency of these interactions, which should be carefully designed and maintained as one of the prerequisites.

2.3.2 Applications of Digital Twin in IIoT Systems

Besides advanced intelligent manufacturing systems, many novel designs had been proposed in the literature aimed at a wide variety of scenarios, which are inspired by the concept of digital twin [56] - [65] as illustrated in Table 2.2. Some representative works of these digital twin-based designs are listed, where the role that the digital twin plays and the expected functions enabled by the digital twin system in each scenario is briefly summarized. Generally, the concept of a digital twin can widely be adopted in all different kinds of industrial systems, including IIoT, smart manufacturing, cyber-physical production system, fault prediction, etc. To be more specific, a digital twin-enabled scheme is proposed to achieve automated data acquisition and processing in a cyber-physical production system [56], while an introduction to the digital twin shop-floor system is given in [57] to discuss the role of digital twin from the

Table 2.2: Typical applications of digital twin and the role it plays in each scenario.

Scenarios	Digital twin-enabled functions	Citation
Production System	Automated data acquisition and processing	[56]
Shop Floor Design	i). Build high fidelity models for shop floor system ii). Converge data from physical virtual systems	[57]
Fault Diagnosis	Fault diagnosis during design and maintenance	[58]
Smart Transportation	i). Establish traffic and vehicle models ii). Avoid traffic congestion and disorders	[59]
Smart City	i). Digital as various smart city services ii). Fully interconnection and integration	[60]
Virtual Reality	Architecture for co-simulation and communication	[61]
Multimedia	Seamless transmission of data during living	[62]
Remote Surgery	i). Analyze communication requirement in remote surgery ii). Ensure security against network outages and attacks	[63]
Elderly Care	Accurate and fast healthcare for elders	[64]
Agriculture	Efficiency and productivity during agricultural managament	[65]

perspective of establishing high fidelity virtual shop floor. To achieve fault diagnosis both in the development and maintenance phases, a deep-learning-based digital twin system is designed based on the comprehensive digital twin model established. More accurate and timely fault detection is accomplished consequently.

Meanwhile, applications in other non-industrial scenarios can also benefit from the development of digital twin. For example, the emerging intelligent transportation system can be enhanced by adopting digital twin-enabled traffic and vehicle data establishment [59], so that potential traffic congestion and disorders can be mostly avoided. Similarly, based on the concept of digital as a service (DaaS), the entire smart city as well as its associated infrastructures are virtualized while the interconnection among those constituents is completely enabled [60].

Moreover, multimedia-oriented applications are also designed with the assistance of digital twin. The convergence of digital twin with multimedia and virtual reality (VR) is proposed in [62] and [61], respectively to enhance the transmission of data as well as the integration between the virtual and physical entities. Furthermore, studies focused on E-healthcare [63] [64] and advanced agriculture system [65] are also proposed in the literature. Different improvements can be achieved based on the digital twin system designed, e.g., the capability to analyze the communication requirement and the possibility of network attacks during remote surgery, accurate and fast healthcare service for elderly people, as well as the management efficiency for agricultural applications. From these applications, it can be observed that the theory of digital twin can be widely spread to various domains with diversified improvements.

2.3.3 Potential Collaboration between Digital Twin and Clock Synchronization

Due to the high adaptability of digital twin system, different applications from various domains can be designed as novel schemes that are smarter, more efficient, and optimized. Typically, digital twin system will play the most critical role in such designs, while all these applications exclusively hinge on the efficient and accurate information exchange between the physical entities and the virtual domain. With inaccurate or delayed status information, the digital twin system established will be untrustworthy or even harmful. As previously introduced, the accuracy of data exchange and transmission is significantly affected by the temporal alignment of the involved devices, which requires accurate clock synchronization to ensure the informative of the data and avoid transmission failure. Meanwhile, the accurate clock synchronization in an IIoT system can also enhance the overall cohesion of the IIoT system, resulting in the improvement of the performance in terms of efficiency, productivity, and intelligence.

Conversely, the development of digital twin system can help to achieve more accurate and optimized clock synchronization. Since the digital twin system is responsible for establishing a highly comprehensive model for each device in the system, the intrinsic feature of each device, its embedded clock, as well as the operating environment will be some of the acquirable information for the data center. Consequently, the real-time behavior of each clock can be

predicted in advance so that excessive network resources used for clock synchronization can be saved for more critical applications. Moreover, the awareness of the operating environment and the surrounding circumstances can help the device achieve fully situation-awareness, i.e., synchronize with the most suitable reference with optimized frequency.

Based on the above observations, there is a mutual benefit between clock synchronization and digital twin system design. Achieving accurate clock synchronization is one of the prerequisites for the design of digital twin systems, while the utilization of digital twin system can make the clock synchronization protocols more intelligent. Therefore, it motivates the design of combining clock synchronization and digital twin system to establish a platform that can support advanced industrial applications with higher requirements on the collaboration and cooperation among the distributed infrastructures.

2.4 Chapter Summary

As one of the most important enablers in industrial IoT systems, clock synchronization can enhance the overall cohesion, seamless interaction, and timely collaborations among different heterogeneous devices and processes in the network. Due to the nature of industrial applications, most of which have a higher quality of service (QoS) and complicated operating environments, more stringent requirements are posed on the synchronization protocol design, including higher achievable synchronization accuracy, reduced network overhead for exchanges of packets containing timestamps, and enhanced security against external attack and internal abnormalities. Those requirements make the synchronization protocol design more challenging as compared to the traditional point-to-point synchronization methods. Moreover, the dynamic and harsh operating environments in IIoT systems will result in more challenges. The temperature-induced clock drift will be more severe in industrial environments, which will lead to a larger re-synchronization interval and waster of resources. Consequently, novel synchronization schemes are required for large-scale IIoT systems. Additionally, one of the most promising enablers in IIoT systems, digital twin can establish a comprehensive understanding of the physical entities by conducting thorough simulation and analysis. Existing digital twin-based solutions, ranging from intelligent manufacturing to data outlier detection, are ef-

fective in enhancing industrial vertical applications. It is observed that the mutual benefit can be achieved by adopting digital twin during clock synchronization so that more accurate clock synchronization is attainable by providing more understanding of the distributed clocks in a real-time manner. Therefore, designing synchronization schemes with this most advanced technique is expected to become one of the most inspiring topics for IIoT systems.

Chapter 3

Distributed Clock Synchronization Based on Intelligent Clustering

Accurate clock synchronization in the Industrial Internet of Things (IIoT) systems forms the cornerstone of distributed interaction and coordination among various infrastructures and machines in an industrial environment. However, due to the widespread use of wireless networks in industrial applications, constraints inherent to wireless networks including uncertain propagation delays, random packets losses, and unguaranteed communication resources are unavoidable, leading to dramatically increased clock synchronization error and unreliable or even outdated information. Meanwhile, time information transmissions are vulnerable to suffer from malicious attacks, causing unreliable timestamps and insecure synchronization. In this chapter, we proposed a distributed clock synchronization protocol based on an intelligent clustering algorithm to achieve accurate, secure and packet-efficient clock synchronization. The varying rate of skew of every clock is collected and utilized for cluster formation as well as malicious node detection. According to established clusters, various synchronization frequencies are assigned, which can avoid excessive network access contention, reduce overall communication resource consumption, and improve synchronization accuracy. Meanwhile, a two-tier fault detection algorithm consists of outlier detection and second-order regressive model prediction is applied to determine potential malicious nodes. The simulation results demonstrate that the proposed protocol overwhelms simultaneous synchronization protocols in terms of synchronization performance and faulty node detection.

3.1 Introduction

Technological advancement of Industrial Internet of Things (IIoT) and cyber-physical systems (CPS) is expected to enable the next industrial revolution through Industry 4.0, whose focus is distributed intelligent manufacturing [2] [66]. Distributed automation systems essentially rely on the reliable delivery and secure exchange of critical sensing and controlling information, which cannot be directly supported by traditional wired communication technologies due to the lack of scalability, convenience, and movability. Inspired by the easy installation and adoption of wireless communication, an increasing number of industrial factories are converting their approaches of information delivery and exchange into wireless. However, the association of the industrial environment and wireless communication should be carefully devised to realize the rigorous requirements of automation processes [67].

Industrial Internet of Things (IIoT) systems are widely investigated and utilized for their intelligence and seamless connectivity. The ubiquitous interconnection among sensors, controllers, and actuators constitutes a critical foundation for many important applications, ranging from large scale data gathering to distributed device cooperation and coordination. Those applications bring a stringent requirement on the time accuracy among devices in IIoT systems. With accurately synchronized clocks throughout the network, severe impact will be caused on the overall performance of the industrial system. Firstly, distributed control-related applications, e.g., collaboration among distributed controllers and actuators, cannot be proceeded in the case of lacking accurate time consensus. The control-related information shared among involved devices should be aligned precisely, which is the basis of the cooperative control in large scale IIoT systems with a large number of devices. This kind of application requires extremely high clock accuracy (around $1\mu s$ [68]), which can be achieved only with a dedicated synchronization process and communication resources. Secondly, data analysis in IIoT systems including large scale data aggregation and decision making, cannot be effectively performed if data collected from distributed devices are labeled with inconsistent and inaccurate timestamps. The misalignment of time-sensitive data among distributed devices will result in an inaccurate correlation analysis. In the scenario that a large number of distributed devices and massive data are involved in the same network, designing a synchronization protocol with

both high accuracy and less communication resource consumption will be of the utmost importance. Furthermore, in industrial systems with devices occupying time division multiple access (TDMA) mechanism to ensure the guaranteed delivery of their packets, accurate time synchronization among devices is extremely critical to avoid the unnecessary occupation of communication resources due to imprecisely divided time slots. Any unexpected conflict and contention will conduct indeterministic information conveyance among infrastructures as well as degraded profit in the overall industry manufacturing.

However, traditional synchronization methods are always achieved by calibrating the massive clocks within the network simultaneously. Due to the heterogeneity of the devices as well as the associated oscillator, their clocks will generate clock outputs with different quality. By assigned the unified synchronization frequency for all devices, insufficient synchronization accuracy or exceeding network resource consumption would be caused as some of the main consequences. Meanwhile, the design of clock synchronization in an IIoT system should never be isolated from communication. On the one hand, communication resources allocated to clock synchronization should be limited. In the literature, a few works were proposed to improve synchronization accuracy by designing complicated algorithms or redundant packets delivery [69], which unavoidably increases the amount of resource consumption, leading to a shortage of network resources and even reduced network lifespan. Therefore, a suitable protocol design is required to keep the overall synchronization process packet-efficient. On the other hand, the secure and reliable delivery of clock information is of the utmost importance in the network with potential packet losses, anomalous nodes, and malicious attacks [70]. Packet dropouts will naturally increase the interval between two synchronization instants, causing increased time imprecision in the network. Meanwhile, faulty or malicious nodes can take advantage of delayed or inaccurate information to mislead clock information calculation. With potential malicious attacks, safety and productivity of physical infrastructures would be diminished. All of those uncertainties caused by the characteristics of wireless communication will result in severe issues and failures without proper interference.

Main contributions of the proposed distributed clock synchronization protocol (DCSP) are summarized as follows:

1) The proposed DCSP can improve synchronization accuracy and packet-efficiency while

reducing the overhead during synchronization processes by adopting clustered network topology. DCSP contains two phases, namely, synchronization initialization phase (SIP) and distributed synchronization phase (DSP). SIP is responsible for correcting initial clock skew and offset while DSP is used for the consecutive procedure. Clocks will be accurately synchronized after SIP, however, time difference among clocks will still appear due to their various varying rates of skew (VRS), i.e., the changing rate of the oscillation frequency of a clock. Therefore, DSP is executed successively to achieve further clock synchronization.

2) The threshold-based K-means clustering algorithm is utilized to reduce the total communication resource consumption during the synchronization process compared to the randomly clustered topology. Specifically, based on the VRS of every node estimated, the overall network is divided into several clusters. A cluster of nodes with larger VRS will be synchronized more frequently while larger synchronization intervals will be appeared at nodes with insignificant VRS to avoid unnecessary packet delivery. Therefore, nodes in the network will not be synchronized simultaneously, as always proposed in the literature. Simulation results proved that the overall communication frequency of DCSP is reduced on the promise that the synchronization accuracy is guaranteed.

3) Meanwhile, VRS information-based fault detection algorithm is developed to improve synchronization security in networks with potential node failure and malicious attacks. We proposed a two-tier fault detection algorithm, where cluster heads are firstly used to decide if any node in their clusters is malicious. Historical VRS values of potential suspicious nodes and their prospective VRS estimated according to a second-order autoregressive model will be further compared in the case that more than one malicious node are detected within one cluster. It could be observed that by adopting the proposed detection approach, the effect of malicious attacks and faulty nodes is significantly reduced.

The rest of this chapter is organized as follows. Related work about synchronization protocols is given in Section 3.2. Protocol initialization consists of network topology initialization, clock model, and the synchronization initialization phase is discussed in Section 3.3, while the distributed synchronization phase is designed in Section 3.4, including distributed synchronization frequency setup and faulty node detection algorithm. In Section 3.5, simulation results are carried out to demonstrate the effectiveness of the proposed protocol in terms of packets

delivery reduction and malicious nodes detection. Finally, we conclude our work in Section 3.6.

3.2 Related Work

Numerous clock synchronization protocols have been proposed to achieve accurate or reliable time synchronization in different scenarios, including wireless sensor networks, IoT systems, and wireless local area networks [71]. Cluster-based topology is one of the most commonly selected structures during network setup to ensure the efficiency of clock synchronization, especially when there are a large number of nodes involoved in one network. Meanwhile, recursive packet exchange between two nodes during clock synchronization is prevalent due to the improved security against various malicious attacks as well as the ability to eliminate symmetric propagation delay. Therefore, works about synchronization protocols using cluster-based recursive packet exchange will be discussed in this section. Furthermore, since the security of clock information is regarded as one of the most fundamental requirements in industrial applications, secure synchronization algorithms in literature will also be summarized.

A) Reference-based Recursive approaches refer to schemes where the bidirectional transmission of clock information between each two participated nodes is necessary. This kind of packets delivery is more commonly adopted in protocol designs [49]-[53], as well as our proposed protocol. Precision time protocol (PTP) [72] was proposed to achieve precise clock synchronization in a wired computer network without considering clock drift correction. Numerous improvements were performed to migrate the traditional PTP to be suitable to time-sensitive wireless networks, from various prospectives. Authors in [33] considered a revised version of PTP with clock skew correction by calculating the relative oscillation factor between every two nodes in an industrial network, however, the network structure and synchronization process flow are not thoroughly developed. Meanwhile, the propagation delay is assumed to be symmetric, which may cause uncertain synchronization inaccuracy. The effect of asymmetric channel is studied and estimated in [73], where the packet delay variations are modeled by a Gamma distribution. On the other hand, two-way time synchronization protocols based on overhearing mechanisms are proposed in [39] and [74], with an expectation that the packet

delivered can be reduced.

B) Clustered Synchronization approaches are commonly adopted in networks with homogeneous nodes, e.g., wireless sensor networks. By dividing distributed nodes into clusters, the time information can be delivered more efficiently while cluster heads can provide greater processing capability to deal with the computation inside each cluster. Based on this idea, the authors in [75] proposed a clustered consensus time synchronization (CCTS) by dividing all nodes in the network into several clusters while the intra-cluster time synchronization is achieved by using an improved distributed consensus time synchronization algorithm. Due to the reason that the global time synchronization is achieved within two successive steps, a lower convergence rate is unavoidable. A cluster-based maximum consensus time synchronization (CMTS) method is proposed in [76] to improve the rate of convergence by applying the maximum consensus and consistency theory while considering bounded delay models. Furthermore, a three-step consensus time synchronization method to reduce the communication overhead was proposed in [77] by setting a threshold during the intra-cluster synchronization stage. The packets required to be transmitted and the convergence rate are further improved in clustered WSNs.

C) Secure Time Synchronization methods are widely explored in literature due to the necessity of security mechanisms to mitigate various kinds of malicious behavior including message manipulation attack and Sybil attack during clock synchronization. A spanning tree was initially developed in [16] to eliminate fake timestamps transmitted from malicious nodes by designing a secure synchronization protocol for each node based on the information from its father node and grandfather node. According to the offset calculated from its parent nodes, the reliability of timestamps can be validated. Meanwhile, a robust and secure time synchronization protocol (RTSP) [44] was proposed to defend Sybil attacks by performing malicious behavior detection according to the improved graph theoretical method. Moreover, node-identification-based time synchronization was proposed in [78] to enhance the security of time synchronization on the basis of the fact that the clock drifting of every hardware is unique [79]. A correlated skew checking scheme was proposed to discover potential anomaly during time synchronization.

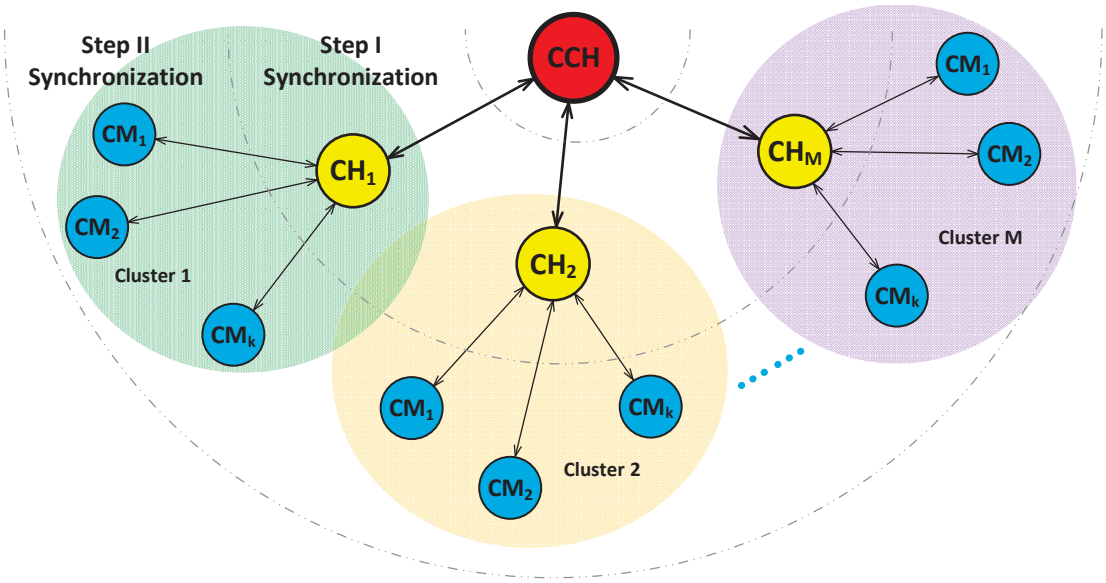


Figure 3.1: The cluster-based network with a few nodes randomly deployed. Chief cluster head (CCH) is selected to provide reference clock information while several cluster heads (CH) are elected for reference expansion.

3.3 Synchronization Protocol Initialization

3.3.1 Network Topology Initialization

Network topology selection is highly dependent on the total number of nodes in the network, distribution of the nodes, as well as the requirement of different scenarios. For clock synchronization, accuracy and efficiency are of the utmost importance, since a severe degradation will be caused if an inappropriate network structure is adopted. Compared to other network structures, e.g., spanning tree or mesh topology, cluster-based architecture has its unique advantages, including the increased distributed computation ability and enhanced efficiency during information exchange. Therefore, cluster-based topology is used in the proposed protocol for a local area IIoT system, as shown in Fig. 3.1. Meanwhile, all nodes in the network are assumed to transmit information within one hop while every node is inside the communication range of other nodes, which are feasible assumptions in a local area network. In this chapter, different

clustering policies are adopted in SIP and DSP, respectively. For SIP, one of the nodes with a stable clock output will be selected as the standard time, and all the other nodes should be synchronized accordingly. The rest $n-1$ nodes in the entire network are gathered into M clusters randomly, M cluster heads (CHs) are elected consequently. These CHs are responsible for the synchronization within their clusters, while the reference node can be regarded as a chief cluster head (CCH), which is used to deliver the standard time to all the CHs. It is worth noting that the clock of CCH is always regarded as reliable and trustworthy, which requires a careful selection at the initialization phase. The clustering algorithm may frequently perform to select more reliable and proper CHs. However, the CCH will remain the same in our assumption. Once the synchronization among CCH and CHs is achieved, the cluster members will synchronize their clocks according to their CHs immediately, so that the overall synchronization can be realized.

3.3.2 Clock Model

The reason for the time difference among distributed nodes in a large-scale network is twofold. Firstly, since devices are equipped with low-cost quartz-crystal oscillators with time-varying frequency, the initial state of clocks would be different. Meanwhile, the varying rate of their skews is also not the same, causing consecutive error after long-term operation. Specifically, the clock of one node can be expressed by

$$C_i(t) = \alpha_i(t)t + \beta_i \tag{3.1}$$

where α_i and β_i are time-varying clock skew and constant clock offset, respectively. An ideal value of α_i and β_i are 1 and 0, which means $C_{ideal} = t$. Clock skew is caused by frequency drift of the oscillator in the node while the clock offset is typical because of the initial setup of the network. Due to the internal or external environment, e.g., voltage, temperature and aging, the oscillation frequency of every device is slightly different from the ideal scenario, leading to significant time difference among infrastructures in the same network. According to [71], the frequency of an oscillator of node i at the instant t_k can be written as

$$f_i(t_k) = f_i(1 + s_i t_k) + \Delta f_i + N_i(t_k) \tag{3.2}$$

where f_i is the ideal frequency of the clock, s_i is its drift rate, Δf_i is the frequency skew of the oscillator, and N_i is the noise of node i, which has an insignificant effect on the frequency variation. Since clock skew is the relative relation between the real clock frequency of each clock and its ideal clock frequency, based on (3.2), clock skew of node i can be expressed as

$$\alpha_i(t_k) = 1 + \Delta\alpha_i + s_i t_k \tag{3.3}$$

where $\Delta\alpha_i$ and s_i are the initial clock skew and VRS, both of which affect the clock accuracy. In this proposed clock model, it is assumed that $\Delta\alpha_i$ is a constant value affected by the manufacturer while the VRS, namely, the changing rate of clock skew, is also assumed to be a constant dependent on a few internal or external factors as mentioned previously.

The synchronization accuracy in this chapter is described by the normalized mean square error (NMSE) of the time difference in the network at any instant t_k, which is written as

$$\epsilon(t_k) = \frac{\sum_{i=1}^{n} (\frac{C_i(t_k) - C_0(t_k)}{t_k})^2}{n} \tag{3.4}$$

where C_0 it the local clock of CCH node. It is worth noting that this clock model is also adopted in the DSP.

3.3.3 Initial Synchronization

The SIP is achieved based on the packet delivery approach used in the PTP scheme. As stated previously, the SIP is divided into two steps. In Step I, the clock synchronization between every CH node and the CCH node can be achieved with four successive packet transmissions. The packet delivery order is shown in Fig. 3.2, where the first two packets *skew_correction_1* and *skew_correction_2* are responsible for calculating the relative skew between any two nodes. Although similar to the PTP scheme, there are still several significant differences in the proposed protocol. Firstly, skew correction based on the first two transmissions with the same direction is adopted, which is not considered in the traditional PTP scheme. Meanwhile, the second pair

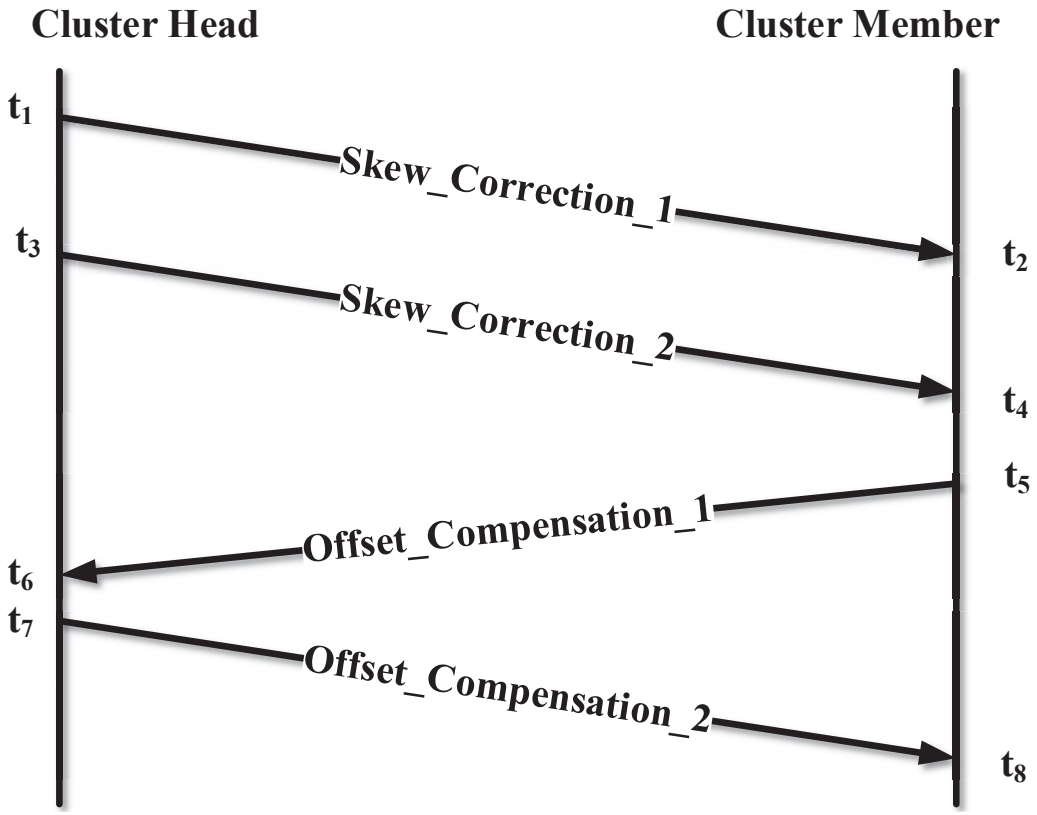


Figure 3.2: Two pairs of packets are required to be exchanged during each synchronization process for skew correction and offset compensation, respectively.

of transmissions in the reversed direction can eliminate the influence of symmetric propagation delay as stated in the PTP scheme, with higher accuracy due to the elimination of clock skew. To be more specific, according to (3.1), clocks of node i and j can be written as

$$C_i = \alpha_i t + \beta_i$$

and

$$C_j = \alpha_j t + \beta_j \tag{3.5}$$

Thus, the relative skew α_{ij} between node i and node j can be obtained by

$$\alpha_{ij} = \frac{\alpha_i}{\alpha_j} = \frac{C_i^{t_3} - C_i^{t_1}}{C_j^{t_4} - C_j^{t_2}} \tag{3.6}$$

It is worth noting that when the CCH node is selected as the node j, its relative skew compared to other nodes will be denoted by r_i, given by

$$r_i = \frac{\alpha_i}{\alpha_{CCH}} = \frac{C_i^{t_3} - C_i^{t_1}}{C_{CCH}^{t_4} - C_{CCH}^{t_2}} \tag{3.7}$$

By multiplying α_{ij} to (3.5), the clock of node j will be increased with the same rate of node i, shown as

$$\hat{C}_j = \alpha_{ij}(\alpha_j t + \beta_j) = \alpha_i t + \hat{\beta}_j \tag{3.8}$$

where $\hat{\beta}_j = \alpha_{ij}\beta_j$ is the new offset of node j after skew correction, while another two packets *offset_compensation_1* and *offset_compensation_2* are transmitted in the opposite direction to compensate this offset with respect to its CH node. The relative offset of any node i and j can be calculated by

$$\beta_{ij} = \frac{1}{2}[(C_i^{t_5} - C_j^{t_6}) + (C_i^{t_8} - C_j^{t_7})] \tag{3.9}$$

By adding (3.9) to (3.8), the newly generated clock of node j will finally run with the same

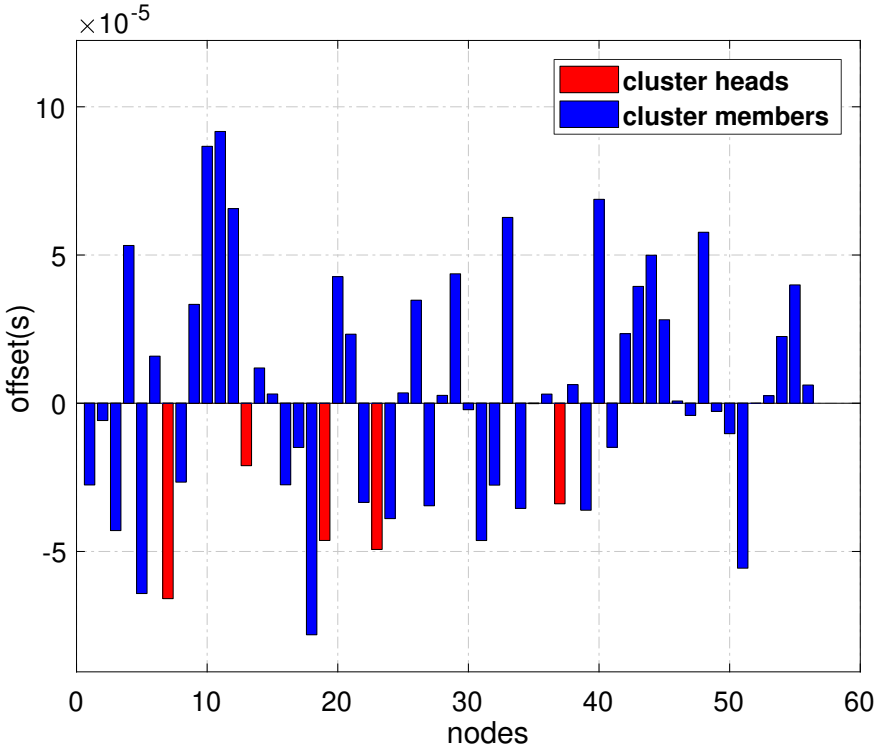


Figure 3.3: The initial offset in the network for distributed clocks, where the offsets are huge and random compared to the CCH node.

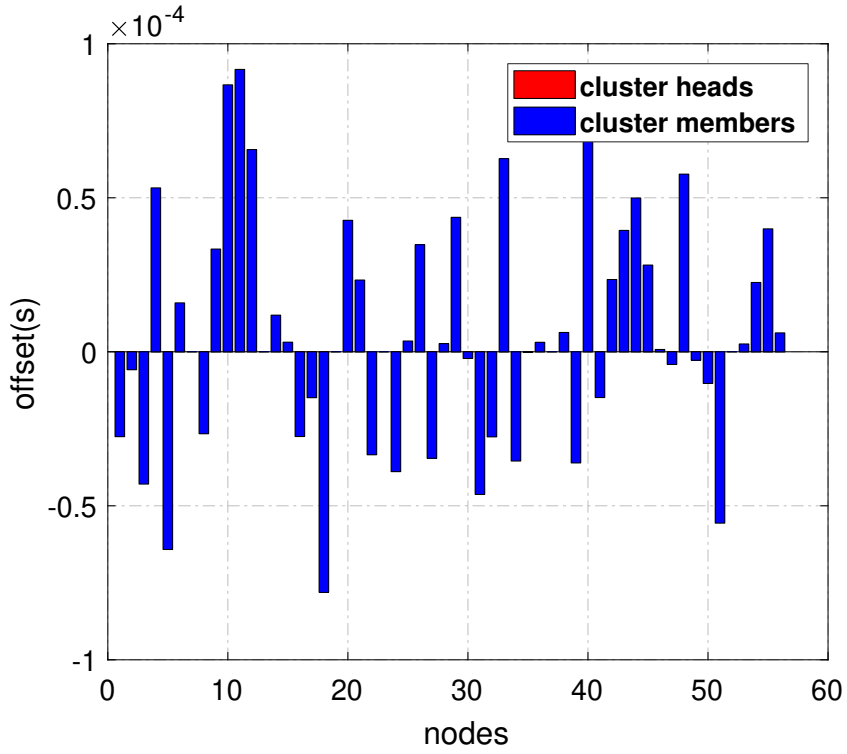


Figure 3.4: The offset evolution after step I synchronization. The offset for cluster heads are eliminated while the offsets in cluster member almost remain the same.

skew and offset as node i. By performing this process to all CHs, Step I synchronization is achieved. In Step II, every node will synchronize according to its CH, with the same procedure of Step I. Global synchronization will be realized once another four packets are successfully delivered between every node and its CH. Theoretically, all nodes in the network will be operated under the same clock parameters after Step I and II synchronization.

A simulation example is performed to show the results and drawbacks of this commonly used synchronization process only considering symmetric propagation delay, which can be thoroughly canceled with round packet delivery. The offsets during synchronization initialization phase is shown in Fig. 3.3, which demonstrates the initial setup of 50 nodes in the network with an offset around 2×10^{-6} seconds. After Step I synchronization, i.e., the synchronization phase between CCH node and every CH node, the offsets among those nodes are almost eliminated compared with cluster members. Fig. 3.5 shows the final synchronization results after Step II, where all the nodes are synchronized with respect to the CCH node with a tiny time error due to the ignorance of packet losses and contention delay. However, this time difference

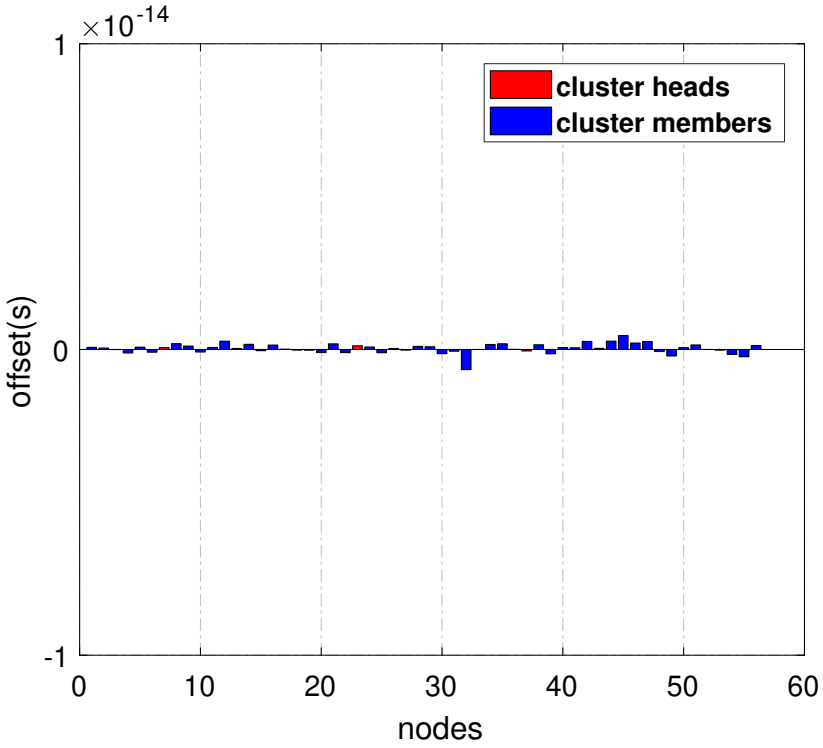


Figure 3.5: The value of offsets for all clocks are extremely small after Step II clock synchronization. Traditional synchronization will stop here and repeat periodically to meet the accuracy criteria.

becomes relatively huge after 10 seconds since the clocks are running freely with a different VRS s_i in every node, as shown in Fig. 3.6. Therefore, the periodic synchronization process is required to maintain an acceptable time difference in the network.

3.3.4 Simultaneous Synchronization Analysis

It is clear that in the synchronization initialization phase, all nodes in the network are randomly divided into a few clusters. Compared to the distributed synchronization phase to be introduced in the next section, SIP synchronizes all nodes simultaneous in the same frequency, called as cluster-based simultaneous synchronization. As shown in Fig. 3.5, the clock difference within the network is reduced dramatically after this synchronization process. Although the time error increased after a few seconds, this error can be limited within the expectation if the SIP scheme can proceed frequently. A proper synchronization frequency should be calculated according to the evolution of the clock error to ensure the overall clock accuracy in the network under the

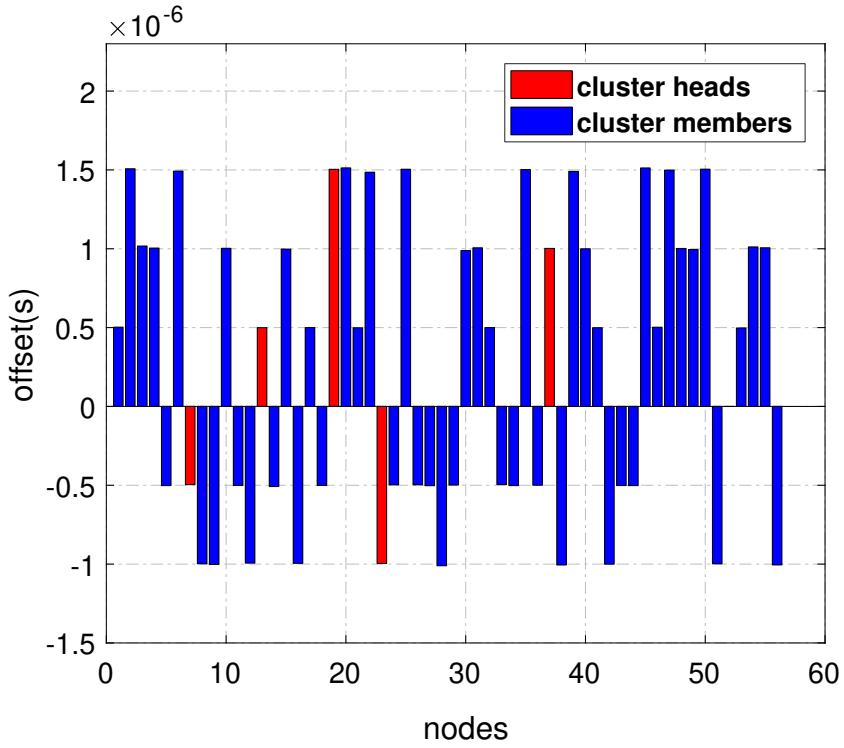


Figure 3.6: Due to the existence of varying rate of skew, the offsets of the synchronized nodes will rise dramatically after a few seconds.

requirement. According to (3.4), the largest NMSE in the network can be obtained by

$$\epsilon_{max} = \frac{\sum_{i=1}^{n} \left(\frac{C_i(\tau^-) - C_0(\tau^-)}{\tau^-}\right)^2}{n} \tag{3.10}$$

where τ^- is the instant right before the synchronization is performed while n is the total number of nodes in the network. Clearly, by setting a proper synchronization frequency, the overall NMSE can be limited within the requirement. However, by synchronizing all nodes simultaneously, redundant resources will be consumed, as illustrated in the simulation section. Therefore, the distributed synchronization scheme is proposed to reduce unnecessary synchronization actions.

3.4 Distributed Synchronization Phase

As stated in Section 3.3, periodic synchronization is necessary, therefore, the frequency of synchronization should be considered comprehensively. This frequency can be related to the VRS of every node, the accuracy requirement according to the potential applications, and the communication constraints including random delay and packet losses. In traditional cluster-based synchronization processes, all nodes in the network will be synchronized with the same frequency, i.e., every node will try to communicate with its CH node simultaneously, causing potential network access contention and increasing communication channel consumption, especially in a large-scale network. Motivated by Fig. 3.6, where the VRS of some nodes might be similar to each other, we proposed an intelligent clustering strategy to allocate heterogeneous frequency to each cluster.

3.4.1 Intelligent Clustering

Instead of randomly forming the clusters, we are inspired by the threshold-based K-means clustering algorithm [80], which is an adaptive K-means algorithm focus on regulating the difference among every node in each cluster to be smaller than a pre-defined threshold. In this chapter, we considered to use the VRS value of each device as the input of the adaptive K-means clustering algorithm to intelligently clustering the heterogeneous devices into desired groups based on their clock quality. The specific steps for the intelligent clustering algorithm 1 are listed as follows:

1. The relative skew r_i between every node and the CCH node selected in the previous phase is calculated according to (3.6). Then, one of the nodes is selected as the CH node of the first cluster.

2. For every node in the network, calculate the difference between its relative skew r_i and the relative skew r_{CH_1} of the CH node. If this difference d_i is larger than the predefined threshold, a new cluster with the centroid of node i should be established. Otherwise, node i will be collected in the corresponding cluster. The difference between every two clusters should be kept larger than the threshold, while the internal difference among

nodes inside each cluster should be limited within the threshold.

3. After establishing the preliminary clusters based on step 2), the node closest to the center of every cluster should be selected as the new CH. If any node in the network has a difference smaller than the threshold compared to more than one new CHs, that node should be allocated to the cluster with a smaller difference.

Based on this intelligent clustering algorithm, the cluster structure of the network is dynamic and adaptive. This is very important since the VRS of every node could be varying due to environmental change. It can be observed that the proposed K-means clustering algorithm is based on the predefined threshold, which determines the clustering size and the inter-cluster interval. A more individualized synchronization strategy can be design throughout the overall IIoT system with a smaller threshold. However, the network overhead and synchronization efficiency may be degraded as a drawback. Therefore, a tradeoff is necessary to balance the clustering design according to the network scale and synchronization requirement. Furthermore, it is worth noting that the re-clustering is always necessary to be conducted by the CCH in some specific situations, e.g., malicious CHs are detected.

3.4.2 Heterogeneous Frequency Setup

After organizing all nodes into several clusters according to their VRS, the more advanced synchronization phase, i.e., distributed synchronization phase, should be considered. The synchronization design in terms of the frequency for each IIoT device should be assigned with tailored feature according to the varying rate of its clock, in other words, a node with a highly unstable clock should be synchronized more frequently than the node with a high-quality clock. After synchronization initialization phase, all nodes in the network are accurately synchronized with respect to the time reference provided by CCH node, which is written as

$$C_i(t) = \alpha_0(t)t + \beta_0 \tag{3.11}$$

Algorithm 1 Intelligent clustering using threshold based K-means algorithm

Initialization:
for $i = 1 : \#node$ **do**
  Calculate r_i according to Eq. (3.7)
  Select $node_1$ as CH_1 of cluster C_1
end for
$\#cluster = 1$
Recursion:
for $i = 1 : \#node$ **do**
  for $j = 1 : \#cluster$ **do**
    if $r_i - r_{CH_j} < threshold\eta$ **then**
      $i \in C_j$
    else
      $CH_{j+1} = i$
      $\#cluster = \#cluster + 1$
    end if
  end for
end for
Finalization:
for $j = 1 : \#cluster$ **do**
  Calculate the mean $\bar{r}_{C_j}$
  for $k = 1 : \#node \in C_j$ **do**
    $d_k = \bar{r}_{C_j} - r_k$
    if $d_k < d_{CH_j}$ **then**
      $CH_j = k$
    end if
  end for
end for

However, due to the uniqueness of the VRS value of each node, the clock output will vary in different extents

$$C_i(t) = (\alpha_0 + r_i s_i t)t + \beta_0 \tag{3.12}$$

Based on (3.4), the averaged time difference inside any cluster compared to the time reference can be estimated by

$$\epsilon(t_k) = \frac{\sum_{i=1}^{n_m} r_i^2(t_{k-1})\bar{s}_i^2 t_k^2}{n_m} \tag{3.13}$$

where n_m is the number of nodes inside the m^{th} cluster, while $r_i(t_{k-1})$ is the relative skew between node i and the CCH node calculated to perform the skew correction in the last synchronization instant. The synchronization accuracy will reduce gradually, and a higher time error will arise with time. To limit this time difference in the desired range, the synchronization frequency should be assigned properly so that the overall synchronization accuracy can be smaller than the requirement ρ, i.e., $\epsilon(t_k) \leq \rho$. According to (3.13), the next synchronization instant can be further derived as

$$t_k = \sqrt{\frac{n_m \rho}{\sum_{i=1}^{n_m} r_i^2(t_{k-1})\bar{s}_i^2}} \tag{3.14}$$

Clearly, the synchronization frequency for each cluster depends on its network scale, synchronization requirement on the accuracy, VRS expectation of the cluster, and the initial difference of skews.

3.4.3 Malicious Node Detection

Due to the feature of wireless communication, faulty nodes and malicious attack are inevitable during clock synchronization. Malicious nodes are assumed to be omniscient and they can take advantage of delay [81] so that during the synchronization period, misleading information can be received by the CH node, leading to unexpected time error. Therefore, an effective approach with limited computation complexity to detect malicious nodes is necessary. Based on the VRS calculated from the previous sections, a two-tier cluster-based detection algorithm is proposed in algorithm 2. In the first step, the CH of each cluster will calculate the difference of VRS between itself and its cluster members. In the case that only one cluster member is beyond the threshold, that member will be recorded as a malicious node. If more than one

Algorithm 2 Malicious node detection based on CH observation and model prediction

Require: a small number ϕ

for $j = 1 : \#cluster$ **do**
  $M_j = 0$
  for k=1:#node in C_j **do**
    $d_k = r_k - r_{CH_j}$
    if $d_k > thre$ **then**
      $M_j + +$
      $n_j = \{k\}$
    end if
  end for
  if $M_j = 1$ **then**
    Report n_j as malicious node
  else
    Calculate expected $\hat{\alpha}_k$ according to Eq. (3.15)
    if $r_{n_j} - \hat{r}_{n_j} > \phi$ **then**
      Report n_j as malicious nodes
    else
      Report $\overline{n}_j$ as malicious nodes
    end if
  end if
end for

node are recorded inside any cluster, the probability that the CH node is faulty will arise significantly. To avoid the effect caused by malicious CH nodes, the potential faulty nodes should be further validated in step 2, where every node inside the interested cluster should calculate its expected VRS according to the historical information, written as

$$\hat{A}(k+1) = \Omega A_H(k) \tag{3.15}$$

where $\hat{A}$ is the prediction while A_H is the historical data given by $A_H(k) = [\alpha(k), \alpha(k-1)]^T$, while Ω is the coefficients of this second-order auto-regressive model. The expansion of Ω is given by

$$\Omega = \begin{bmatrix} \omega_{11} & \omega_{12} \\ \omega_{21} & \omega_{22} \\ \cdots & \cdots \\ \omega_{n1} & \omega_{n2} \end{bmatrix} \tag{3.16}$$

Then nodes with an observed value $\alpha_i(k+1)$ significantly different from its predicted value $\hat{\alpha}_i(k+1)$ will be reported as malicious. By setting this two-tier fault detection algorithm, unnecessary computation is saved in the case with an individual malicious node. Meanwhile, the potential dangers caused by faulty CH nodes can be avoided.

After malicious node detection, there will be two different approaches to deal with abnormal nodes according to their roles. If a cluster member is detected as malicious, its time information generated will be ignored to avoid excessive clock inaccuracy within the network. Instead, the clock information provided by its cluster head will be utilized for the clock synchronization. By contrast, if a cluster head is regarded as malicious, the clustering algorithm should be executed once again to ensure that another reliable cluster head is selected. Although this process will bring more communication overhead, the security of the synchronization process can be significantly improved.

3.5 Performance Evaluation

3.5.1 Simulation Setup

In this simulation, 50 heterogeneous IIoT devices with different clock qualities are randomly deployed in an industrial environment, while the VRS value for the distributed clocks are assigned from 1×10^{-8}, 3×10^{-8}, 5×10^{-8}, 7×10^{-8}, and 9×10^{-8}. After SIP, all nodes will operate with an identical clock skew, however different VRS will cause clock error gradually. The CCH node used in SIP will continually act as the reference node, while new CH nodes will be appointed according to the clustering algorithm. Each cluster member is assumed to be connected with its cluster head within one hop.

3.5.2 Intelligent Clustering

In this section, the effectiveness of the threshold-based clustering algorithm is evaluated. The threshold is set to be 1×10^{-8} so that smaller VRS differences can be captured. The clustering result for the interested network is shown in Fig. 3.7, where all the nodes are classified into 5 clusters with distinct VRS. It is clear that the difference of VRS inside every cluster is limited to smaller than the threshold, while the distance between the centroid of each two clusters is kept far enough. Therefore, various synchronization frequencies can be assigned to different clusters without compromising any synchronization accuracy.

3.5.3 Distributed Synchronization Analysis

According to section 3.4.2, synchronization frequencies are calculated by (3.14). Compared to the simultaneous synchronization protocol, the proposed protocol will assign a higher frequency to clusters with larger VRS, while clusters with a smaller VRS will be synchronized less frequently. Therefore, the time difference of every cluster compared to the reference time will not be the same. By adopting the proposed scheme, the overall clock synchronization accuracy should be guaranteed on the basis that less resource is consumed.

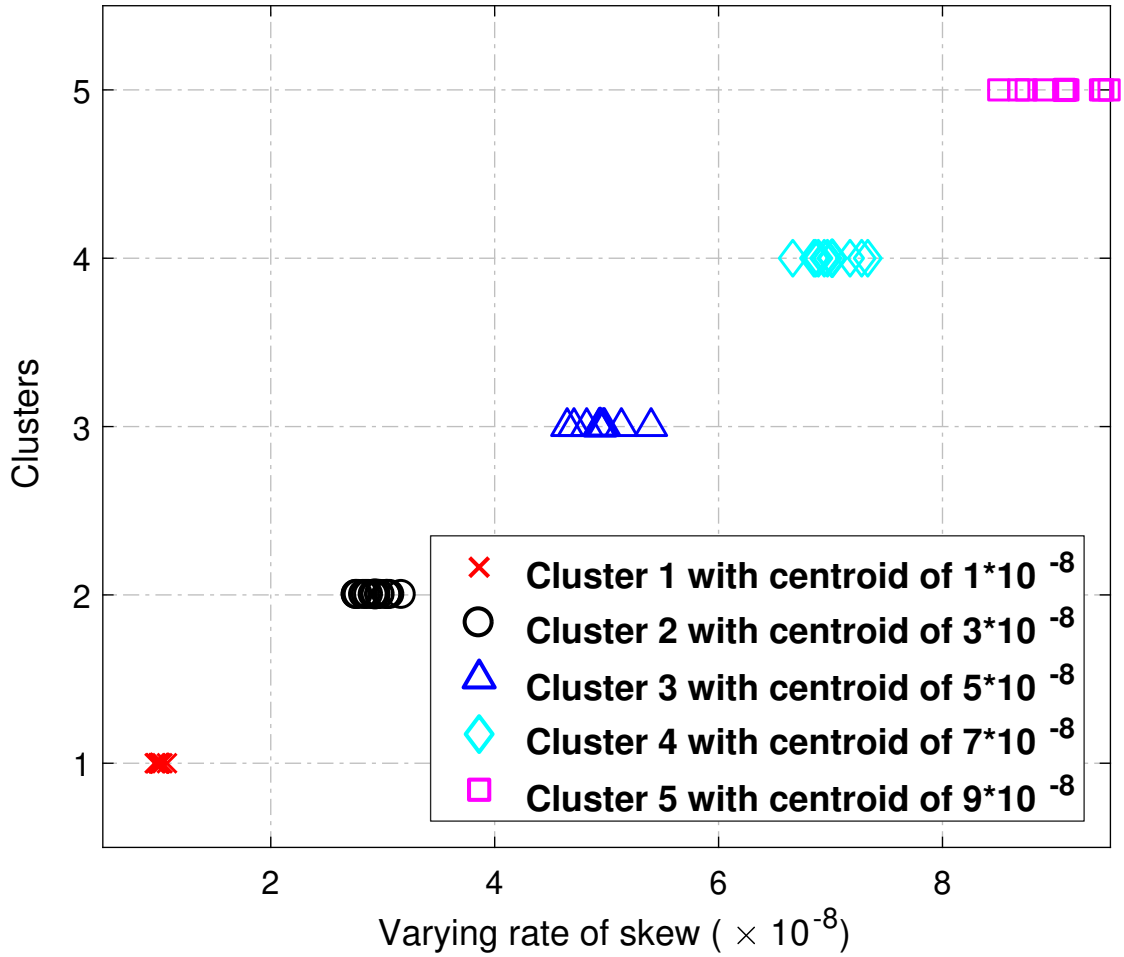


Figure 3.7: A number of 50 nodes are organized into 5 clusters by the proposed intelligent clustering algorithm. Five clusters are formed, where expected cluster centers are 1×10^{-8}, 3×10^{-8}, 5×10^{-8}, 7×10^{-8}, and 9×10^{-8}, respectively.

3.5.3.1 Accuracy Analysis

In this subsection, the overall synchronization accuracy is evaluated. As stated previously, the clock accuracy is analyzed based on (3.4). The main difference between the simultaneous scheme and proposed scheme is that, for simultaneous synchronization, the largest NMSE in the network is calculated by (3.10) based on the averaged clock difference of all nodes in the network with respect to the reference node. By contrast, the NMSE for the proposed scheme will be calculated on the basis of every cluster. For the j^{th} cluster, its NMSE at instant τ^- is given by

$$\epsilon_{j_{max}} = \frac{\sum_{i=1}^{n_j} (\frac{C_i(\tau_j^-) - C_0(\tau_j^-)}{\tau_j^-})^2}{n_j} \tag{3.17}$$

where n_j is the number of nodes in cluster j. The overall NMSE in the network can be written as

$$\hat{\epsilon}_{max} = \frac{\sum_{j=1}^{M} (\epsilon_{j_{max}})}{M} \tag{3.18}$$

where M is the number of clusters established according to the K-means clustering algorithm. By limiting the maximum NMSE of every cluster to smaller than the required accuracy, it can be guaranteed that the overall NMSE in the network will meet this requirement. Therefore, it can be observed that the simultaneous synchronization protocols always give a suboptimal frequency to guarantee the accuracy, however, unnecessary packets are delivered.

Although both (3.10) and (3.18) are responsible for calculating the NMSE in the overall network, the freedom of NMSE restriction is different in those two schemes. In (3.10), the NMSE will increase simultaneously, while it will drop only if the synchronization is performed. Therefore, the evolution of clock accuracy in the simultaneous synchronization scheme is centralized. By contrast, in (3.18), the overall NMSE in the network is restricted in a distributed manner, namely, by reducing the NMSE within each cluster. Therefore, for clusters of clocks with different qualities, a unique synchronization frequency can be assigned to achieve the overall clock accuracy.

Meanwhile, the effect of synchronization frequency in simultaneous protocols is shown in Fig. 3.8. It is clear that by increasing the synchronization frequency, clusters with higher VRS are more beneficial. By contrast, the NMSE of nodes with a high-quality clock will be reduced not as significantly as other nodes. Therefore, increasing the synchronization frequency for nodes with larger VRS will cause more gain in terms of averaged NMSE.

The comparison between simultaneous protocol and proposed asynchronous protocol in terms of clock accuracy is shown in Fig. 3.9, where it can be observed that although some nodes are synchronized less frequently in the proposed scheme, the overall clock accuracy is still less than the pre-defined accuracy requirement, i.e., 1×10^{-6}s. Meanwhile, compared to the simultaneous scheme, this accuracy is even better due to the reason that the averaged NMSE in the network is mainly caused by clocks with larger VRS, who are assigned with a

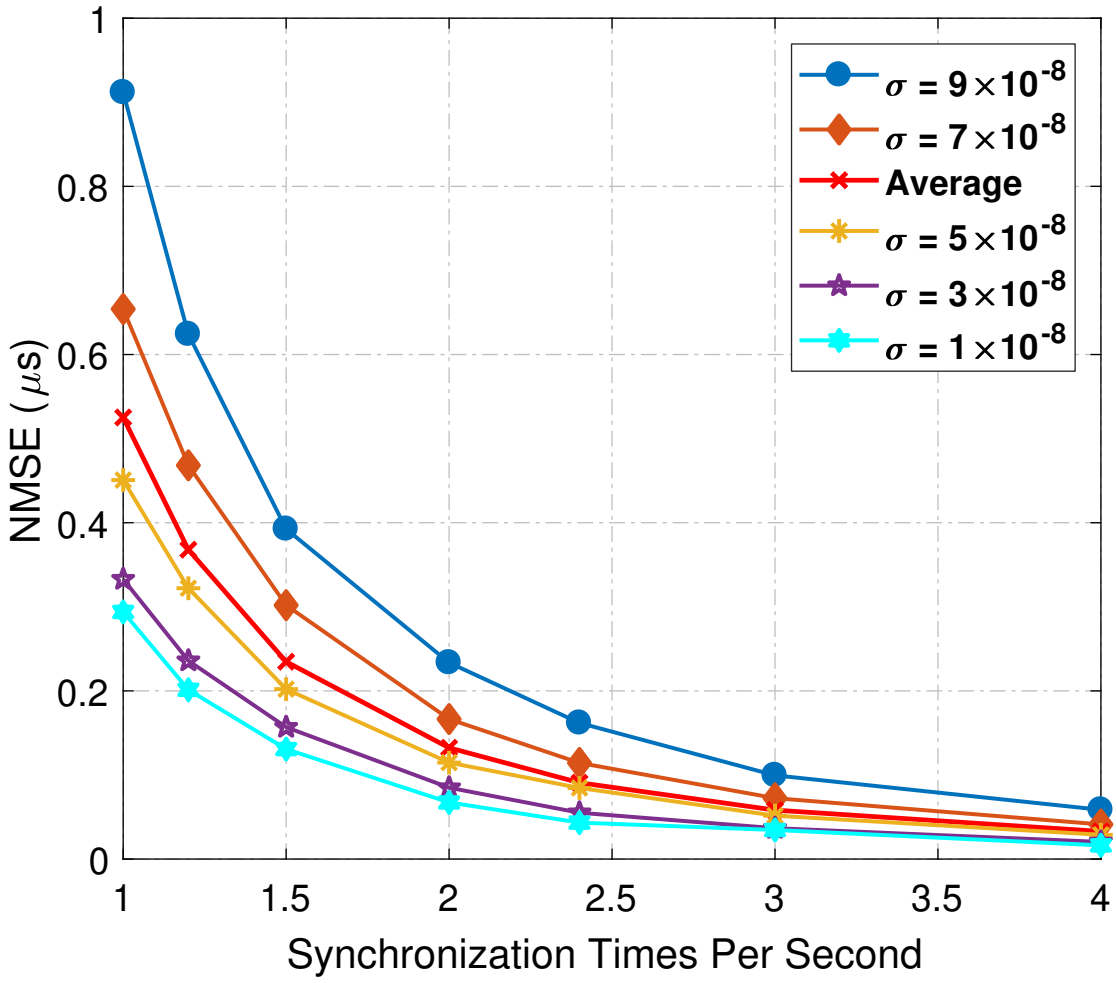


Figure 3.8: The effect of synchronization frequency with respect to the averaged NMSE in each cluster. Synchronization with the same frequency cannot achieve identical benefit for heterogeneous clocks.

higher synchronization frequency in the proposed scheme. Therefore, by using the proposed synchronization protocol, a higher averaged clock accuracy is achieved.

3.5.3.2 Packet Requirement Analysis

It can be observed from Fig. 3.9 that more synchronization actions are performed (when the line goes down) in the proposed scheme compared to the simultaneous case. However, during each synchronization process, only nodes in the same cluster are triggered. In other words, the number of nodes synchronized each time is less than the simultaneous protocol. Since the wireless communication resources are limited, it is meaningful to study the packets required

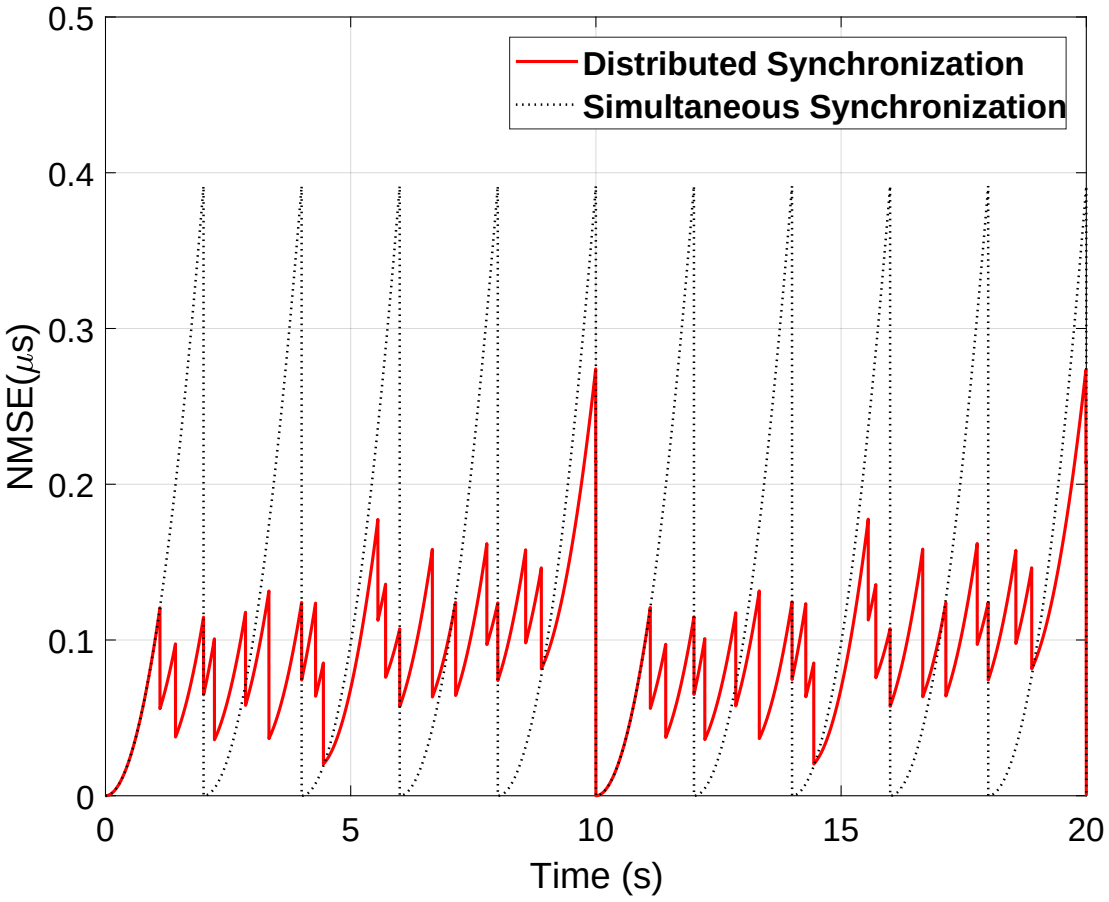


Figure 3.9: The evolution of NMSE with respect to time. When the lines go down, one or more synchronization actions are performed. The proposed method achieved accurate synchronization in a distributed manner.

to be delivered and resources required to be occupied during the synchronization process for further validation of the proposed protocol.

Fig. 3.10 demonstrates the overall packets required to be transmitted per second with different synchronization accuracy requirements. Clearly, with a higher accuracy requirement, there are more synchronization actions required to proceed in restricting the clock instability within the desired range. Compared to the simultaneous case, the proposed distributed approach always requires fewer packets to be delivered to guarantee the overall accuracy, meaning that more communication resources are saved in the synchronization with an intelligent clustering algorithm. In the case that clock accuracy is loose, the gap between these two schemes is even more significant, due to the reason that fewer nodes should be synchronized with lower

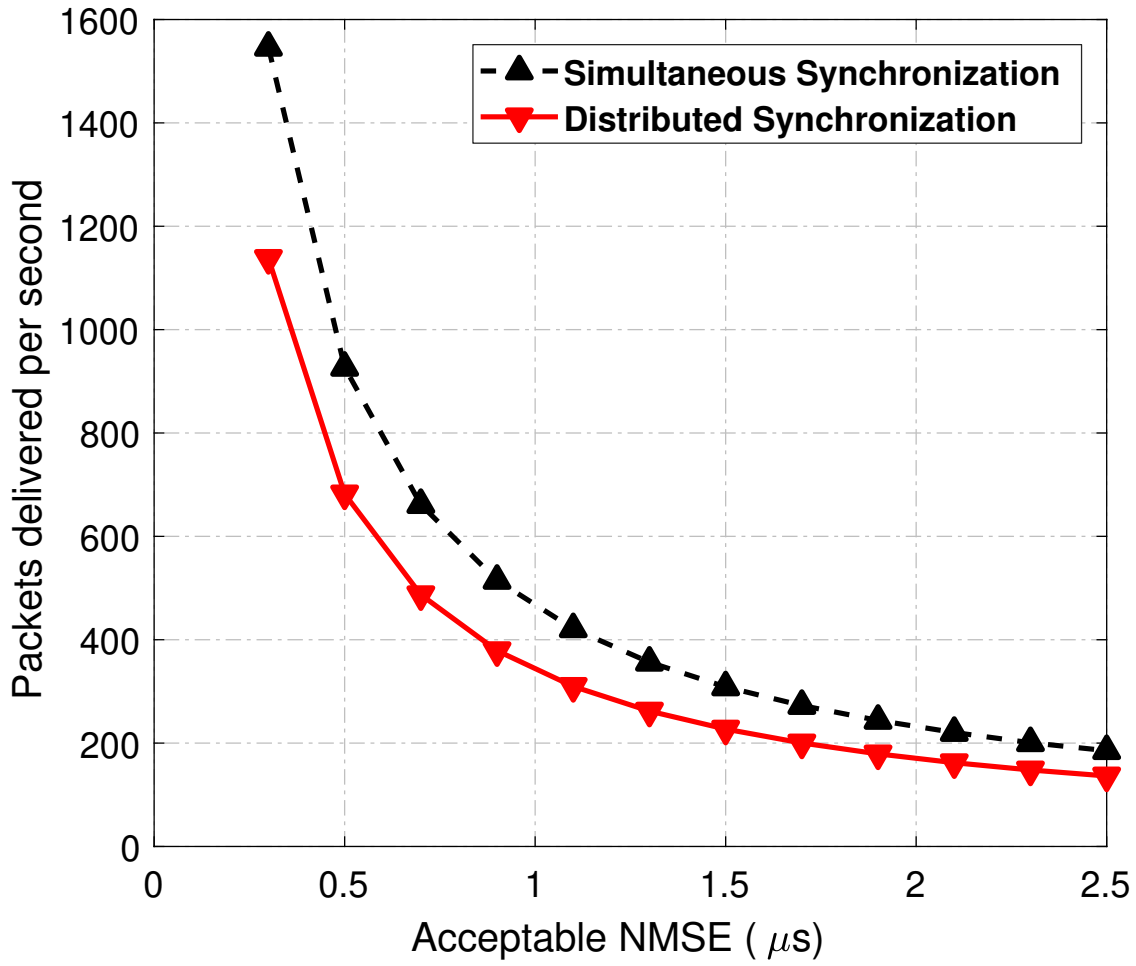


Figure 3.10: Comparison of averaged packets required per second in simultaneous protocol and the proposed one in terms of synchronization accuracy requirement in a network with 150 nodes.

accuracy requirements in the proposed protocol.

Meanwhile, Fig. 3.11 illustrates the effect of the total number of nodes in the network with respect to the packets delivered. The proposed protocol overwhelms the simultaneous one with different network scales since fewer packets are required for nodes with a smaller VRS. Meanwhile, with a larger number of nodes deployed in the system, more resources are preserved in the proposed distributed scheme. This is due to the reason that when the simultaneous synchronization scheme is adopted in larger-scale networks, more nodes are required to synchronize at the same time, leading to more limited communication resource and competitive contention of accessing the network. Thus, unavoidable packet losses will be caused, and a re-transmission mechanism is required to guarantee the delivery of the time information. By contrast, in the dis-

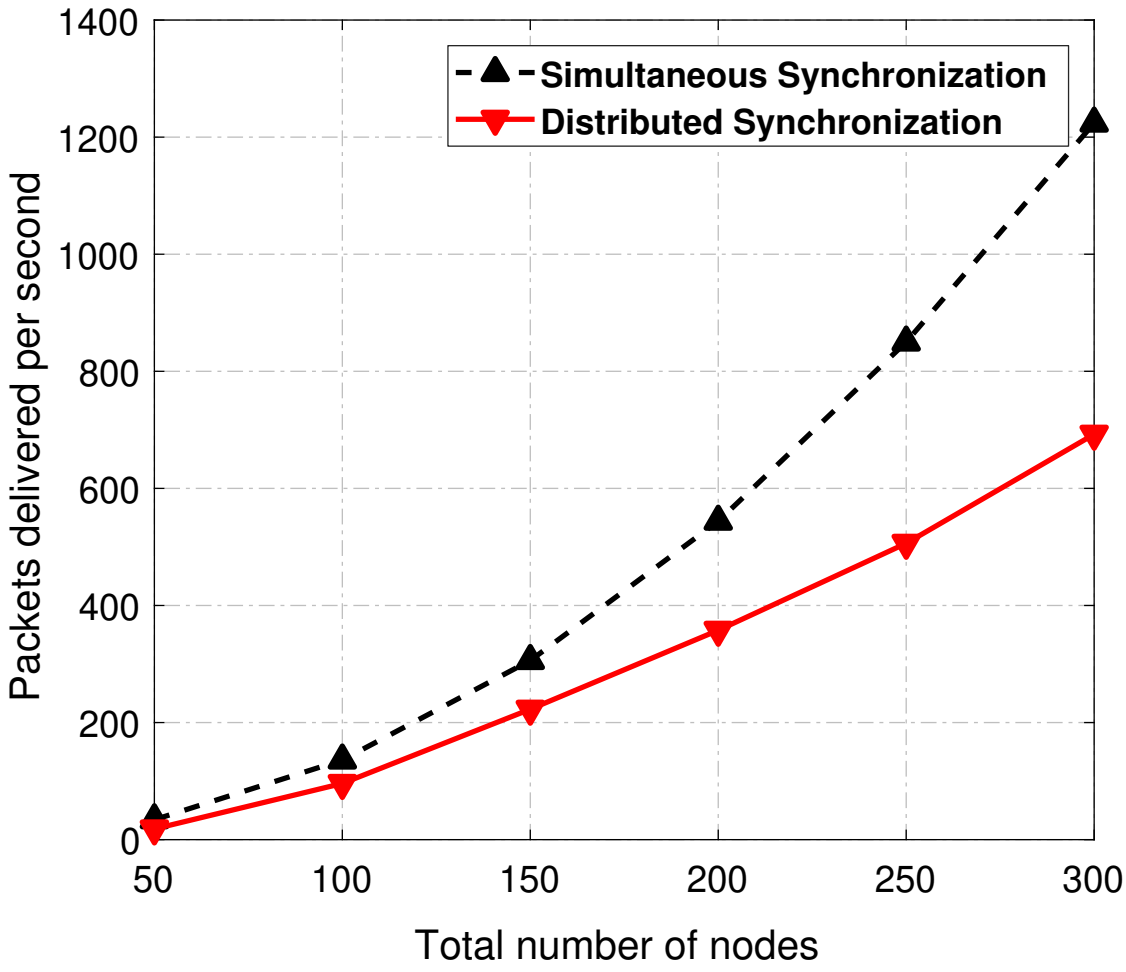


Figure 3.11: With the same accuracy requirement of $1.5 \times 10^{-6} s$ in networks with different scales, the proposed scheme will always save more resources for different network scales.

tributed synchronization protocol, fewer nodes will synchronize simultaneously. Packet losses will be significantly reduced, and the conveyance of timestamps will be more reliable. It is worth noting that with the proposed protocol, more communication resources can be saved for further application-oriented information transmission, leading to a better performance in the overall system.

3.5.3.3 Security Analysis

It should be mentioned that the previous simulations are achieved in the absence of malicious attacks and faulty nodes. Due to the ubiquity of malicious nodes in wireless communication applications, the proposed synchronization protocol should be still effective with potential ma-

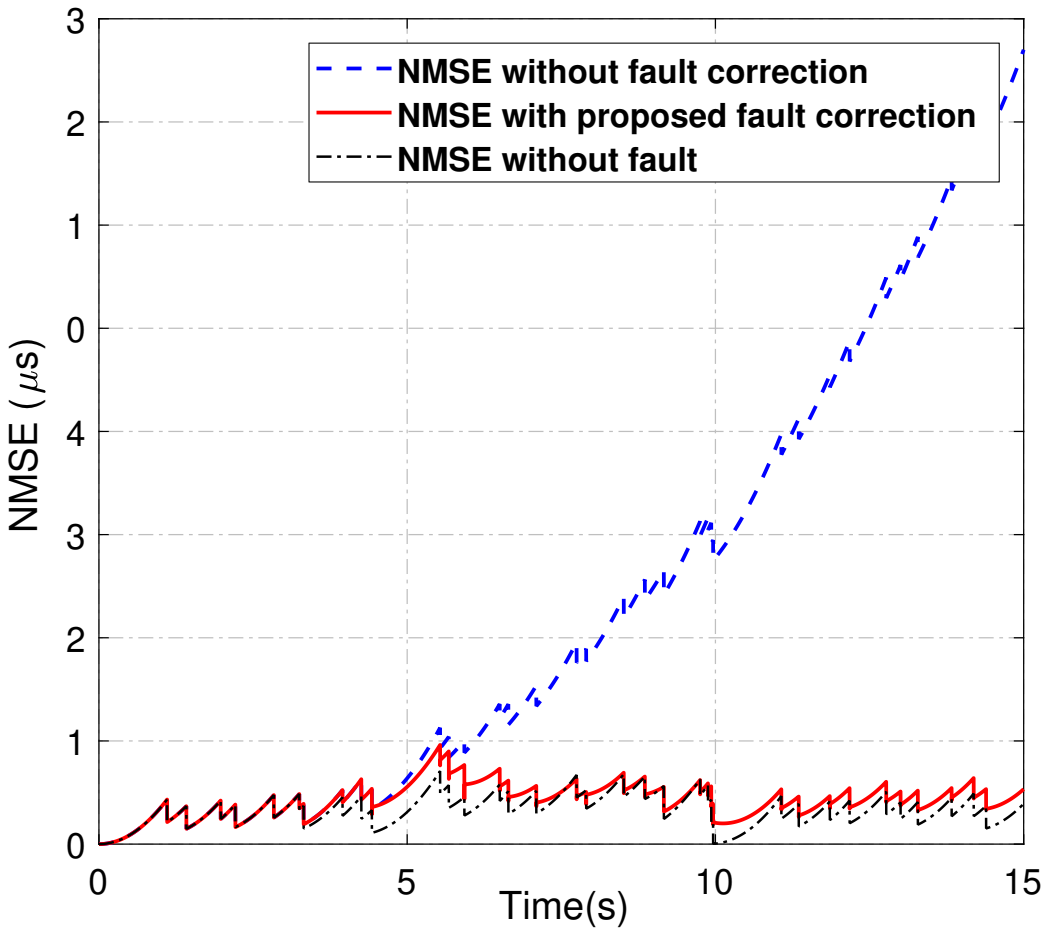


Figure 3.12: The effectiveness of the proposed fault node detection method. The malicious nodes are appeared and detected at around the fifth second, then predicted values are utilized for the further clock synchronization process.

licious attacks. In this simulation, three kinds of message manipulation attacks are considered, namely, malicious nodes try to transmit a random value instead of their real clock information, malicious nodes suddenly change their parameter, e.g., give out a piece of misleading topology information, and malicious nodes will confuse their CH by delayed time information.

- Malicious node will continuously generate fault value at its synchronization instant, pretending to be its local clock information, i.e.,

$$C_m(\tau) = v(\tau)$$

- Malicious node will suddenly change its topological information to mislead its cluster

head and cluster member, so that the synchronization of the clock could be interrupted.

- Malicious node will take advantage of delayed information and provide an outdated clock information to its cluster head, i.e.,

$$C_m(\tau) = C_m(\tau - d_m)$$

where d_m is the random and unpredictable delay occupied by malicious node m.

Fig. 3.12 compares the cases with and without malicious node detection algorithm. It can be observed that without fault detection mechanism, the clock error will increase dramatically after 5 seconds, where the faulty nodes are expected to behave maliciously. This increment will be unbounded since misleading information is exchanged continuously. Fortunately, with the proposed detection algorithm, all these three kinds of malicious attacks are resolved by detecting potential malicious nodes and using the reliable clock information instead. The synchronization accuracy is improved by adopting the proposed algorithm, and only a little difference is caused due to the imprecise prediction compared to the case without malicious nodes. However, it is worth noting that this algorithm will become less effective if the malicious node continually transmits eavesdropped clock information obtained from its neighbor. In this case, the fault detection algorithm will firstly detect it as a malicious node, but the clustering algorithm will regard the malicious node like a regular node with varied VRS, leading to slightly weakened synchronization accuracy in a short period. Finally, it is worth noting that, the number of malicious device will not affect the proposed two-tier abnormal detection scheme, due to the historical tracking of the VRS value on each clock. By comparing the estimated value with the real-time VRS of the potential malicious nodes, the abnormal device can be efficiently distinguished with high detection precision.

3.6 Chapter Summary

In this chapter, an intelligent clustering-based distributed synchronization protocol is proposed for local area industrial IoT systems to achieve reliable, packet efficient, and secure clock synchronization. To be specific, the overall synchronization process consists of two phases. The

first phase is responsible for initial skews correction and offsets compensation. After the first phase, all clocks will operate in the same frequency with different drift rates. In the second phase, all nodes in the network are grouped into several clusters according to their VRS values, which is relevant to their clock quality and the environment. One of the nodes is selected as the chief cluster head based on the clock quality to provide reference time while several cluster heads are elected, acting as connectors between the CCH node and cluster members. Different from simultaneous synchronization, in the proposed scheme, nodes with a high-quality clock are synchronized less frequently than those with an unstable clock. Meanwhile, according to their VRS values, a fault detection algorithm is also developed to overcome malicious attacks during synchronization processes. Simulation results proved that the proposed clock synchronization protocol could achieve a better performance in terms of accuracy and resource consumption. Furthermore, the proposed fault detection algorithm is capable of detecting harmful attacks and improving clock accuracy.

Chapter 4

Digital Twin Enabled Intelligent Distributed Clock Synchronization

Temporal cooperation among connected industrial equipment and infrastructures in an industrial Internet of things (IIoT) system hinges on low latency data exchange and accurate synchronization within sophisticated networks. However, the temperature-induced clock drift in connected industry facilities constitutes a fundamental challenge for conventional synchronization techniques due to dynamic industrial environments. Furthermore, a large variation of packet delivery latency in IIoT networks hinders the reliability of time information exchange, leading to deteriorated clock synchronization performance in terms of synchronization accuracy and network resource consumption. In this chapter, a digital-twin-enabled model-based scheme is proposed to achieve an intelligent clock synchronization for reducing resource consumption associated with distributed synchronization in fast-changing IIoT environments. By leveraging the digital-twin-enabled clock models at remote locations, necessary interactions among distributed IIoT facilities to achieve synchronization is dramatically reduced. The virtual clock modeling prior to clock synchronization helps to characterize each involved clock so that its behavior under dynamic operating environments is predictable at a remote location to avoid excessive synchronization-related data exchange and inaccurate estimations for clock calibration. An edge-cloud collaborative architecture is also developed to enhance the overall system efficiency during the development of remote digital-twin models. Simulation results demonstrate that the proposed scheme can create an accurate virtual model remotely for each

local clock according to the information gathered. Meanwhile, a significant enhancement on the clock accuracy is accomplished with dramatically reduced communication resource consumption in networks with different packet delay variations.

4.1 Introduction

The ongoing convergence of information and communication technologies (ICT) and vertical industry applications in the forms of advanced manufacturing, smart factories, and industry 4.0 will directly boost the efficiency, quality, and productivity of many industrial processes as well as reducing the corresponding production costs [82]. By integrating communications, networking, and computing technologies, future cyber-physical systems (CPS) and the industrial Internet of Things (IIoT) will enable the ubiquitous connectivity and interactivity among large-scale industrial infrastructures [2] [83]. Assisted by the ongoing expansion and deployment of the fifth-generation (5G) wireless technologies, timely large-scale sensing and controlling information exchange among distributed physical infrastructures are becoming realities [84], leading to frequent interactions among the connected facilities. However, the involvement of a massive number of devices and their associated frequent data exchange will result in unexpected latency, which hinders the intelligence and precognition of the distributed process in IIoT systems [85]. The long and nondeterministic network latency inevitably becomes a significant impediment for the timely information exchange and efficiency enhancement in distributed IIoT systems [86].

As a critical requirement to guarantee efficient distributed interaction, accurate clock synchronization within an IIoT system empowers the collaborative synergy among distributed physical elements [20]. Without a consistent clock reference among the industrial infrastructures, data packets associated with misaligned time indices will be interchanged, leading to inaccurate analysis of crucial data and decisions relying on stale data. The increasingly sophisticated industrial environment poses even more challenges on clock synchronization among all involved entities. On the one hand, with the extensive development of ICT technologies, an increasing number of factories will rely on 5G wireless technologies as one of their fundamental ways to meet critical requirements on time-sensitive applications in the near future [87].

Consequently, a heterogeneous IIoT network based on wired standards (e.g., field bus), traditional wireless standards (e.g., wirelessHART), and arising 5G standards (e.g., URLLC) is emerging [88], leading to new synchronization challenges including non-deterministic latencies and cross-standard synchronization for distributed nodes using various communication standards [21]. On the other hand, the dynamic industry operating environment of an IIoT system makes the synchronization process more challenging. As previously studied and validated [48] [68], clocks in complicated environments are susceptible to drift compared to those in moderate residential and office circumstances. The temperature-induced clock drifting inherent to inexpensive crystal oscillators will inevitably result in inconsistent clock outputs among IIoT devices, necessitating the situation-aware synchronization to fully address susceptibility of distributed devices to the external influence.

Traditional synchronization techniques adopted in an IIoT environment, including precision time protocol (PTP) [72] and flooding time synchronization protocol (FTSP) [89], are typically achieved through frequent calibration of the target clock by minimizing the observed clock deviation from the time reference, without understanding the intrinsic model of the clock drift. Consequently, a distributed synchronization process based on such mechanisms in a sophisticated industrial environment could become highly complex, leading to significantly reduced synchronization accuracy and efficiency. With the emergence of digital twin, a situation-aware and model-based distributed clock synchronization for IIoT systems could be enabled. By definition, digital twin is the digital counterpart of a physical infrastructure established through comprehensive observation, modeling, and simulation processes that can reflect the real-time relation between the physical entity and its virtual representation [90]. Consequently, digital twin is indispensable to bridge the gap between the physical world and the virtual world while enhancing the seamless integration between these two domains, which tends to be unattainable without the involvement of accurate clock synchronization among all physical and digital entities. Therefore, it is critical to design an accurate and efficient clock synchronization technique for IIoT systems coupled with the digital twin equivalents to enhance the coherent interaction among all the constituents. Meanwhile, the establishment of the digital twin of an IIoT system can promote the understanding of its distributed physical constituents. The behavior of all physical elements in an IIoT synthesis can be thoroughly modeled and predicted by their

correlated digital counterparts, which are originated and updated exclusively based on the distributed information collected from their physical equivalents [91]. By leveraging the strength of digital twin, the varying pattern of the distributed clocks in an IIoT system is predictable under fluctuating external environment, so that the synchronization process can be adapted with situation-awareness. Meanwhile, by adopting the well-established digital twin at each local device, the real-time behavior of each clock can be estimated, based on which the proactive synchronization with reduced network resources can be expected.

Motivated by these considerations, a digital-twin-enabled distributed clock synchronization scheme is proposed in this chapter to tackle the previously mentioned synchronization challenges in IIoT systems. To be more specific, the primary contributions of this chapter comprise the following two aspects. Firstly, an edge-cloud collaborative system architecture is proposed to establish the digital-twin models. The data generated at each local device is efficiently processed at an edge device to minimize the effect of uncertain network latency on the timely analysis. Secondly, the characteristics of each clock are modeled by exploring the consecutive time information uploaded from the distributed devices, based on which a digital twin counterpart is established to predict the clock skew for each node under various operating environments. By leveraging the digital-twin-enabled clock skew estimation, the clock accuracy is improved with reduced consumption of network resources during the distributed clock synchronization.

The remainder of this chapter is organized as follows. Related work in terms of clock synchronization under network uncertainties, temperature-compensated skew correction, and digital twin-related designs are summarized in Section 4.2. The overall physical system description including the edge-cloud collaborative architecture and heterogeneous clock models is illuminated in Section 4.3, while the realization of digital-twin-enabled clock synchronization is designed in Section 4.4, including information acquisition, model establishment, and clock skew correction. Simulations are carried out to demonstrate the effectiveness of the proposed scheme on the aspects of model validation, clock accuracy improvement, and network overhead reduction in Section 4.5, followed by the conclusion in Section 4.6.

4.2 Related Work

Clock synchronization in the industrial environment has been widely investigated in recent years. In this section, related studies in terms of clock synchronization considering network latency, temperature-induced clock drifting issues, and the recent advancements on digital twin in industrial applications are summarized in detail.

Many synchronization techniques have been developed to overcome the impact of uncertainty and asymmetrical network latency on clock synchronization accuracy. This effect is typically alleviated by designing more robust synchronization protocols, e.g., using prioritized data transfer protocols [29] to guarantee the channel access for sensors with critical tasks so that the synchronization precision can be improved. However, those efforts generally depend on the feasibility of modifying communication protocols. In dealing with the random network latency, the authors in [92] tried to jointly evaluate clock parameters and the network latency by proposing a low-complexity maximum-likelihood estimator. Meanwhile, a constant gain approach is designed [93] for consensus-based synchronization with random bounded communication delays. However, the effectiveness of these methods primarily relies on the accurate estimation and characterization of the random network latencies in the system.

Moreover, the temperature-induced clock drifting for inexpensive oscillators in dynamic environments is explored and analyzed in the literature. According to the quadratic relation between temperature and oscillation frequency summarized in [48], a temperature-assisted and self-calibrated clock synchronization mechanism is proposed for sensor networks [55] by dynamically compensating clock skews based on the working temperature. The improvement of clock accuracy depends on the assumption that the relation between its frequency and the corresponding temperature always follows the theoretical expression. A different scheme on environment-aware clock skew estimation and compensation is proposed for the synchronization in wireless sensor networks [51] by designing a Kalman filter to evaluate the impact of the operating temperature, which poses a more stringent requirement on the processing capability for each sensor node. Additionally, the authors in [54] proposed a neural network-based approach to explore the characteristics of the synchronization error, which is mainly affected by the external temperature. However, the feasibility of such a method is dubious due to the high

cost of adopting the learning-based algorithm in distributed inexpensive devices. It also should be noted that none of these studies considered the effect of network latency on the clock relationship estimation and calibration, which motivated the design of the situation-aware clock synchronization protocol with dynamic network conditions.

On the other hand, as one of the enabling technologies in IIoT systems, digital twin has received increasing attention in the last few years. Digital twin can duplicate and imitate the physical characteristics of distributed entities to demonstrate the natural trend of each party based on historically collected data, which enables remote analysis, modeling and prediction of the future behavior for device of interest. Based on this observation, a digital twin-assisted fault diagnosis paradigm [58] is developed using the deep-learning methodology. A model with the high-fidelity is designed in the virtual domain to train the learning model so that the dynamics of data is thoroughly traceable. Similarly, digital twin-based product design, manufacturing, and service provisioning with the assist of big data analysis are proposed in [94], although lacking a comprehensive demonstration of the data regulation and analysis. Inspired by the potential of digital twin that the data can be acquired and traced in a real-time manner, the authors in [95] implemented an online analysis digital twin system to obtain a real-time solution for the power grid. These investigations indicate that digital twin is effective in dealing with the massive data on real-time acquisition, accurate analysis, and timely decision-making.

4.3 Cloud-Edge Collaborative Architecture

In this chapter, a three-tier cloud-edge collaborative architecture with heterogeneous physical clocks distributed in an IIoT system is adopted to form the foundation of the proposed digital-twin-enabled clock synchronization.

4.3.1 Overall System Architecture

The overall architecture of an IIoT system designed to maintain a desirable clock accuracy at every clock within the network is shown in Fig. 4.1, which is organized as a three-layer hierarchy, namely, cloud center, edge devices, and end-nodes. In this section, the function of each layer and the interaction among them will be introduced in detail.

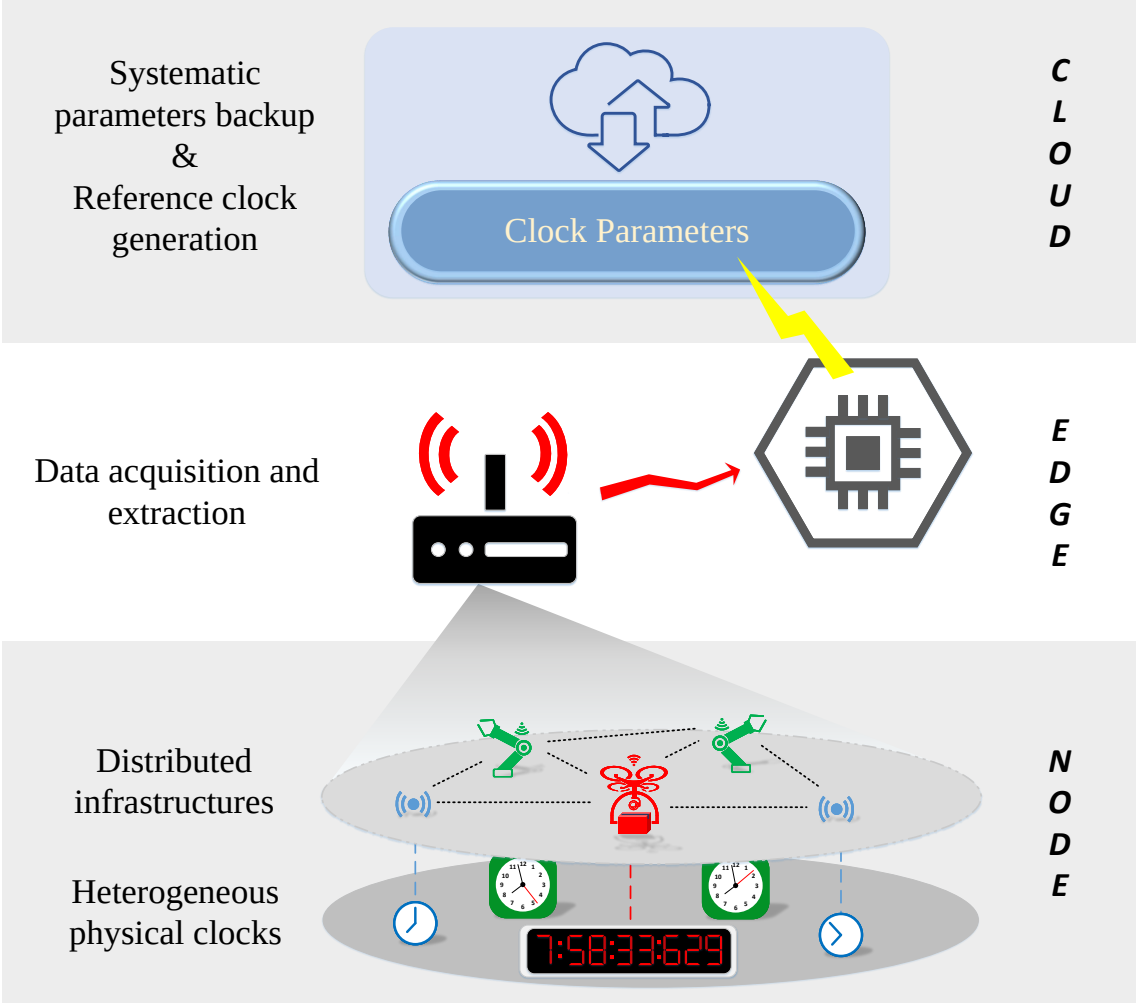


Figure 4.1: The overall architecture of a physical system consists of a cloud center for data backup, some edge devices provide preliminary data processing capability, and end-devices with clocks required to be synchronized occasionally.

Node devices, at the most fundamental layer, are various classes of end-devices equipped with a clock of diverse conditions, which is required to be synchronized periodically regarding the time reference. All types of time-tracking devices, including sensors, actuators, controllers, schedulers, and unmanned vehicles, can be considered as node devices. All clocks in an IIoT system will function uniquely due to their diverse operating environments, inconsistent manufacturing specifications and packaging materials for the crystal oscillators, as well as their different capacities of receiving the time reference for local clock modification. Therefore, the clock of each facility will evolve with different drifting rates, and conduct an incoherent timestamp at any instant. This nondeterministic clock inaccuracy under complicated indus-

trial operating and communication environments necessitates frequent and resource-consuming synchronization actions associated with massive time information conveyance for maintaining a global time consistency within the network.

Edge devices are intelligent devices with higher processing capacity, e.g., smart gateways, assigned to accomplish preliminary and uncomplicated computation physically close to the node devices [96]. As the intermediate layer of the entire system, edge devices are responsible for collecting physical information, including timestamps and temperature, from the node devices, supervise relatively straightforward data processing, and convey the outputs to the cloud center for further computation or backup. As a consequence, edge devices can enhance the distributed intelligence during real-time raw data gathering and processing while alleviating the unsatisfactory impact of the significant network latency between node devices and the cloud center. Each edge device will be responsible for collecting and processing the clock information only from the node devices within its coverage so that excessive or unbalanced computation burden is avoided. Generally, edge devices are gateways or IIoT devices assigned by the cloud center according to a series of metrics, including its functionality, processing capacity, and the physical topology, which is discoverable by adopting appropriate algorithms, e.g., [97].

The functions of a cloud center in the proposed scheme are threefold. Most fundamentally, the cloud center serves as the time reference, according to which every device in the network will synchronize its local clock. On recognizing the potential necessity of the clock modification, the cloud center will disseminate this reference time to the entire system progressively. Meanwhile, the cloud center is responsible for the initial synchronization of the selected edge devices. The rest of this chapter will focus on the description of the digital-twin model establishment and clock synchronization between the edge and node devices under the assumption that a digital twin of each edge device is established and, its clock is accurately synchronized according to the same procedures. Furthermore, the cloud center acts as the data center of the overall system for data storage, management, and backup. The parameters obtained historically or updated periodically from the edge devices will be preserved in the cloud, which is physically distance from other devices.

The coherent collaboration among these three layers with the efficient data exchange among

all involved elements plays a critical role in supporting the establishment of a situation-aware synchronization protocol. Consequently, the extensive understanding of the intrinsic parameters of each clock at distributed physical entities is indispensable for the accomplishment of the successive synchronization design.

4.3.2 Heterogeneous Clock Model

In the node layer of the proposed hierarchical architecture, each end-device encloses a quartz-crystal oscillator-driven clock. For a large-scale IIoT system with the complicated operating environment, these clocks generally behave differently due to the existence of initial clock offsets and skews, which are mainly caused by the original manufacturing defect of their oscillators, as well as the successive clock drifting resulted from the effect of the external operating environment, e.g., temperature. Especially when an inexpensive oscillator embeds in each device for reducing the implementation cost, the inconsistent and varying drifting incurred will lead to long-term clock inaccuracy. To be more specific, the clock at node i is a function of its time-varying clock skew α_i and its constant clock offset β_i. According to [98], it can be written as

$$C_i(t) = \alpha_i(t)t + \beta_i \tag{4.1}$$

where t is the real time. Generally, the clock skew α_i of an oscillator is caused by the dynamic frequency variation from its dominated frequency, while the clock offset is typically incurred by the various initial phases at the distributed clocks.

The frequency-drifting of a clock is originated from both the inherent defect and its congenial disposition, e.g., its susceptibility of temperature. According to the correlation between temperature and oscillation frequency validated in [48], the real-time frequency of the crystal oscillator at node i is defined as

$$f_i = f_{i0}(1 + \eta_i(T(t) - T_{i0})^2) \tag{4.2}$$

where $T(t)$ is the operating temperature at the instant t, while f_{i0}, η_i, and T_{i0} are the dominated frequency, temperature sensitivity factor, and ideal operating temperature of the clock at

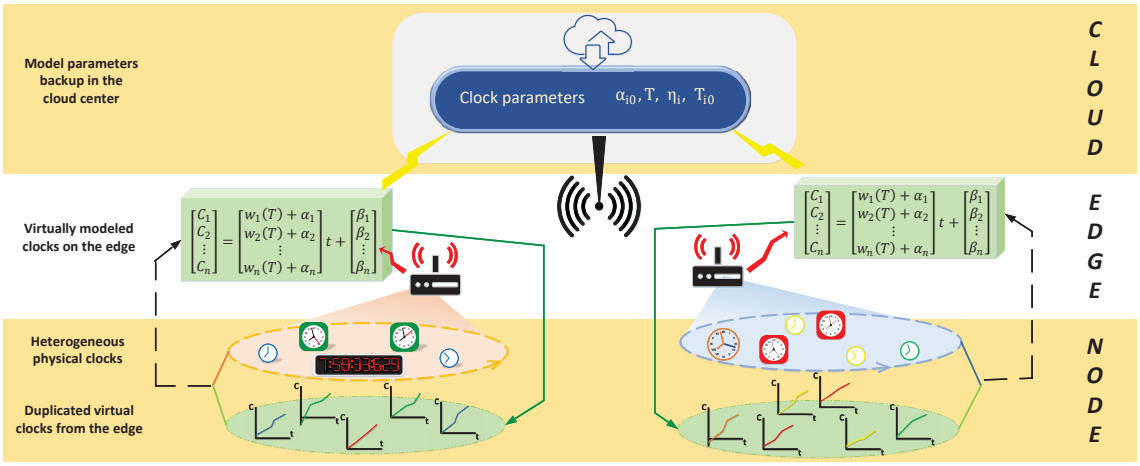


Figure 4.2: The cloud-edge-node hierarchy after inducing its virtual counterparts.

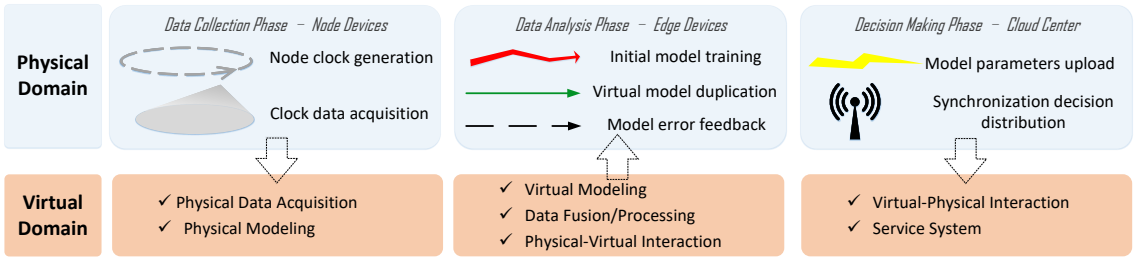


Figure 4.3: The icon description and the dataflow for the physical-virtual interaction during clock synchronization.

node i, respectively. Affected by the frequency-drifting, the overall clock skew at the instant t cumulated from its initial skew α_{i0} is written as

$$\alpha_i(t) = 1 + \eta_i(T(t) - T_{i0})^2 + \alpha_{i0} \tag{4.3}$$

Consequently, the clock output at time t can be derived by substituting (4.3) into (4.1), while its error $e(t)$ regarding the absolute time t is given by

$$e(t) = C_i(t) - t = (\eta_i(T(t) - T_{i0})^2 + \alpha_{i0})t + \beta_i \tag{4.4}$$

which is affected by its initial inaccuracy as well as its intrinsic temperature-related parameters.

However, it is worth noting that none of those skew-related parameters for the distributed devices is identical, even if they are manufactured from the same batch. Meanwhile, this dif-

ference for heterogeneous entities in a IIoT synthesis will be even more critical, indicating that it is infeasible to derive a universal expression to represent all clocks. Especially when applications with extreme time-sensitivity are adopted in the IIoT system, the clock inconsistency incurred from the skew will be intolerable and fatal. Therefore, instead of adopting the general model given by (4.1), an appropriate approach to achieve parameter determination is of the utmost importance to further investigate an individualized model of each clock.

4.4 Digital-Twin-enabled Clock Synchronization

Based on the information from the heterogeneous clocks, a digital-twin system can be established to fully investigate their characteristics, which can be further used for clock synchronization. The expanded system architecture with the involvement of a digital-twin system is initially introduced.

4.4.1 Physical-Virtual-Collaborative System Design

The seamless collaboration between the physical process and the virtual model is indispensable in realizing the timely data analysis and conscious decision-making throughout its life cycle. In the proposed clock synchronization design, the physical elements extracted from the distributed infrastructures are a set of heterogeneous clock outputs, based on which models with various parameters are attainable. The physical-virtual-collaborated system architecture is shown in Fig. 4.2, which is an enlargement of the three-tier hierarchy introduced in 4.3.1 after initiating its digital twin counterpart.

As previously stated in 4.3.2, each device in the system is associated with an oscillator-driven clock, which generates a sequence of timestamps continuously. The timestamps given by each clock are always inconsistent, due to the heterogeneity of the enclosed oscillators. Each group of nodes will send the timestamps during the network initialization period continuously to their dedicated edge device, who is responsible for recording and investigating the behavior of these end-devices so that a virtual representative for each individual can be created in the digital twin domain. After conducting the initial digital model for all clocks within its coverage, the edge device will send a duplication of each model to every end-device for

validating the correctness of the coefficients. After the training of several iterations based on the successive observation and feedback, a refined model can be established, while the model parameters will be transmitted to the cloud center as a backup. As shown in (4.4), the critical parameters required to be stored in the cloud center including the initial clock skew α_i, temperature sensitivity factor η_i, and its ideal working temperature T_{i0}. Meanwhile, the cloud center will make decisions on the reference time dissemination to guarantee the overall clock accuracy in the network.

The overall clock synchronization scheme enabled by the digital twin system is shown in Algorithm 3 and the dataflow of the proposed digital twin-based clock synchronization scheme is shown in Fig. 4.3, where the entire dataflow comprises three successive phases, namely, data collection phase aiming at physical data acquisition, data analysis phase for the model establishment and its subsequent refinement, as well as the decision-making phase, which is responsible for finalizing the model parameters and calibrating clock offsets. Instead of organizing the devices into the cloud-edge-node hierarchy according to their classes, the overall dataflow is classified based on the intention of each interconnection among the entities involved. Moreover, those three phases are mapped into the digital twin domain with their corresponding components to demonstrate their ubiquitous interactions.

A more detailed physical-virtual-collaborative procedure of the digital-twin model establishment between the edge device and its node devices is shown by the sequence diagram in Fig. 4.4, which consists of the three different phases. After the initialization period, all nodes in the network are orchestrated into a cloud-edge-node architecture with the assignment of the edge devices. The collaboration initiates from the physical information collection at the local devices, where nodes will start to generate local timestamps under different temperatures. The local information consists of the generated timestamps and the corresponding temperatures will be successively transmitted to its superior edge device, which will be responsible for investigating the characteristics of each clock. Based on the obtained information, a digital-twin model will be formed in the node device, and corresponding feedback information will be delivered to the edge device for necessary update. Details of each phase will be introduced in the following subsections.

Algorithm 3 Digital twin-based clock model establishment and skew synchronization between nodes and edge devices

Select M edge devices in the network
for Each edge device E_j **do**
  Obtain the local information I_{ij} as Eq. (4.5)
  for Each node device i belongs to E_j **do**
    for Iterations fewer than its upper bound **do**
      Calculate the clock error evolution $\hat{\epsilon}_i$
      Estimate the coefficients $\hat{\boldsymbol{p}}_i$
      Establish the virtual skew model $\hat{\alpha}_i$ according to Eq. (4.14) and adopt it to i
      Generate feedback-based coefficients $\check{\boldsymbol{p}}_i$
      if training accuracy < threshold **then**
        Record the final $\check{\boldsymbol{p}}_i$ at E_j
        break
      end if
    end for
    Upload $\check{\boldsymbol{p}}_i$ to the cloud center as backup
    Distribute $\check{\boldsymbol{p}}_i$ to each node device
  end for
  Node i generates a virtual skew according to $\check{\boldsymbol{p}}_i$ based on Eq. (4.19)
  Node i corrects its local before network operation
end for

4.4.2 Clock Information Acquisition

As the external temperature has an apparent effect on the clock skew, it is meaningful to collect a sequence of timestamps under each operating temperature so that the relation between the clock variation and its temperature is discoverable. According to the observation that the temperature in an IIoT system typically varies gradually, a series of l local clock timestamps at each node device is continuously generated at the same temperature. Meanwhile, a total number of k distinct temperatures are selected for the relationship analysis. In other words, the node i will deliver its local clock information as one timestamp to its edge device, while each timestamp is denoted in the form of $C_i(t_k^l)$, indicating the l^{th} clock information obtained under

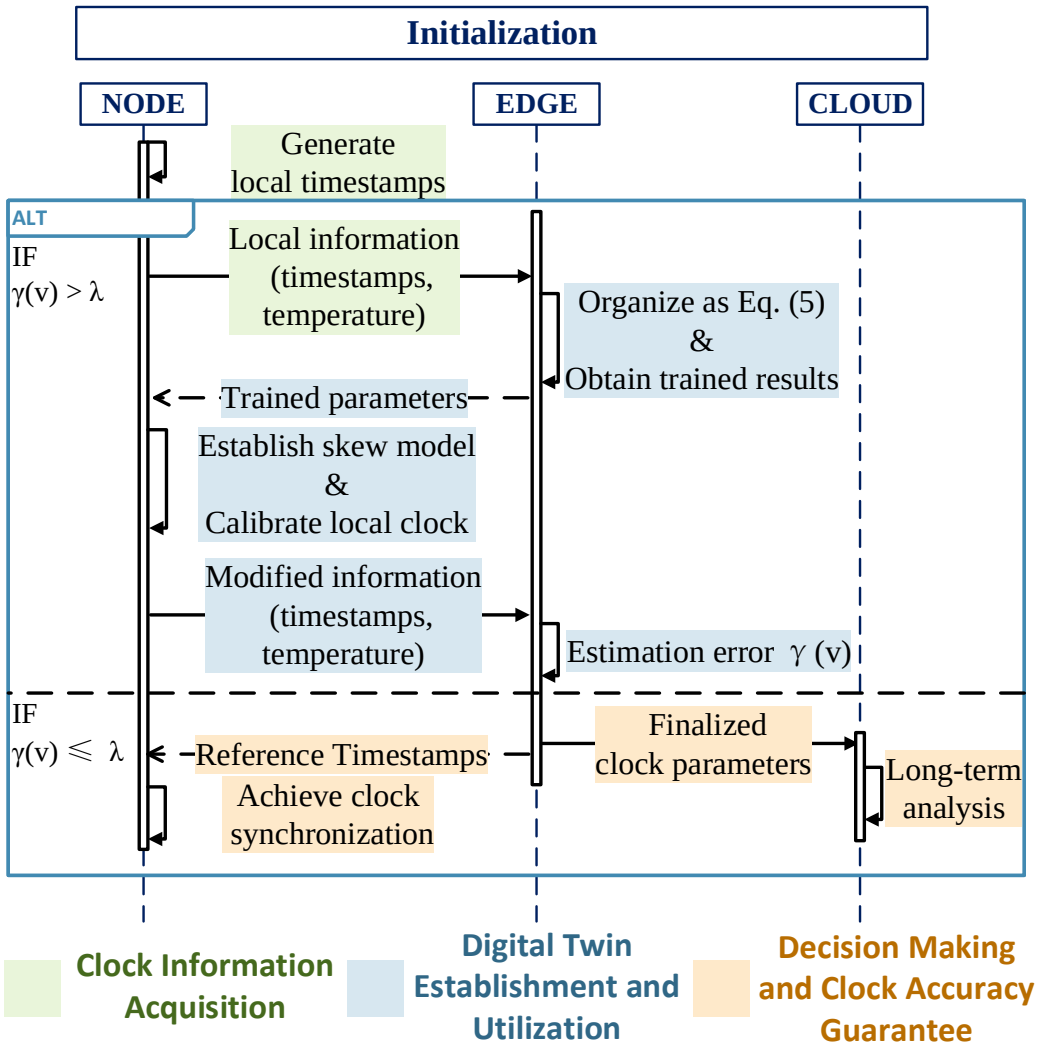


Figure 4.4: The sequence diagram of the digital twin-based clock model establishment and skew synchronization between a node and its edge device in the proposed cloud-edge collaborative architecture.

the operating temperature T_k at the node device i. After an apparent change of the operating temperature, each local device will update its operating temperature T_k associated with the timestamps to its superior device for the successive relationship discovery purpose.

After the reception of each timestamp $C_i(t_k^l)$ from its subordinate end-device i, the edge device E_j will record its local clock value $C_{ij}(t_k^l)$ for further calculation. As previously stated in 4.3.1, we assume that all edge devices were already synchronized according to the cloud center with the same approach shown in Fig. 4.4. Therefore, once a packet is received at the edge device, its timestamp recorded is the addition of the absolute time that the packet is transmit-

ted at the node device i and the network latency inevitably induced between these two nodes, namely, $C_{ij}(t_k^l) = t_k^l + d_{ij}(t_k^l)$. The overall latency is assumed to be random and nondeterministic due to the existence of packet delay variation (PDV) inherent to the network uncertainties, e.g., access contention, channel asymmetry, and dynamic communication environment. The PDV is generally time-varying, manifested as the difference of delays when delivering two successive packets between the same pair of nodes. As the PDV dominates the uncertainty during timestamp calculation, larger PDV will lead to significantly degraded synchronization performance. The network delay can be denoted by $d_{ij}(t_k^l) = \bar{d}_{ij} \pm \frac{1}{2}\delta_{ij}$, where $\bar{d}_{ij}$ is the expectation of the delay between node i and E_j, while δ_{ij} is the PDV affected by its communication protocol and network conditions, and its factor is adopted for further derivation simplicity. At the edge device E_j, the clock information delivered from one of its members i can be recorded and regulated into a matrix, shown as

$$\hat{I}_{ij} = \begin{bmatrix} T_i(t_1) & T_i(t_2) & \cdots & T_i(t_k) \\ C_i(t_1^1) & C_i(t_2^1) & \cdots & C_i(t_k^1) \\ \vdots & \vdots & \ddots & \vdots \\ C_i(t_1^l) & C_i(t_2^l) & \cdots & C_i(t_k^l) \\ C_{ij}(t_1^1) & C_{ij}(t_2^1) & \cdots & C_{ij}(t_k^1) \\ \vdots & \vdots & \ddots & \vdots \\ C_{ij}(t_1^l) & C_{ij}(t_2^l) & \cdots & C_{ij}(t_k^l) \end{bmatrix} = \begin{bmatrix} \mathbf{T_i(t_k^l)} \\ \mathbf{C_i(t_k^l)} \\ \mathbf{C_{ij}(t_k^l)} \end{bmatrix} \tag{4.5}$$

which will be further used for the successive preprocessing and model training.

4.4.3 Digital Twin Establishment and Utilization

The function of the data analysis layer in the proposed synchronization scheme can be classified into clock information preprocessing, initial clock model establishment, and feedback-based model update, which jointly develop a comprehensive understanding of each physical clock according to its time information (4.5) regulated at the superior device. Correspondingly, the data analysis layer can be mapped as data fusion, virtual modeling, and physical-virtual interaction in the digital twin domain.

4.4.3.1 Clock Information Preprocessing

The purposes of data fusion are twofold, namely, preliminarily alleviating the effect of random network latency and extracting critical information from the massive timestamps delivered. Based on (4.5), it is straightforward to obtain the clock error $e_i(t_k^l)$ at the time instant t_k^l under the external temperature T_k for the node i as

$$e_i(t_k^l) = C_i(t_k^l) - C_{ij}(t_k^l) \tag{4.6}$$

By comparing (4.6) with (4.4), the clock error can be rewritten as

$$e_i(t_k^l) = (\eta_i(T_i(t_k) - T_{i0})^2 + \alpha_{i0})t_k^l + \beta_i - d_{ij}(t_k^l) \tag{4.7}$$

Similarly, for the $(l+1)^{th}$ sampling under the same temperature $T_i(t_k)$, the clock error is given by

$$e_i(t_k^{l+1}) = (\eta_i(T_i(t_k) - T_{i0})^2 + \alpha_{i0})t_k^{l+1} + \beta_i - d_{ij}(t_k^{l+1}) \tag{4.8}$$

By subtracting (4.8) from (4.7), the clock error evolution between the two consecutive samplings can be written as

$$\epsilon_i^l(T_k^i) = (\eta_i(T_i(t_k) - T_{i0})^2 + \alpha_{i0})\tau_k + (d_{ij}(t_k^{l+1}) - d_{ij}(t_k^l)) \tag{4.9}$$

where τ_k is the fixed sampling interval between t_k^l and t_k^{l+1}. By taking the average of $\epsilon_i^l(T_k^i)$ for all the l samplings, the variation of clock errors can be obtained as

$$\hat{\epsilon}_i(T_k^i) = (\eta_i(T_i(t_k) - T_{i0})^2 + \alpha_{i0})\tau_k \pm \delta_{ij} \tag{4.10}$$

which is the effect of the operating temperature on the clock outputs under the network uncertainty δ_{ij}. According to (4.10), the $\epsilon_i(T_k)$ of the node N_i is quadratically related to its operating environment, while the coefficients are discoverable from the consecutive observations under diverse temperatures. In summary, the critical information extraction based on the clock infor-

mation recorded in (4.5) is given by

$$\begin{bmatrix} \mathbf{T_i}(\mathbf{t_k^l}) \\ \mathbf{C_i}(\mathbf{t_k^l}) \\ \mathbf{C_{ij}}(\mathbf{t_k^l}) \end{bmatrix} \rightarrow \begin{bmatrix} \mathbf{T_i}(\mathbf{t_k}) \\ \hat{\boldsymbol{\epsilon}}_\mathbf{i}(\mathbf{T_k^i}) \end{bmatrix} \tag{4.11}$$

4.4.3.2 Initial Model Training and Formation

Based on the critical information extracted, the edge device will build a model for each of its subordinates to track the trend of its local clock, which is affected by the physical characteristics of its enclosed crystal oscillator, as shown in (4.10). Based on the error difference $\hat{\boldsymbol{\epsilon}}_\mathbf{i}(\mathbf{T_k^i})$ and its related temperature $\mathbf{T_i}(\mathbf{t_k})$ summarized in (4.11), a Vandermonde matrix can be established, which will result in a linear relationship, shown as

$$\begin{bmatrix} T_i^2(t_1) & T_i^1(t_1) & 1 \\ T_i^2(t_2) & T_i^1(t_2) & 1 \\ \vdots & \vdots & \vdots \\ T_i^2(t_k) & T_i^1(t_k) & 1 \end{bmatrix} \begin{bmatrix} \hat{p}_{i1} \\ \hat{p}_{i2} \\ \hat{p}_{i3} \end{bmatrix} = \begin{bmatrix} \hat{\epsilon}_i(T_1) \\ \hat{\epsilon}_i(T_2) \\ \vdots \\ \hat{\epsilon}_i(T_k) \end{bmatrix} \tag{4.12}$$

where the vector $\hat{\boldsymbol{\epsilon}}_\boldsymbol{i}$ is the system output correspondingly attained from the system input $\boldsymbol{T_i}$. Based on the observation, the polynomial coefficients $\boldsymbol{p_i}$ can be derived by solving

$$\hat{\boldsymbol{p}}_\boldsymbol{i} = \boldsymbol{T_i}^{-1}\hat{\boldsymbol{\epsilon}}_\boldsymbol{i} \tag{4.13}$$

from which the temperature-sensitivity factor $\hat{\eta}_i$, ideal operating temperature $\hat{T}_{i0}$, and the initial clock skew $\hat{\alpha}_{i0}$ can be estimated. These three coefficients will be distributed to each local device for further validation and updates.

Once the model coefficients are received at each local device, a virtual skew can be obtained as

$$\hat{\alpha}_i(t) = 1 + \hat{\eta}(T_i(t) - \hat{T}_{i0})^2 + \hat{\alpha}_{i0} \tag{4.14}$$

which is the digital twin counterpart of the real skew at the local clock. This physical clock

skew can be calibrated from dividing the local clock by the virtual skew, given by

$$\hat{C}_i(t) = \frac{C_i(t)}{\hat{\alpha}_i(t)} = \frac{\alpha_i(t)t + \beta_i}{\hat{\alpha}_i(t)} \tag{4.15}$$

where β_i is the constant initial offset. A precise modeling of the clock skew $\hat{\alpha}_i$, i.e., $\hat{\alpha}_i(t) = \alpha_i(t)$, will result in $\hat{C}_i(t) = t + \hat{\beta}_i$, indicating that the local clock at node i will operate with the identical rate as the reference clock at its edge device with a tiny constant offset, which can be easily eliminated via any offset compensation process (OCP) once, e.g., two-way packet exchange used in [99]. However, due to the network uncertainty δ_{ij} involved during clock modeling, the calibrated clock will operate with a slight deviation from its ideal scenario, leading to the necessity of successive model update according to the feedback information.

4.4.3.3 Error Feedback and Model Update

After the establishment of virtual clock models at distributed node devices, both the original and the calibrated clocks outputs will be recorded, while those timestamps will be periodically transmitted to their edge devices in improving the accuracy of the modeling. Each device will transmit its clock information in the same format of the initial training phase, while all the conveyed information will be recorded at the edge device as

$$\check{I}_{ij}(t_k) = \begin{bmatrix} \mathbf{T_i}(\mathbf{t_k^l}) \\ \mathbf{C_i}(\mathbf{t_k^l}) \\ \hat{\mathbf{C}}_\mathbf{i}(\mathbf{t_k^l}) \\ \mathbf{C_{ij}}(\mathbf{t_k^l}) \end{bmatrix} \tag{4.16}$$

Based on the feedback information received, the edge device will further update the model in the same procedures introduced above, while a group of feedback-based estimation of the coefficients $\check{\boldsymbol{p}}$ can be obtained as

$$\check{\boldsymbol{p}}_{\boldsymbol{i}} = \boldsymbol{T}_{\boldsymbol{i}}^{-1}\check{\boldsymbol{\epsilon}}_{\boldsymbol{i}} \tag{4.17}$$

where $\check{\boldsymbol{\epsilon}}_{\boldsymbol{i}}$ is the error evolution of the calibrated clock extracted from the feedback information $\hat{\mathbf{C}}_\mathbf{i}$. Therefore, an updated estimation of the coefficients in terms of temperature-sensitivity

factor $\check{\eta}_i$, ideal operating temperature $\check{T}_{i0}$, and the initial clock skew $\check{\alpha}_{i0}$ can be extracted with enhanced precision.

Since the edge device is aware of the two groups of coefficients obtained from both the training information and its corresponding feedback, a difference between these two groups of coefficients can be observed at the edge device for each training iteration. The estimation bias of the clock skew at the v^{th} training iteration can be calculated as

$$\check{\boldsymbol{\gamma}}_i(v) = \frac{\check{\boldsymbol{p}}_i(v) - \hat{\boldsymbol{p}}_i(v)}{\hat{\boldsymbol{p}}_i(v)} \tag{4.18}$$

which can be evaluated as evidence of the modeling credibility. The edge device will conduct continuous training processes to obtain more accurate coefficients until it reaches the upper bound of the training iterations, or the observed difference is smaller than an ideal threshold, indicating that the training results reach its convergence.

4.4.4 Decision Making and Clock Accuracy Guarantee

After the accomplishment of the training process, each edge device will transmit the finalized parameters ($\check{\eta}_i$, $\check{T}_{i0}$, $\check{\alpha}_{i0}$) to both the cloud center and node devices for different purposes. On one hand, an updated digital twin counterpart of its physical skew will be established at each node device according to the delivered parameters, shown as

$$\check{\alpha}_i(t) = 1 + \check{\eta}_i(T_i(t) - \check{T}_{i0})^2 + \check{\alpha}_{i0} \tag{4.19}$$

which will be used to achieve the situation-aware calibration of its local clock skew under different operating temperatures. A virtual clock model will be established at each node to generate time information which is calibrated by the estimated clock skew. For the ideal situation, the estimation of clock characteristics will be identical to its real behavior, indicating that the skew of each virtual clock is eliminated while the device will not require any synchronization action during its further operation. However, the modeling inaccuracy incurred by the network uncertainty will lead to defective correction at the distributed devices, so that further clock modification and model updates are necessary.

During the network operation stage, each node device will occasionally deliver its timestamps associated with other application-related messages to the edge device, which will be responsible for observing the difference between its local clock outputs and the conveyed timestamps. In the case that the clock difference is foreseeable to be larger than the accuracy requirement at one device, its edge device will be managed to eliminate the clock error by triggering the OCP at this subordinate. Since the OCP consumes critical communication and computation resources during network operation, a minimal triggering frequency of the OCP is desired.

On the other hand, a long-range analysis will be continuously executed at the cloud center based on the historical models trained at each edge device and the recently delivered timestamps from the distributed nodes. Another comprehensive training process illustrated in Fig. 4.4 will be conducted for a node only if its previous model stored in the cloud center presents a significantly different behavior, which violates its newly generated timestamps.

4.5 Performance Evaluation

In this section, a series of simulations are carried out to evaluate the clock accuracy improvement and the network overhead incurred by adopting the proposed digital-twin-enabled clock synchronization (DTCS) scheme. The modeling of an industrial environment and the distribution of local devices are introduced first.

4.5.1 Temperature Modeling

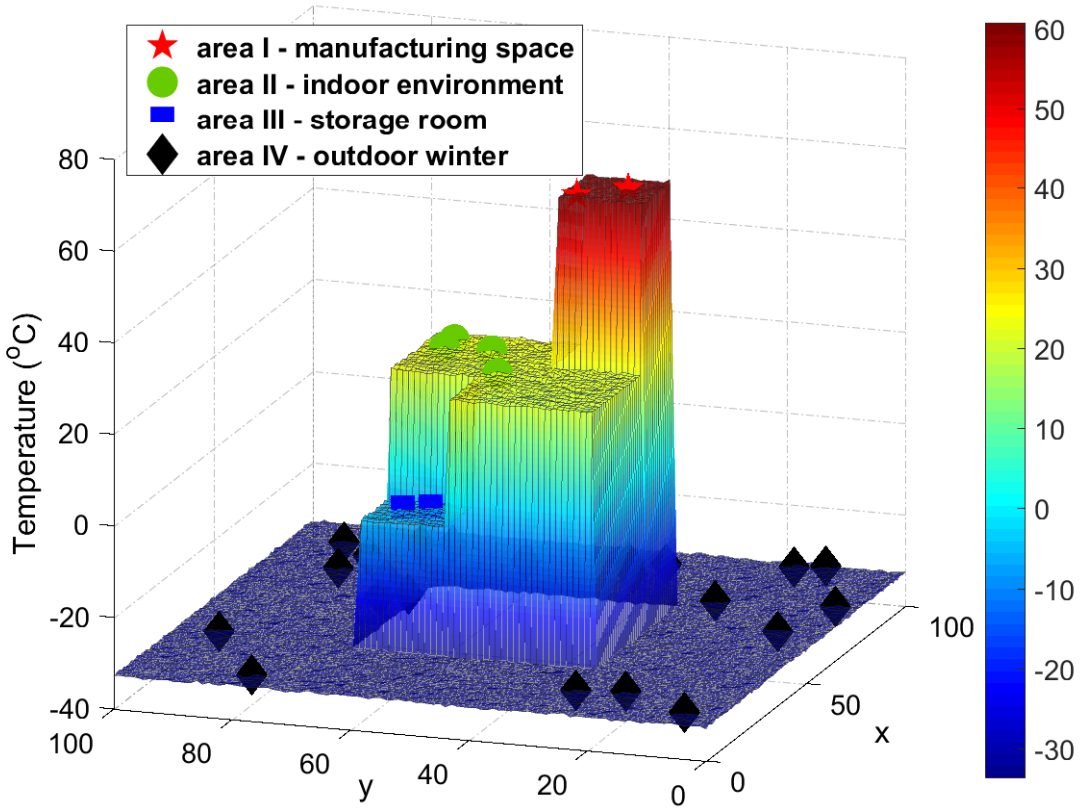
In this simulation, a common industrial environment with different areas associated with diverse temperature variations is simulated in MATLAB. The outdoor temperature information is collected from real environment while the indoor data are generated by simulation. As shown in Fig. 4.5a, 30 heterogeneous devices equipped with an oscillator-driven clock are randomly deployed in an industrial environment consists of 4 different temperature distributions, including the extreme and unpredictable outdoor environment where the temperature can be as low as -40^oC, well-controlled storage room with a constant temperature of -5^oC after its initialization, normal indoor areas with around 23^oC during daytime and around 16^oC during the

night, as well as manufacturing space, which can reach 60 ^oC during its operation. The oscillation frequency of each clock will vary with a different extent in its environment, leading to nonidentical clock errors after a certain time.

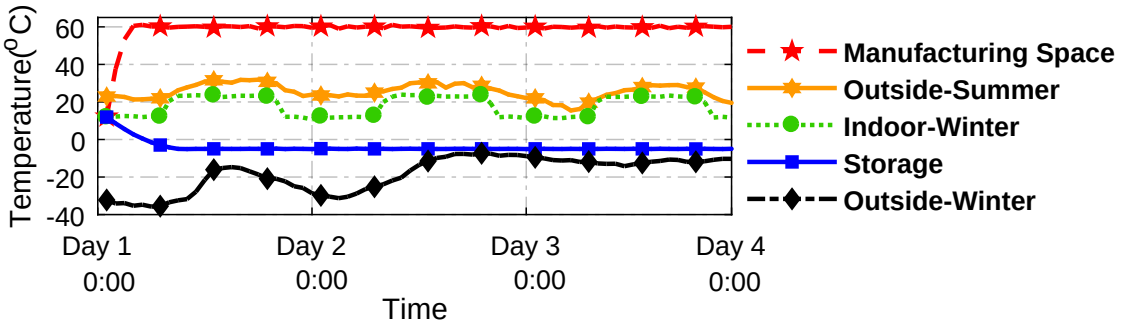
Meanwhile, the temperature in different areas will vary with time in its unique rate, as shown in Fig. 4.5b. For instance, the fluctuation of outdoor temperature during both summer and winter periods will be much more nondeterministic, while temperatures in manufacturing and storage space will keep almost unchanged after their initialization. Due to the heterogeneous susceptibility of the distributed devices to the external environment, each clock will present a distinctive error, especially after long-term operation. Therefore, without an ideal approach to compensate for the effect of temperature, the overall coherence of the IIoT system would be suspicious.

4.5.2 Modeling Accuracy Evaluation

To deal with the effect of the temperature and enhance the understanding of the distributed clocks over the network, a unique skew model comprises three coefficients, namely, temperature sensitivity, ideal operating temperature, and initial clock skew, is created for each clock according to (4.14) and (4.19) for initial training and successive feedback, respectively. Due to the existence of network uncertainty, PDV incurred will inevitably cause observation error during parameter estimation. Furthermore, a larger PDV will naturally result in increasingly unpredictable variation between every two successive receptions of the timestamp, leading to more apparent randomness of the estimation error during both the training and calculation processes. According to the deployed network and different service contents provided by the IIoT system, dynamic PDVs fluctuating to a large extent will arise, which are summarized in Table 4.1. In strictly-controlled networks, e.g., guaranteed service transport [100], the PDV can be eliminated by preserving dedicated communication resources for critical tasks. A small PDV is emerged in voice over LTE (VoLTE) supported by cellular networks with a PDV around $5\times10^{-5}s$ [101], while video-related services including video over LTE and video conferencing services can lead to a larger PDV up to $2\times10^{-2}s$ [102]. Furthermore, as illustrated in [101], the amount of PDV can be as high as $0.1s$ in some machine-type communication (MTC) scenarios.



(a) The spatial distribution of the temperature in an industrial environment, where multiple facilities are randomly deployed.



(b) The temporal evolution of the temperature in different areas.

Figure 4.5: Spatial and temporal distribution of the temperature in a representative IIoT environment deploying heterogenous devices.

Table 4.1: A summary for the values of PDVs during clock synchronization used in this simulation.

Operating networks	Packet delay variations (PDV)
Prioritized communication	0
Voice over LTE	$5 \times 10^{-5} s$
Video over LTE	$2 \times 10^{-2} s$
Machine-type communication	$0.1 s$

Based on the above observation, simulations were conducted to validate the estimation accuracy under various PDV conditions ranging from 10^{-5} to 10^{-1}, which can reveal most of the existing networks in practice. The estimation accuracy for the v^{th} iteration is calculated by comparing the estimated clock coefficients with the ground truth value $\boldsymbol{p}_i$, given by

$$\boldsymbol{\gamma}_i(v) = \frac{\check{\boldsymbol{p}}_i(v) - \boldsymbol{p}_i}{\boldsymbol{p}_i} \tag{4.20}$$

Taking the estimation of the ideal operating temperature T_0 as a representative, which is shown in Fig. 4.6, an accurate estimation is attainable when a well-controlled network ($\delta \leq 10^{-3}$) is employed. With the increment of PDV, the accuracy of the initial modeling and first-round feedback for T_0 are both severely degraded. A larger network uncertainty will even lead to over 100% initial estimation error, meaning that this estimation is not trustworthy. However, it can be observed from the enlarged linear inset that the feedback-based estimation can reduce the modeling inaccuracy from 35% to 15%, which will boost the precision of the model established. Similar observations are also found for the other two coefficients, while the feedback-based observations always increase the modeling accuracy significantly.

However, it can be observed that the initial training results still lack precision for the highly accurate clock synchronization purpose. Therefore, succeeding model training based on the continuously collected clock information is conducted to further decrease the error incurred during the model establishment. As shown in Fig. 4.7, an increasing training iteration will result in improved estimation accuracy of the three coefficients to different extents with the

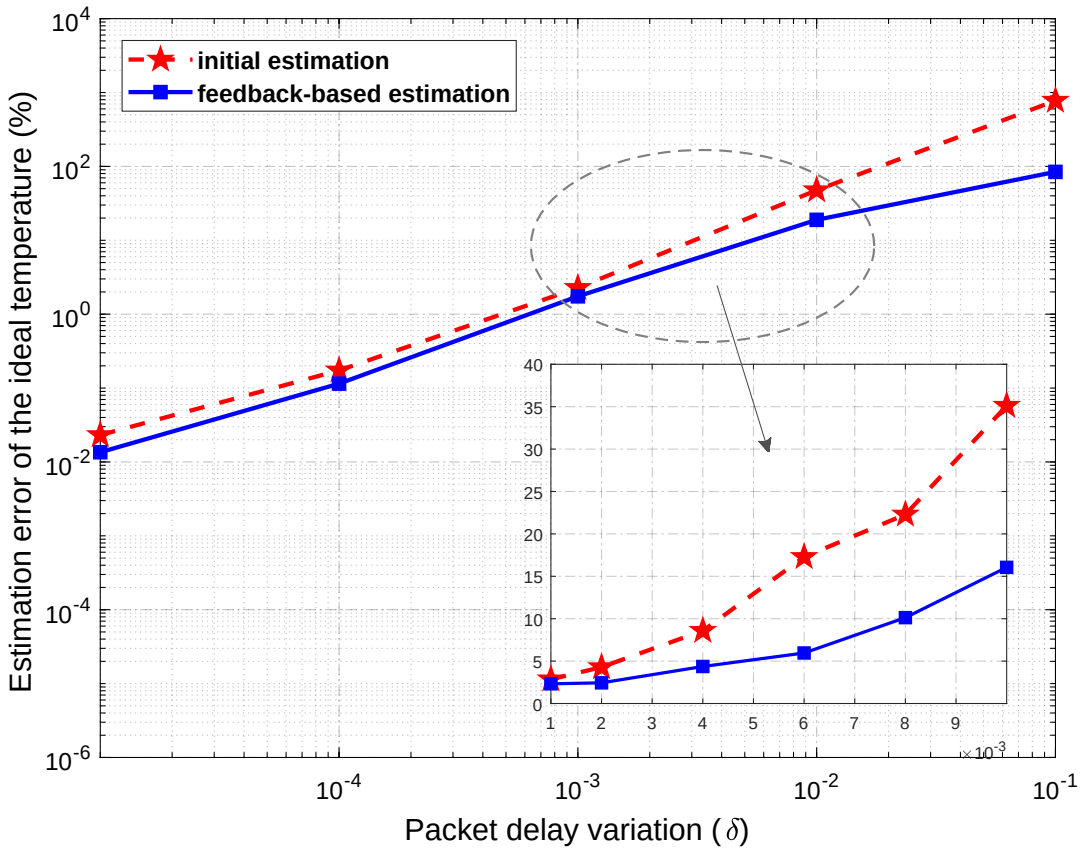


Figure 4.6: The initial training and initial feedback of clock skew estimation are severely affected by the fluctuation of the network. The feedback-based estimation is more reliable with the increment of the network uncertainty.

network uncertainty of 10^{-2}. The averaged estimation error will drop from over 100% to under 15% within only 5 iterations. Moreover, the accuracy of the estimation on initial skew is improved the most dramatically compared to the other two parameters, due to its extreme estimation imprecision during initial training. It can be concluded that the feedback-based estimation can converge to an accurate result with an error of 7% after 10 iterations in imperfect networks, while a much higher accuracy is obtainable more efficiently under better network conditions.

4.5.3 Clock Accuracy Improvement

As each clock under different environments varies with time at a distinctive rate, clock synchronization algorithms for the calibration of clock skews should be periodically adopted to ensure

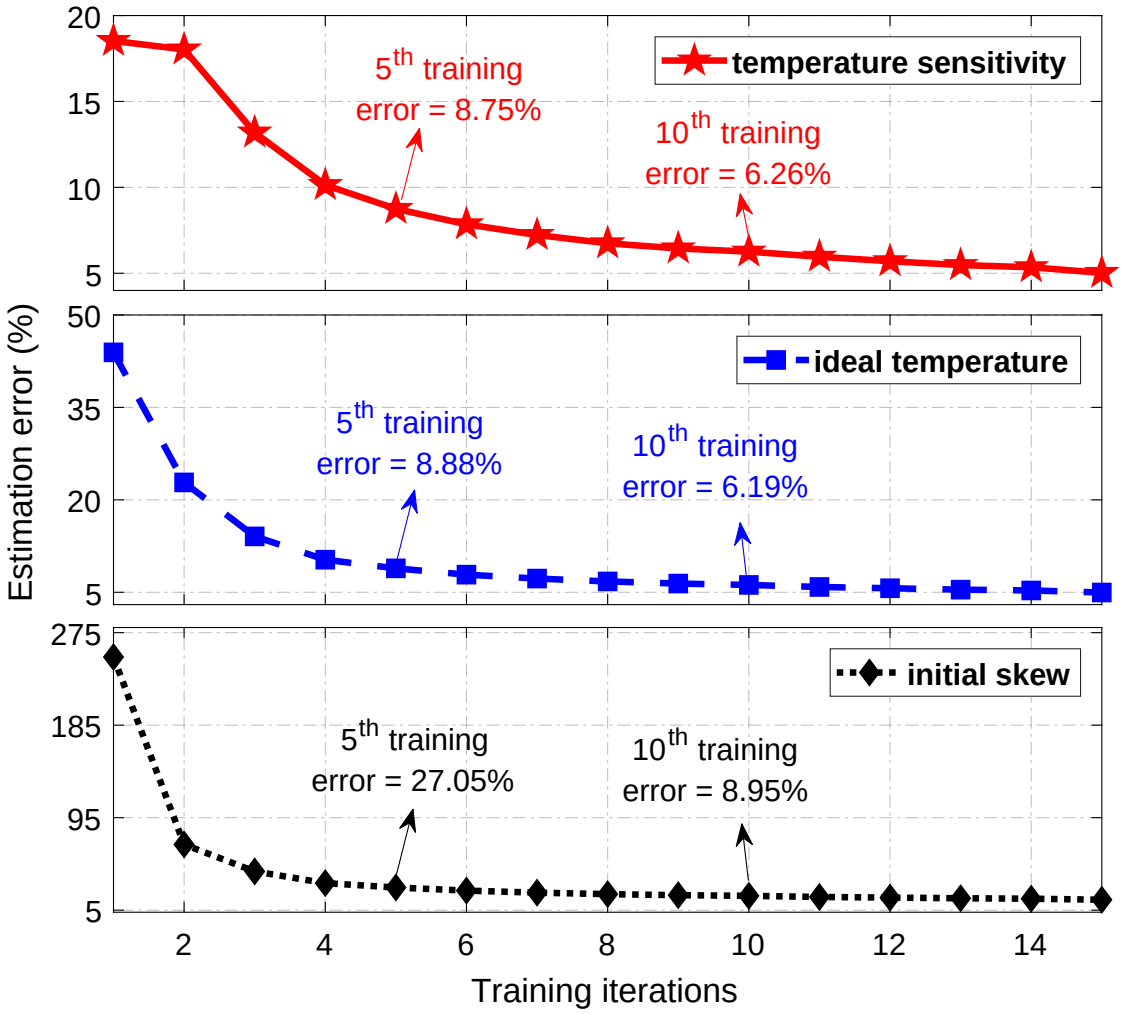


Figure 4.7: The feedback-based coefficients estimation inaccuracy will decrease with the increment of training iterations, while an estimation error within 10% is achievable after the 10^{th} iteration with a network that $\delta = 1 \times 10^{-2}$.

the overall clock accuracy within the network to be under the application-oriented requirements. Traditionally, synchronization methods adopting one-way or round-trip packet delivery are typically used in IoT systems, e.g., DCSIC [99] and MLE-based synchronization [103], where two successive messages containing timestamps will be exchanged between the reference node i and each local node j to calculate the clock skew for further correction. The corresponding estimation for the clock skew is based on four consecutive timestamps, which can be described as

$$\hat{\alpha}_e = \frac{C_j(t_2 + \delta_{ij}) - C_j(t_1)}{C_i(t_2 + \delta_{ij}) - C_i(t_1)} \tag{4.21}$$

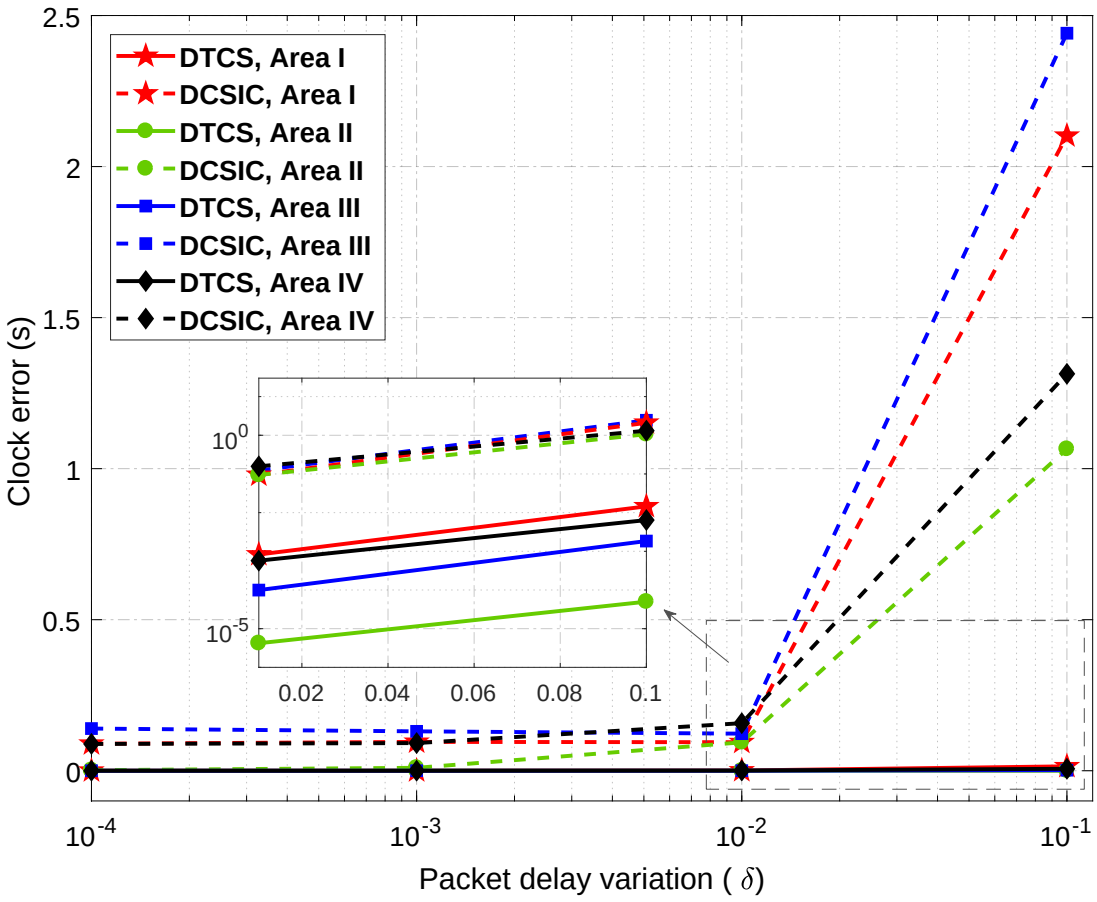


Figure 4.8: The cumulated clock errors of one hour in each environment will vary in different rates for both the proposed digital-twin-enabled method and the traditional calculation-based method.

where δ_{ij} is the random and variant PDV between the two nodes during different sequences of packet transmission. Moreover, another skew estimation method referred as TACSC [55] was proposed in literature based on the experimental results, without considering the packet delay variations.

Both the traditional methods (DCSIC and TACSC) and the proposed DTCS approach are sensitive to the network uncertainty δ, which has an everlasting effect on the performance of both the three schemes originally from the model initialization phase. Consequently, a simulation is conducted in 4 different operating areas to compare the performance of the DTCS scheme and the traditional methods under various network situations. As shown in Fig. 4.8, in a network of ideal conditions, the clock errors cumulated within 2 hours are extremely tiny for all

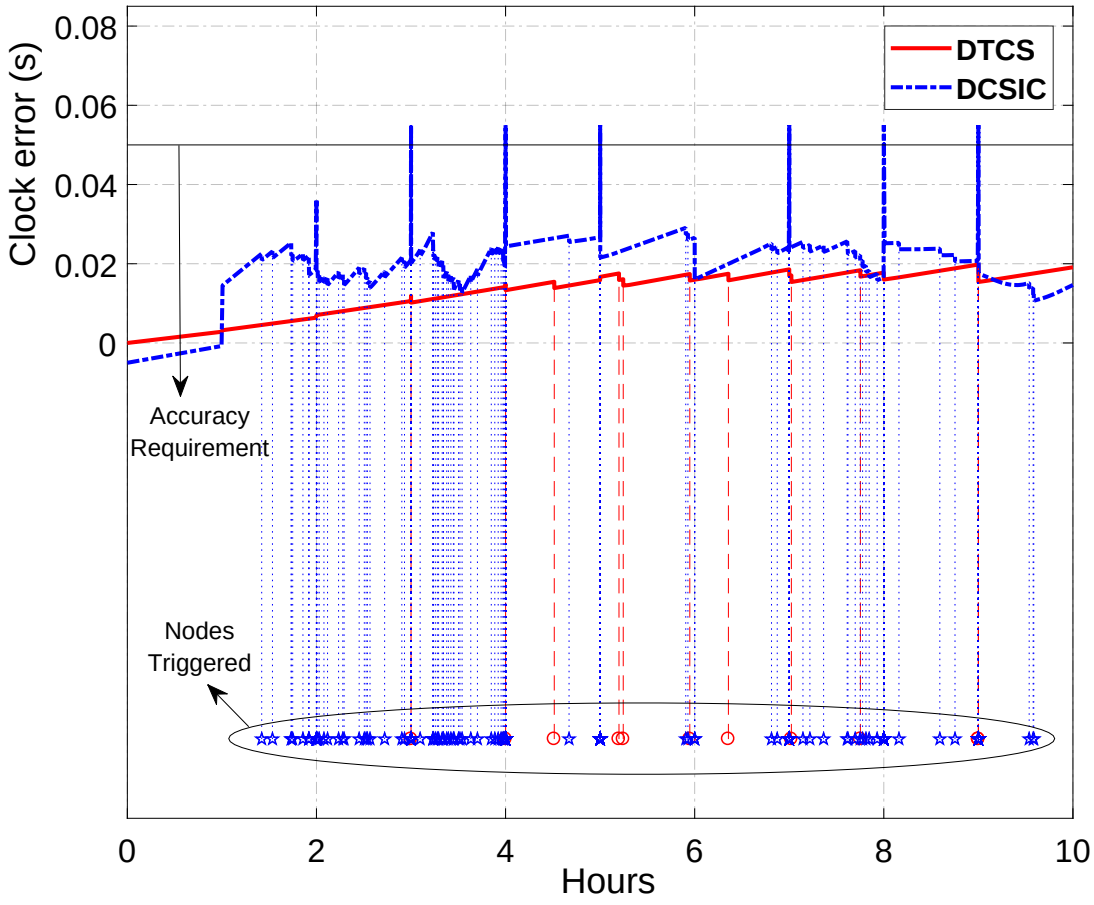


Figure 4.9: The proposed synchronization scheme realizes a higher synchronization accuracy while requiring fewer actions compared to the DCSIC method to ensure the accuracy requirement for 10 hours of long-term analysis.

the 4 cases, while the clock errors in the proposed scheme are always slighter than the DCSIC scheme. Meanwhile, with the increment of PDV, the clock errors increase exponentially for the nodes adopting DCSIC. Specifically, when the network uncertainty exceeds 1×10^{-1}, the timestamps inaccuracy incurred during clock synchronization will affect the traditional skew calculation method significantly, however, the proposed DTCS approach is still able to obtain an accurate estimation with a much slighter increment of the clock error under worse network conditions, which is shown in the enlarged inset with linear PDV and logarithmic clock error value.

Furthermore, since the proposed DTCS approach is proposed to achieve a better understanding of the environmental effect on the clock oscillators, another long-term observation of 10 hours is conducted to demonstrate the robustness of the proposed scheme under different

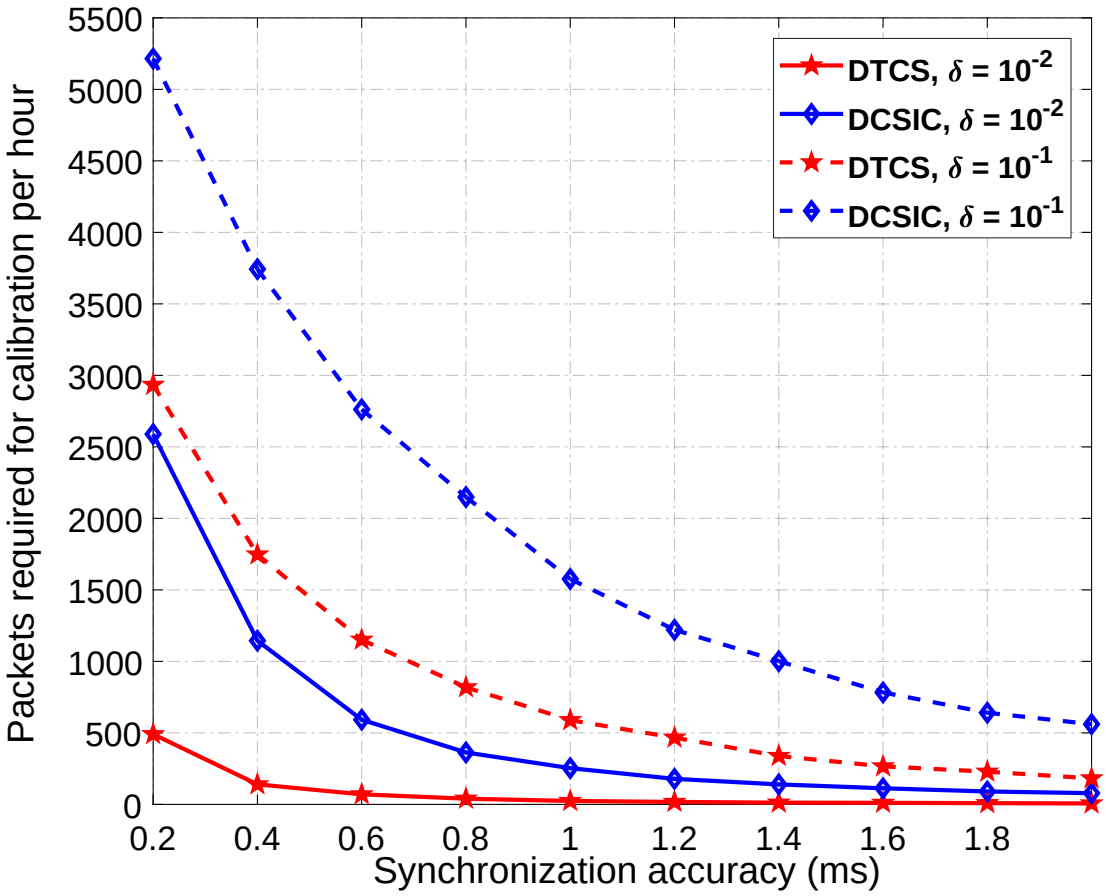


Figure 4.10: The total number of packets required during clock synchronization for the proposed digital-twin-enabled scheme is always smaller than the DCSIC scheme, especially when the accuracy requirement is stringent.

operating temperatures. In this case, a synchronization requirement is set to be $50ms$ so that any node violates such a threshold will trigger an OCP to eliminate its offset. It can be observed from Fig. 4.9 that, in the proposed scheme, the clock error increases much slower and smoother compared to the DCSIC scheme. In the meantime, there are 237 times that one of the devices was triggered in the traditional approach to guarantee the accuracy requirement, which is much higher than the proposed scheme of 11 times. Therefore, both the synchronization accuracy and the required actions in the DTCS scheme are dramatically improved.

4.5.4 Network Resource Consumption

Besides the achievable clock accuracy empowered by different clock synchronization methods, a critical criteria to validate the synchronization algorithm is the network resource consumed during the network operation stage. In traditional synchronization methods, e.g., the DCSIC, both the skew correction and offset compensation are executed occasionally to meet different accuracy requirements. By contrast, in the proposed digital-twin-enabled skew estimation, due to the reason that the superior device has a better understanding of the clock skew at its subordinate devices, the network resource used for skew correction can be saved. Furthermore, since the clock skew estimated through the proposed algorithm is more accurate and environment-aware compared to the traditional approaches, the interval between two OCPs is significantly increased. As shown in Fig. 4.10, with a tighter synchronization requirement on the clock accuracy, an increasing number of packets is required for both the two scenarios. However, the total resource consumed by the digital-twin-enabled method is always less than the DCSIC, especially when the underlying network condition is poor, indicating that the proposed scheme outperforms traditional methods under diverse operating environments.

Moreover, the packets requirement for different operating environments is also evaluated by comparing the proposed scheme with the other two methods, one of which (TACSC) has a better understanding on the effect of the operating environment. As demonstrated in Fig. 4.11, the packets used for clock calibration in the digital-twin-enabled synchronization is always fewer than the other two approaches under different operating environments. Furthermore, the requirement for the proposed scheme is set to be 50 times of that in the traditional scheme under a network with PDV equals 1×10^{-5}. Therefore, the overall packets used for clock calibration can be significantly reserved for other more critical applications.

Finally, it is worth noting that, the computation and communication resources of the proposed scheme is mainly consumed during the model training phase, which is achieved before the network operation. After the thoroughly collecting and processing the critical parameters of the digital-twin model for each clock, the proposed method can save resources for clock calibration process significantly, and the tolerance against network uncertainties will be much stronger than traditional packet-based synchronization methods.

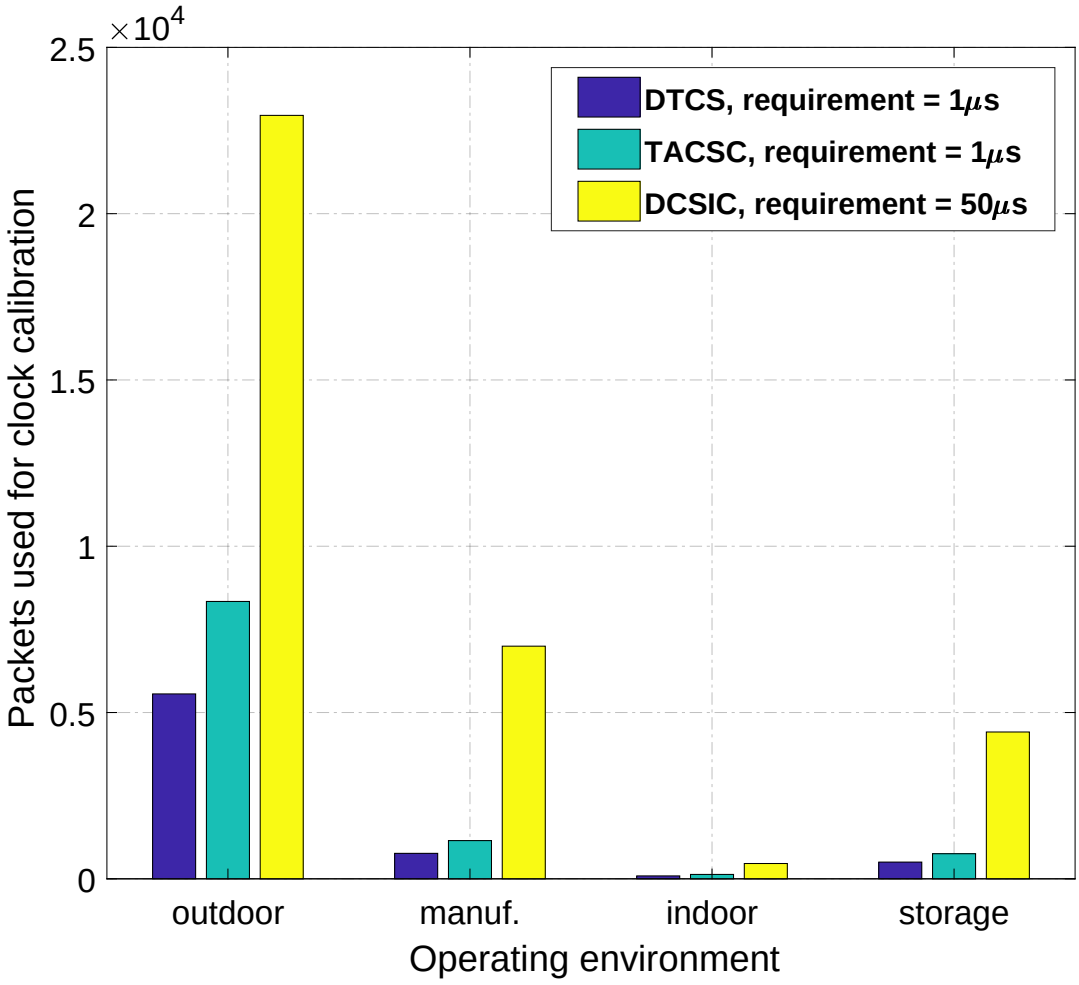


Figure 4.11: The total number of packets used for clock calibration under different operating environments. Even with a tighter accuracy requirement, the proposed scheme still always requires fewer packets for calibration.

4.6 Chapter Summary

A digital-twin-enabled intelligent clock skew estimation and distributed synchronization approach had been proposed in this chapter to compensate for the effect of complicated environments on the heterogeneous oscillators in industrial IoT systems. A comprehensive digital twin model of each clock had been established by modeling the effect of external operating environments on the clock drift. By adopting the established virtual model at each device, the distributed local clock skews were compensated adaptively. Compared to the traditional clock synchronization approaches where clock skews are calculated frequently during network

operation, the proposed method can avoid unnecessary and excessive packet exchange benefiting from a better understanding of clock behavior under different operating environments. The proposed digital-twin-enabled synchronization approach accomplished much higher clock accuracy with fewer packets required during network operation. Meanwhile, the performance improvements were even more significant under challenging network conditions, indicating that the proposed method is less sensitive to the packet delay variation and the dynamic operating environment for maintaining higher clock accuracy.

Chapter 5

Passive Network Synchronization based on Concurrent Observations

Accurate network synchronization is crucial to orchestrate distributed infrastructures in industrial Internet of things (IIoT) systems for accomplishing network-wide tight temporal collaboration. Traditional clock synchronization schemes are achieved with extensive exchanges of explicit timestamps for estimating clock offsets, which becomes impractical due to high overhead with the expansion of the network scale. Performance of conventional synchronization will also be dramatically deteriorated due to many uncertainties in IIoT networks, including unguaranteed communication resources, dynamic network conditions, and unpredictable behaviors of IIoT devices. In this chapter, we propose a passive network synchronization scheme based on concurrent passive observations to calibrate the distributed clocks in IIoT systems while significantly reducing the explicit interactions and network resource consumption during synchronization. By processing the physical phenomena observed concurrently by a group of selected IIoT devices, the local clock offsets of the passive observing devices can be efficiently estimated according to the common time reference linked to the event observed. Multiple relay nodes are further coordinated by the cloud center to disseminate the reference time information throughout the IIoT system in accomplishing global network synchronization. Simulation results demonstrate that by utilizing a series of concurrent observations with efficient coordination, the proposed scheme can achieve accurate and reliable network synchronization for large-scale IIoT systems with significantly reduced network overhead.

5.1 Introduction

As one of the indispensable enabling technologies for cohesive collaboration within distributed industrial systems, accurate clock synchronization is playing an increasingly important role in the design of industrial Internet of things (IIoT) systems to accommodate the stringent industrial requirements [68]. A broad variety of advanced industrial subsystems of large-scale IIoT networks, including wireless sensor and actuator networks (WSAN), intelligent transportation system (ITS), and advanced manufacturing system (AMS), are often designed to enhance the efficiency, productivity, and reliability of traditional industrial applications [2]. However, these subsystems exclusively hinge on the precise temporal alignment of the data packets exchanged among the involved IIoT devices throughout the entire IIoT network for achieving tight distributed collaboration. Consequently, the network-wide synchronization becomes one of the prerequisites in fulfilling the advanced applications enabled by IIoT systems in terms of ubiquitous sensing and distributed cooperation [22].

Conventional point-to-point clock synchronization schemes, most of which evolved from the Precision Time Protocol (PTP) [72], rely heavily on the frequent exchange of timestamps between the master node (i.e., the node providing reference time) and the slave nodes with inaccurate clocks that require frequent clock calibration [33]. Therefore, extensive interactions among a large number of industrial devices are considered indispensable in achieving accurate network synchronization, which poses daunting challenges in terms of network resource consumption for reliable timestamp provisioning throughout the large-scale IIoT systems. On the one hand, the excessive network overhead and accumulated latency are inevitably incurred by the frequent explicit interactions during concurrent clock alignment processes of network synchronization, which become intolerable for IIoT systems due to the associated overhead with the increase of the network scale. The high synchronization overhead and the delayed scheduling of critical data due to the lengthy synchronization procedures for real-time industrial applications will lead to significant deterioration of the overall productivity and potential risk of shutdown [88].

On the other hand, the underlying network dynamics and uncertainties among distributed IIoT devices will inevitably reduce the reliability of timestamps used during synchronization

processes. Due to the dynamic communication conditions in the IIoT networks, including varying Internet link quality [104], random accessing contentions, and dynamic routing processes during end-to-end packet transmission, the timeliness of the timestamps exchanged will become unguaranteed. Meanwhile, the susceptibility of IIoT devices to internal malfunctions and external attacks will lead to unreliable timestamps during clock calibration [45], especially for resource-constrained IIoT devices [105]. Consequently, designing efficient network synchronization protocols without posing strict requirements on the network conditions is particularly useful for IIoT systems.

Motivated by these challenges, we are interested in developing new efficient network synchronization mechanisms by eliminating the synchronization overhead due to explicit timestamps exchanges. Inspired by the fact that neighboring network nodes, which are physically close to each other, can simultaneously observe the highly correlated physical phenomena around them, the concurrent observations of the same event by neighboring devices could be useful in analyzing the temporal-correlation among a group of devices. Some typical applications of this include data fusion and reduction [106] [107], which are enabled by analyzing the temporal consistency of the different observations on the same target. Conversely, by processing concurrent observations of the same physical phenomenon from nearby nodes, the local clocks can be calibrated according to the obtained observation misalignment. As a result, the authors in [108] proposed a control message free synchronization scheme by utilizing the detected events in a wireless sensor network. The local detection time of the involved multiple sensors for the same event is regarded as identical so that their associated clocks can be corrected and synchronized accordingly. However, new protocol designs for global synchronization among distributed devices with different processing capabilities are still needed to support large-scale IIoT systems.

Therefore, to take advantage of the concurrent observations while avoiding the associated deficiencies, a *PAssive Network Synchronization based on concurrent Observation* (PANSO) scheme is proposed in this chapter, with the support of cloud-orchestrated reference time dissemination and PCA-assisted reliability enhancement. Specifically, by processing the concurrent observations of the same physical phenomena at a group of selected devices, a novel passive clock calibration mechanisms with minimized explicit interactions is designed to reduced

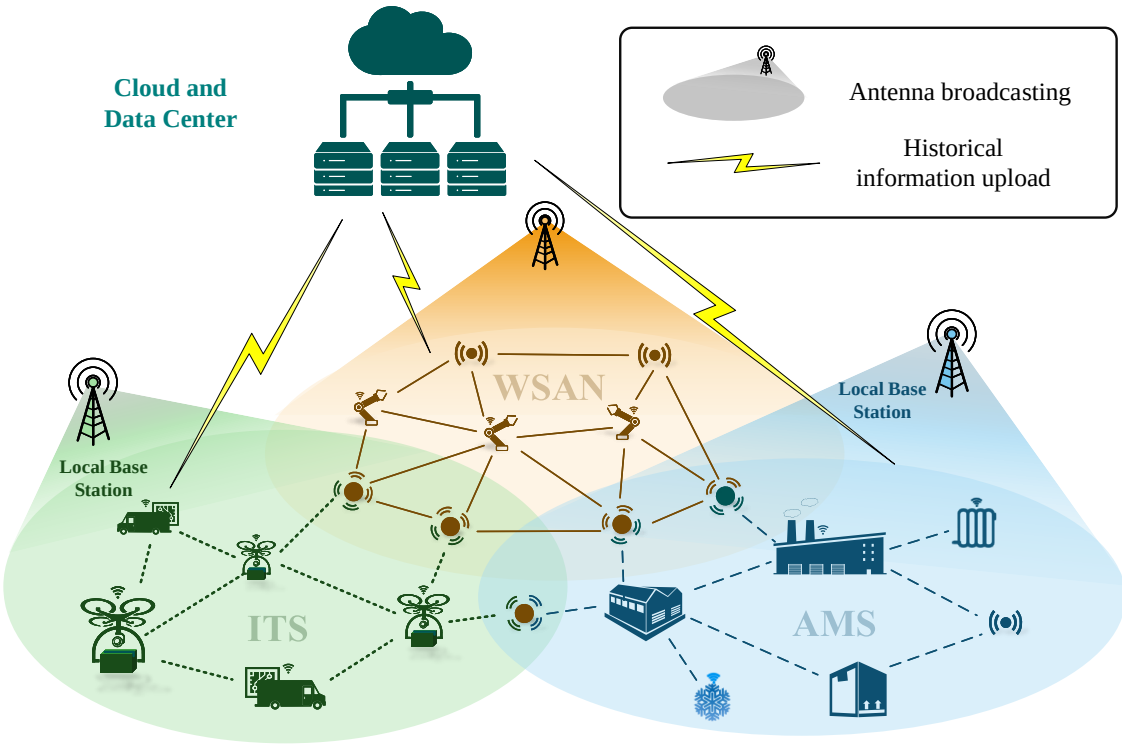


Figure 5.1: The IIoT system consists of multiple subsystems with heterogeneous devices and plants communicating via various protocols. Interactive nodes can assist to share critical information among subsystems.

network overhead during the network synchronization process. According to the number of concurrent observations, the distributed nodes in large-scale networks are further orchestrated into isolated nodes and interactive nodes by the cloud center, and reliable interactive nodes are selected as relays to expand the reference time throughout the network. A receive-signal-strength-based distance compensation mechanism is further designed to minimize the effect of observer bias due to the corresponding signal propagation latency. Additionally, principal component analysis (PCA) is adopted in the cloud center to ensure reliability during network synchronization by analyzing the historical observation instants uploaded from the local devices while filtering unreliable nodes. The network synchronization performance in terms of achievable accuracy and reliability is significantly enhanced as the expected improvements.

The remainder of this chapter is organized as follows. The concurrent observations in a typical IIoT system are introduced in Section 5.2. The proposed PANSO scheme is designed in Section 5.3 in detail, including the observation selection, observation processing, and passive

offset estimation. Some proper mechanisms are further designed in Section 5.4 to enhance the performance of PANSO for large-scale IIoT networks, including distance compensation and PCA-assisted reliability enhancement. Simulation results are carried out in Section 5.5 to demonstrate the effectiveness of the proposed scheme in terms of the achievable accuracy and reliable enhancement, followed by the conclusion in Section 5.6.

5.2 Concurrent Observations in IIoT systems

Concurrent observations of the same physical phenomenon at different neighboring IIoT devices are critical to provide the time reference to the group of observing devices, forming the cornerstone of the PANSO scheme. In a large-scale IIoT system, the available phenomena for concurrent observations are diverse, including the events during environment monitoring and the broadcasting of electromagnetic signals.

For most of the IIoT systems, a large number of sensor devices are typically deployed in achieving ubiquitous sensing and real-time monitoring of the target environment. Many external events could cause large variations of the environmental parameters monitored, which can be observed by the involved concurrent sensors. The observation instants can be recorded by different observers for further centralized correlation analysis in achieving network synchronization. However, the concurrent observation of the same environmental event requires common sensing capability, which could be hindered by the heterogeneity of the IIoT devices. Furthermore, the response time, processing capacity, limited sampling rate, and diverse locations of the observing sensors will inevitably lead to observation bias and recording inaccuracy, limiting the achievable synchronization accuracy.

By contrast, electromagnetic signals radiated from the nearby wireless transmitters are more accessible to IIoT devices for concurrent observations. Since neighboring network nodes can observe the signal from the same transmitter almost simultaneously at disparate locations, the receiving time at each IIoT device regarding the same signal can be recorded for further analyzing the observation difference. By sharing the temporal recorded signals with neighboring nodes, synchronization among a group of concurrent observers can be achieved. Moreover, proper coordination for selecting the communication channel and target signals is critical to

ensure observation accuracy and synchronization efficiency. Since electromagnetic signals are more convenient concurrent observations in an IIoT system, the rest of this chapter will choose the radio signals for a group of neighboring devices as the concurrent observation.

5.3 Passive Network Synchronization based on Concurrent Observation

The proposed PANSO scheme is achieved by processing the commonly observed physical phenomena for multiple groups of industrial devices in a large-scale IIoT system with the assistance of the cloud center, as demonstrated in Fig. 5.1. The overall process of PANSO consists of four successive phases, including concurrent observation selection, concurrent observation processing, reference time expansion, and synchronization performance enhancement.

5.3.1 Concurrent Observation Selection

As shown in Fig. 5.2, the concurrent observation selection (i.e., *Phase I*) is initiated by selecting a proper coordinator node (CN) by the cloud center. After selection, CN is responsible for two different tasks in guiding the following synchronization procedures. Initially, CN should define the communication channel and target transmitter that the neighboring devices will listen to, which lays the foundation of the concurrent observation. In the proposed PANSO scheme, target transmitters should be defined as the IIoT device with strong communication capability, which can frequently broadcast signals to their neighboring devices. Moreover, since the target transmitter in IIoT systems is frequently transmitting a large number of packets, CN will be required to define the target signal to enhance the observation efficiency. In practice, there are many different kinds of signals that are unique and obvious to be observed, e.g., the boosting signal of a transmitter always changes dramatically from silence to a strong electromagnetic signal wave, which can be clearly detected and recorded by nearby devices. Moreover, for communication protocols adopted in an IIoT system, the broadcasting signals are usually associated with the transmitting timestamps in its packets, which can be further used for validation to ensure observation accuracy.

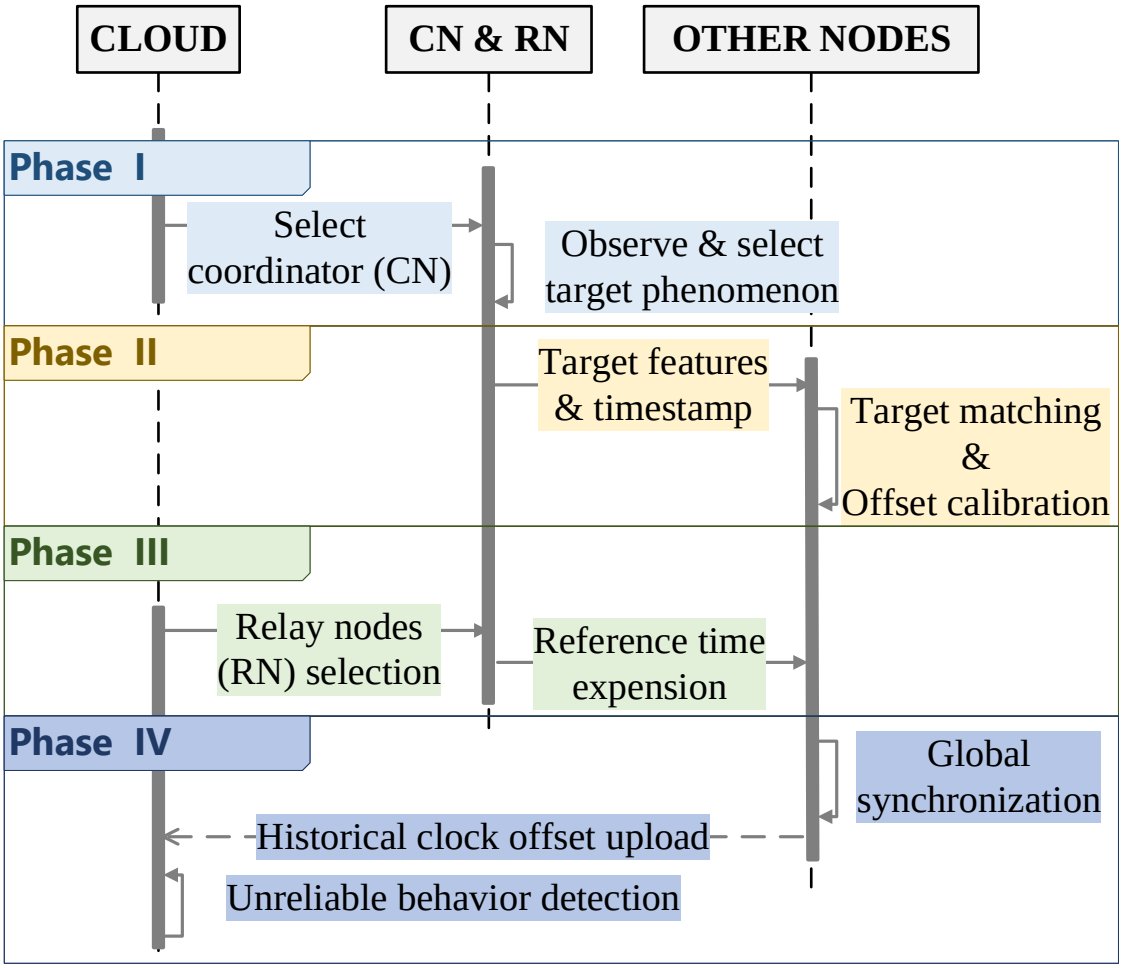


Figure 5.2: The proposed PANSO scheme consists of four successive phases aiming at different tasks, namely, observation selection, observation processing, reference time expansion, and synchronization performance enhancement.

After recording a sufficient number of signals transmitted from the target transmitter, CN will select one of the signals, s_1^{CN}, as the target signal, which is associated with a broadcast address in the MAC layer and obvious features. The transmitting time of s_1^{CN} from the target transmitter (i.e., t_1) enclosed in the corresponding PHY packet P_1 will be recorded for further verification during the successive signal matching process. Meanwhile, the local reception time $\hat{t}_1^{CN}$ and the receive signal strength (RSS) value RSS_1^{CN} associated with the signal s_1^{CN} will also be recorded. Thereby, CN will transmit the relevant information to all the other neighboring devices that are associated with the target transmitter in data field of its packet P_{CN}. An example that CN adopting IEEE 802.11ac during concurrent observation is shown in Fig. 5.3, where

the shared critical information I_{CN} in the data field of the packet P_{CN} is closely related to the subsequent processing and passive synchronization, given by

$$I_{CN} = (s_1^{CN}, t_1, \hat{t}_1^{CN}, RSS_1^{CN}) \tag{5.1}$$

5.3.2 Common Observation Matching

In *phase II*, once the information I_{CN} is received at the local device i, it will compare the target signal s_1^{CN} with its previously recorded local signals s_1^i that delivered from the same transmitter. To ensure the matching accuracy, it is necessary that both the features of the signals and the corresponding transmission timestamps should be identical. The matched signal is given by

$$s_1^i = \{s | s = s_1^{CN} \cap t_1^i = t_1, \forall s \in s^i\} \tag{5.2}$$

where t_1^i is the timestamp associated with the signal s^i transmitted from the first target transmitter, which is recorded at the device i. The verification condition $t_1^i = t_1$ can guarantee that the matched signal s_1^i is identical to the target signal defined by CN. Meanwhile, the local device i will also record the reception time $\hat{t}_1^i$ and the associated RSS value RSS_1^i of the received signal for the following processing procedures. Consequently, the local information recorded at the device i after successful signal matching regarding the target transmitter can be written as

$$I_1^i = (\hat{t}_1^i, \hat{t}_1^{CN}, RSS_1^i, RSS_1^{CN}) \tag{5.3}$$

Since a group of devices even close to each other will receive signals via dynamic communication environments, multipath propagation effect will pose noticeable influence on the target signal analysis. In this case, each local device might receive the same signal multiple times continuously with similar data information at slightly different time instants, leading to confusion and inaccuracy of the signal matching. To tackle this issue, the most straightforward and effective approach is to sort all matched signals based on the reception time at the local devices and select the earliest one while ignoring the rest. Although it cannot guarantee that the effect of multipath communication is eliminated, the observation bias and estimation error

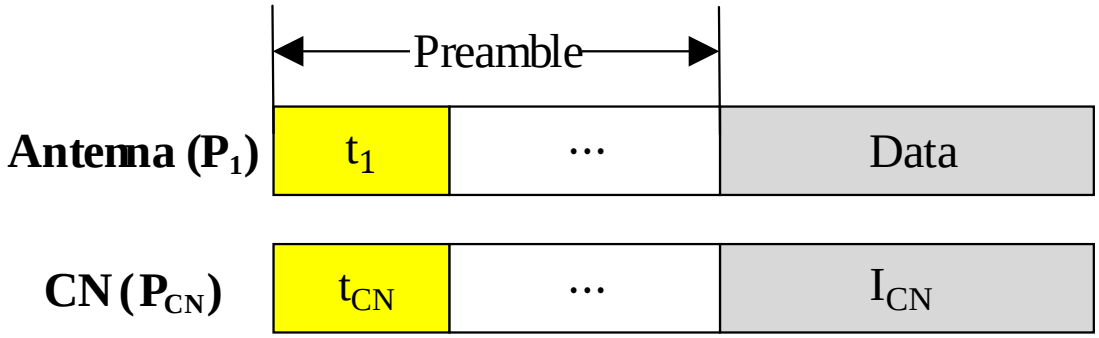


Figure 5.3: An example of the PHY packets of CN with regard to the signal transmitted from the target transmitter adopting IEEE 802.11ac protocol. The synchronization-related information is contained in the data field as I_{CN}.

induced is minimized. The final recorded matching time at each device i can be further written as

$$\hat{t}_1^i = min\{\hat{t}_1^i(k)\} \tag{5.4}$$

where k denotes a series of observation instants that the target signal is matched at the local device.

5.3.3 Passive Offset Estimation

Passive network synchronization of each device is achieved by eliminating the estimated clock offset passively according to the concurrent observation difference between its local information and CN. In achieving passive offset estimation, all devices only need to stay listening to the broadcasting signals transmitted from the surrounded devices. Only CN will be responsible for actively coordinating and transmitting temporal information as the references for other nodes.

A group of devices nearby the target transmitter can have slightly different observation instants on the same signal, shown as the group 1 in Fig. 5.4. Each device will be associated with a unique record due to its clock offset and the observation bias on the target signal defined by CN. The generation time at transmitter 1 and the local observation time at each device regarding the target signal is listed in Table 5.1, where the local observation instants are different, affecting by various issues including the distance between each device and transmitter 1. For a small-scale network, it is rational to assume that the same signal can be received at each device

almost simultaneously with a negligible error.

Table 5.1: The timestamps for signal generated by the first target transmitter recorded at distributed devices

Node	Transmitter 1	CN	N_1	N_2	...	N_m
Timestamp	t_1	$\hat{t}_1^{CN}$	$\hat{t}_1^{1.1}$	$\hat{t}_1^{1.2}$	...	$\hat{t}_1^{1.m}$

Inspired by the fact that the local clock offset will affect the observation instants, CN will transmit its observation time $\hat{t}_1^{CN}$ on the target signal to its neighboring devices. Based on the information, device i can compare its local observation timestamp t_1^i in discovering the clock offset, given by

$$o_i = \hat{t}_1^i - \hat{t}_1^{CN} \tag{5.5}$$

which will be used to calibrate its local clock so that passive synchronization can be achieved without explicit resource consumption at device i.

However, in the case that all nodes in one area are distantly distributed, the difference of the signal propagation time will lead to an obvious observation bias, meaning that each device even with an accurate clock will have its own observation time for the target signal. By simply assuming every device can observe the same signal simultaneously will cause synchronization error as an inevitable consequence. Affected by the difference of the propagation distance, the updated offset of device i compared to CN can be estimated as

$$\hat{o}_i = t_i^n - t_{CN}^n - \frac{d_{ci}}{c} \tag{5.6}$$

where c is the transmission speed of an electromagnetic signal, while d_{ci} is the relative difference between the distance of CN and device i with respect to their common target transmitter.

5.4 Global Synchronization and Performance Enhancement

With the expansion of the IIoT network scale, local passive synchronization is insufficient to guarantee the overall performance of the clock behavior due to the long propagation distance and limited communication range. As a consequence, proper mechanisms should be adopted

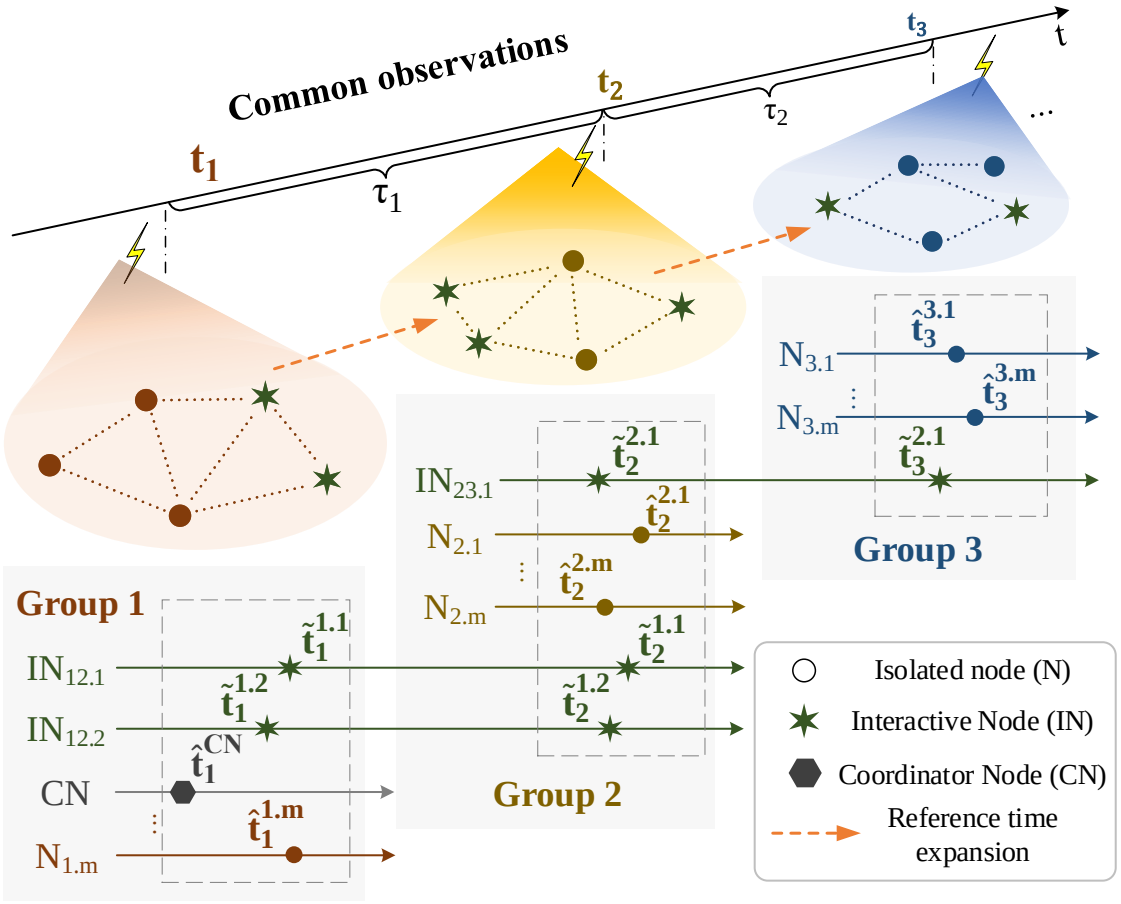


Figure 5.4: Time information for every device in a network corresponding to each concurrent observation. The interactive nodes are aware of more common observations compared to isolated nodes.

in enhancing the global synchronization performance.

5.4.1 Propagation Effect Compensation

Due to the distributed deployment of IIoT devices in large-scale industrial systems, observation bias will be inevitably induced, which will cause inaccuracy during offset estimation. Depending on the network scale, this inaccuracy can be tiny as microsecond with a distance of tens of meters, or a few milliseconds if the distance increases to kilometers. Therefore, distance compensation is necessary to enhance observation accuracy. The location-awareness for the IIoT devices is different due to their heterogeneity in terms of mobility, cost, and functionality. For instance, in the smart transportation subsystem, every node is typically equipped with a

location-aware unit, e.g., GPS, the distance between each device and their selected target transmitter can be accurately estimated. However, for devices without costly GPS units or deployed in indoor and underground environments, other methods for calculating the relative distance should be adopted. Since these devices are normally assigned to perform routine tasks with steady objectives, schedules, and places, it is reasonable to assume that their locations are fixed in most of their life cycles.

Since the target signals are always transmitted with the corresponding receive signal strength (RSS) values during information exchange, the RSS values can be used to compensate for the effect of distance [109] according to the fact that each device can calculate the approximate distance to the target transmitter based on the RSS value. However, due to the dynamic environment and inaccurate propagation modeling, the distance estimated based on the RSS values obtained will be less accurate, which is one of the main reason of the synchronization error after adopting the proposed PANSO scheme. To be more specific on the RSS-based relative distance estimation, CN will deliver the RSS value for the received signal from the target transmitter together with the time information to its subordinates, as shown in Fig. 5.5. By assuming the fixed devices follow the free space propagation model during their communication, the relation between the distance and the corresponding RSS value can be obtained, while the distance between the target transmitter and CN can be calculated by

$$R_c = A_0 - 10nlg(d_c) \tag{5.7}$$

and the distance for the device i can be similarly written as

$$R_i = A_0 - 10nlg(d_i) \tag{5.8}$$

where n is the path loss factor affected by the environment and A_0 is the transmission power from the target transmitter. d_c and d_i are the distances from the transmitter to CN and local devices, respectively. According to (5.7) and (5.8), the difference between the two distances can be obtained as

$$d_{ci} = d_c - d_i \tag{5.9}$$

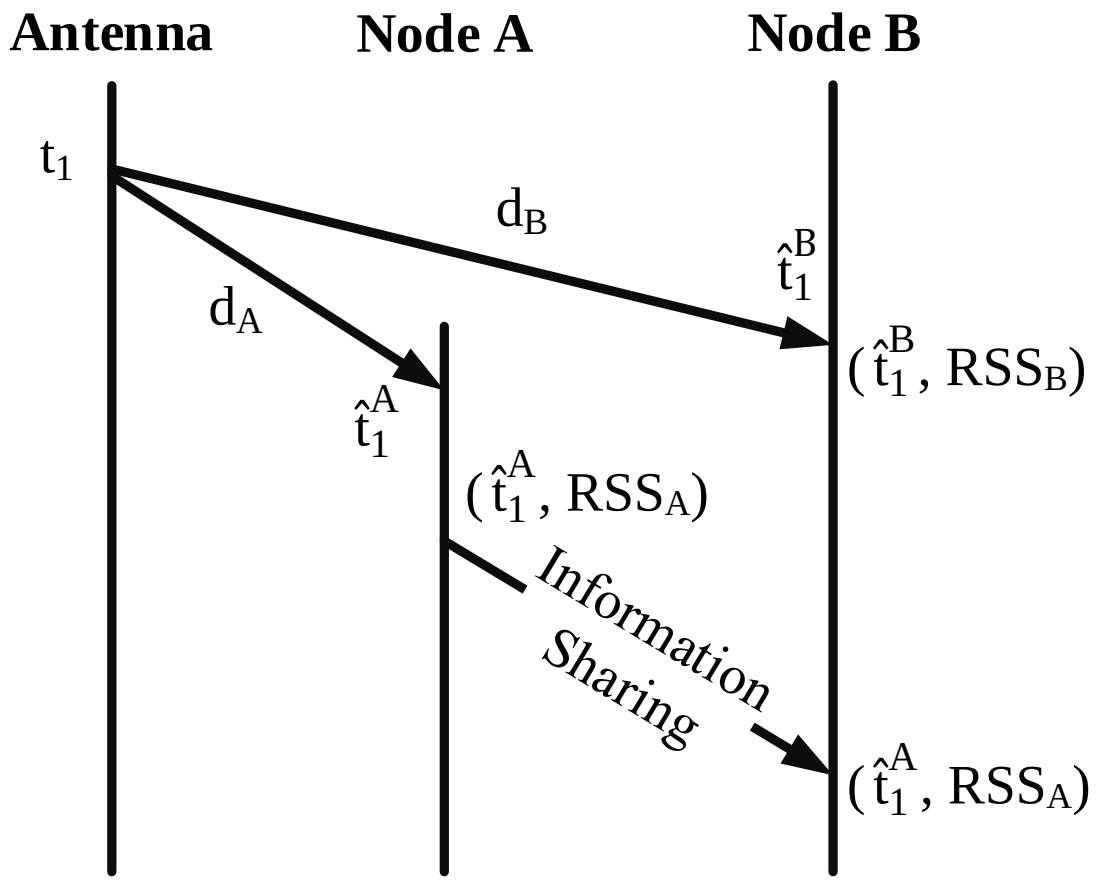


Figure 5.5: The critical information transmission in terms of observation instants and RSS values between the common target transmitter and two devices, where the difference of the distance will induce synchronization error.

so that the offset estimation can be further updated from (5.6).

5.4.2 Reference Time Expansion

By calibrating the estimated offset $\hat{o}_i$ for each local device, the nodes around CN can be synchronized accurately. However, for a large network with massive nodes which cannot receive the same signal from a common target transmitter, global synchronization cannot be achieved with only the coordinator node. To efficiently disseminate the reference time throughout the entire network without reducing the clock accuracy for the distantly distributed devices, multiple node are selected for reference clock expansion in *Phase III* of the proposed scheme. Since interactive nodes (IN) can receive signals from multiple transmitters in different areas, some of

them are assigned as relay nodes (RN) according to Algorithm 4.

To be more specific, one of the nodes with higher computation capability and accurate clock generations is selected as the coordinator, which will also be responsible for generating global time reference in accomplishing the proposed network synchronization. Since CN might only belong to a single subsystem, other devices are necessary to disseminate the reference clock information to achieve global synchronization for a large-scale network. Initially, all devices in the network will listen to the broadcasting signal from the surrounded transmitters and upload the received transmitter ID to the cloud center. If more than one IDs of the target transmitters are recorded at the cloud center for a single device, it will be marked as an IN, and one of the INs will be randomly selected as the relay node (RN) to expand the time reference if multiple INs exist between any two subsystems. Due to the randomness of the initial RN election, successive RN selection and filtering mechanisms are necessary to guarantee the accomplishment of global synchronization. Finally, all other devices will only need to passively synchronize to the received time reference without expending the limited network resources on explicit synchronization processes.

After the selection process, RNs will be responsible for defining succeeding target signals and deliver the associated temporal information to the neighboring areas. As shown in Fig. 5.4, the selected RNs are aware of the information from multiple target transmitters, while isolated nodes can only sense partial information due to lack of connectivity. Consequently, RNs can expand the reference time gradually in achieving global synchronization. For the isolated node i within the j^{th} group, its local clock offset can be calculated as

$$o_i = \hat{t}_j^{j,i} - \tilde{t}_j^{j-1} = \hat{t}_j^{j,i} - \tilde{t}_1^1 - \sum_{k=1}^{j-1}(o_k^{RN} + \tau_k) \tag{5.10}$$

where o_k^{RN} is the local clock offset for each RN compared to the clock reference and τ_k is the interval between each two common signal observations, which can be accurately calculated by each RN.

Algorithm 4 Cloud-Assisted Initial Node Arrangement

Cloud select one node with high-quality clock as the coordinator node (CN)
for Every target transmitter m in its subsystem **do**
    Broadcast its identification ID_m to surrounding devices
    for Each of its belonging devices **do**
        Record ID_m and upload to cloud center
    end for
end for
for Each device in the system i **do**
    Cloud center calculate its uploading times k_i
    if $k_i > 1$ **then**
        Mark it as an interactive node (IN)
        Record uploaded transmitters as neighboring transmitters
    else
        Mark it as an isolated node
    end if
end for
for Each neighboring transmitter **do**
    if Only one IN exists **then**
        Mark it as the relaying node (RN)
    else
        Randomly select one IN as RN
    end if
end for

5.4.3 PCA-Assisted Reliability Enhancement

Another important issue in *Phase IV* is the robust synchronization provisioning under various security and reliability issues during synchronization in hostile industrial environments. Lacking security-related mechanisms can lead to untrustworthy clock information expansion and potential inconsistent cooperation among the connected devices. Since principal component analysis (PCA) is effective in investigating abnormal behaviors, in this section, a PCA-assisted scheme is designed to enhance the synchronization reliability among the involved devices. Moreover, due to the fact that the performance of PCA algorithms are affected by the network issues severely, it will be beneficial to consider a series of historical concurrent observations, which is not significantly affected by the network conditions, as the core of the successive analysis.

Principal component analysis (PCA) is a commonly used mathematical tool to analyze the collected massive data and establish the corresponding predictive model [110] [111]. Consequently, PCA is powerful in detecting anomalous data and unreliable behaviors in applications with a large amount of temporal correlated data. For the synchronization-related timestamps, as the cloud center will perform the reliability enhancement scheme, it will collect the observation timestamp at the k^{th} physical phenomenon from each node i as $\hat{t}_k^i$ and its corresponding relay timestamp as $\tilde{t}_k^1$, so that the relative clock offset can be calculated as

$$o_k^i = \hat{t}_k^i - \tilde{t}_k^1 \tag{5.11}$$

while the relative skew between node i and its relay node can be estimated by

$$\hat{\alpha}_k^i = \frac{\hat{t}_k^i - \hat{t}_{k-1}^i}{\tilde{t}_k^1 - \tilde{t}_{k-1}^1} \tag{5.12}$$

Due to the uniqueness of each clock skew and the effect of external operating temperature on the clock oscillation frequency as validated in [12], the estimated clock skew can be used as a critical evidence for detect abnormality of the clocks. Together with the local temperature for the node denoted by T_k^i, a $4 \times k$ matrix can be formulated as X_i for the node i with 4 physical observations and k samples, i.e.,

$$\boldsymbol{X_i} = \begin{bmatrix} \hat{\boldsymbol{t}}^{\boldsymbol{i}}_{\boldsymbol{k}} \\ \boldsymbol{o}^{\boldsymbol{i}}_{\boldsymbol{k}} \\ \hat{\boldsymbol{\alpha}}^{\boldsymbol{i}}_{\boldsymbol{k}} \\ \boldsymbol{T}^{\boldsymbol{i}}_{\boldsymbol{k}} \end{bmatrix} = \begin{bmatrix} \hat{t}^i_1 & \hat{t}^i_2 & \dots & \hat{t}^i_k \\ o^i_1 & o^i_2 & \dots & o^i_k \\ \hat{\alpha}^i_1 & \hat{\alpha}^i_2 & \dots & \hat{\alpha}^i_k \\ T^i_1 & T^i_2 & \dots & T^i_k \end{bmatrix} \tag{5.13}$$

Thereby, the raw data matrix X_i should be transformed into a new coordinate subsystem by

$$\boldsymbol{W_i} = \boldsymbol{Q_i X_i} \tag{5.14}$$

where Q_i is the transformation matrix and W_i is the corresponding projection referred as score matrix. Since the time information and temperature information have different scales, data normalization should be adopted to mitigate the effect. Therefore, a zero-mean and unit-variance matrix $\hat{X}_i$ for each node is obtained, which can be used to derive the covariance matrix by

$$\boldsymbol{C}_{X_i} = \frac{1}{n-1} \hat{\boldsymbol{X}}_{\boldsymbol{i}}^{\,*} \hat{\boldsymbol{X}}_{\boldsymbol{i}} \tag{5.15}$$

where * is the conjugate transpose operator.

According to the calculated covariance matrix, its eigenvalue and eigenvector can be obtained by

$$\boldsymbol{V_i} = \boldsymbol{U_i}^{-1} \boldsymbol{C}_{X_i} \boldsymbol{U_i} \tag{5.16}$$

where V_i is the diagonal matrix with eigenvalues and U_i is the matrix of the eigenvectors of C_{X_i}, respectively. By selecting the a few largest columns in U_i as the principal components, the original eigenvalue matrix V_i and eigenvector matrix U_i can be reduced to $\hat{V}_i$ and $\hat{U}_i$, with the principal components, i.e., the most dominant parts. By selecting the principal parts of the eigenvector matrix as the transformation matrix Q_i, the projection of the raw data can be described as

$$\hat{\boldsymbol{W}}_{\boldsymbol{i}} = \boldsymbol{Q_i} \hat{\boldsymbol{X}}_{\boldsymbol{i}} = \hat{\boldsymbol{U}}_{\boldsymbol{i}}^{\,T} \hat{\boldsymbol{U}}_{\boldsymbol{i}} \tag{5.17}$$

Consequently, the residual value, namely, the remaining value after extracting the principal components, can be obtained as

$$\boldsymbol{e}_{X_i} = \hat{\boldsymbol{X}}_{\boldsymbol{i}} - \boldsymbol{Q_i}^T \hat{\boldsymbol{W}}_{\boldsymbol{i}} \tag{5.18}$$

based on which a SPE score for each local device can be calculated as the criteria for anomalous timestamps detection, given by

$$SPE_i = \sqrt{\boldsymbol{e}_{X_i}^T \boldsymbol{e}_{X_i}} \tag{5.19}$$

Since SPE_i score after principal extraction is typically very small, any induction of anomalous timestamps will result in a huge fluctuation, based on which the corresponding local device can be diagnosed as unreliable. The real-time SPE_i value will be calculated and compared to the historical score $\overline{SPE_i}$, so that any newly collected timestamps meet the corresponding criteria will be reported by the cloud center as unreliable, which is given by

$$SPE_i(t) - \overline{SPE_i} > \rho_i \sigma_{SPE} \tag{5.20}$$

where σ_{SPE} is the standard deviation of the SPE score for the historically collected data while ρ_i is the threshold for determining the abnormality.

In terms of any unreliable nodes are detected, synchronization will be still performing, however, the reference time information will not be delivered to the unreliable node to enhance the overall synchronization reliability. Meanwhile, relay nodes should be reselected from the normal devices to avoid unexpected deterioration of the synchronization performance.

5.5 Performance Evaluation

In this section, a series of simulations are conducted to evaluate the performance of the proposed PANSO scheme from several aspects, including the distance-compensated clock offset estimation, reference time expansion, and reliability enhancement with abnormal timestamps during network synchronization.

5.5.1 Simulation Settings

In this simulation, a total number of 40 nodes are randomly deployed in a large industrial environment of $3000m \times 2000m$, where the distance between two nodes in one group can range from $30m$ to $1000m$. Surrounded by the 40 nodes, there are 3 target transmitters that can transmit

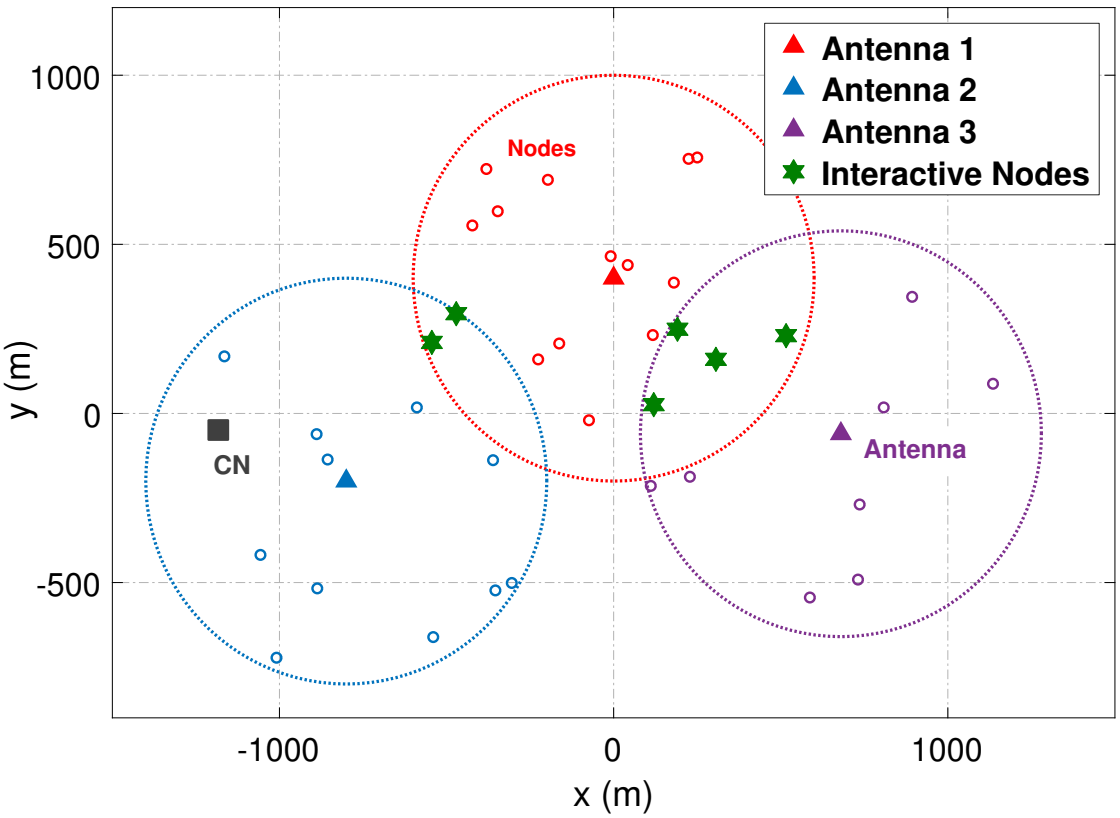


Figure 5.6: The distribution of the devices with three target transmitters in the proposed system. One CN and multiple INs exist in the network.

and broadcast wireless signals to their neighboring nodes occasionally. The distribution of the devices are shown in Fig. 5.6, where CN is selected by the cloud center prior to the simulation. Meanwhile, there are multiple INs between each two groups, which can receive the broadcast information from more than one target transmitters. Two relay nodes will be further selected from the available INs during reference time expansion phase. The simulations will be initiated by the CN in terms of the concurrent observation selection within its local group of devices, which surrounds the same target transmitter.

Moreover, to evaluation the effect of unreliable issues in terms of internal malfunctions of the IIoT devices and external malicious attacks, three kinds of different abnormalities that can lead to anomalous timestamps are considered, namely, inner malfunction, delay attacks, and tempering attacks, which are listed in Table 5.2 for definition and mathematical expressions. More specifically, external attacks are contrary to internal malfunctions, where the faulty times-

Table 5.2: Three kinds of abnormal behaviors considered in the simulation evaluation Subsection 5.5.4.

Types of Unreliability	**Descriptions**	**Quantitative Expression**
Internal malfunctions	Unexpected skew variation with environmental changes	$\eta_i = \gamma\eta_i$
Delay attacks	Delayed timestamps are delivered by the malicious relay nodes	$t_i^k = t_i^k - d_m$
Tampering attacks	Fake timestamps are transmitted by the abnormal devices	$t_i^k = t_m^k$

tamps are caused by the physical damage or interference of the enclosed crystal oscillator. The corresponding consequent of the internal malfunctions is the abnormal sensitivity of the enclosed oscillator to the environmental variations, where η_i is the temperature sensitivity factor for node i and γ_i reflects the degree of abnormality. Moreover, two types of external attacks are respectively considered while each of them will deteriorate the trustworthy of the timestamps delivered. On the one hand, malicious relay nodes will take advantage of delay attacks to transmit outdated timestamps to the local devices that are waiting for reference time information. By synchronizing with delayed timestamps, the clock accuracy cannot be improved as one of the direct consequents. Meanwhile, since the delayed timestamps are generated by the same clock, some of the existing powerful mechanisms that relies on the accurate clock modeling in dealing with malicious nodes, e.g., [112], may be impotent in addressing the issue. On the other hand, tampering attacks will be harmful for both local devices and relay nodes since their observation instants recorded can be manipulated so that the offset estimation and succeeding synchronization will be totally inaccurate. The synchronization performance under tampering attacks will be random and unpredictable due to the injection of unreliable timestamps.

Table 5.3: Receive-signal-strength-based distance estimation accuracy for three groups of devices.

	Real Avg. distance (m)	*EST Avg.* distance (m)	*EST bias* (m)	*EST bias* (μs)
Group 1	250.40	260.95	10.55	0.035
Group 2	365.29	368.92	3.62	0.012
Group 3	136.90	132.17	4.73	0.016

5.5.2 Distance-Compensated Offset Estimation

The most fundamental part of synchronization is the estimation of the clock offset for the distributed devices within the network compared to the reference time, as well as the timely elimination of the clock offset. Typically, one of the critical criteria for determining the synchronization performance is the averaged clock error among the involved devices throughout the network operation. In the proposed PANSO scheme, clock offset is estimated by comparing the timestamps of the locally matched observation and the received target signal at each local device with corresponding distance compensation based on the associated RSS values, as given in (5.6). As analyzed in Section 5.4.3, the effect of propagation delays will lead to inevitable observation bias if the location of each device is unknown, which will further result in significant synchronization inaccuracy. As a consequence, RSS-based method is adopted in the proposed scheme to address this issue, where the relative distances from the target transmitter to its neighboring devices are calculated according to the RSS values when the target signals are locally received, in accordance with (5.7) to (5.9).

The distance estimation accuracy prior to clock calibration is shown in Table 5.3, where the distance from each local device to its belonging target transmitter is different due to the random deployment, leading to an averaged distance in each group ranging from around $100m$ to $400m$. According to the RSS values obtained from the target signals, the estimated averaged distances are very close to the ground truth, with only a few meters as the estimation bias. As a consequence, the estimation bias in terms of the propagation time is much smaller than $1\mu s$, meaning that the effect of estimation bias on the synchronization performance can be negligible

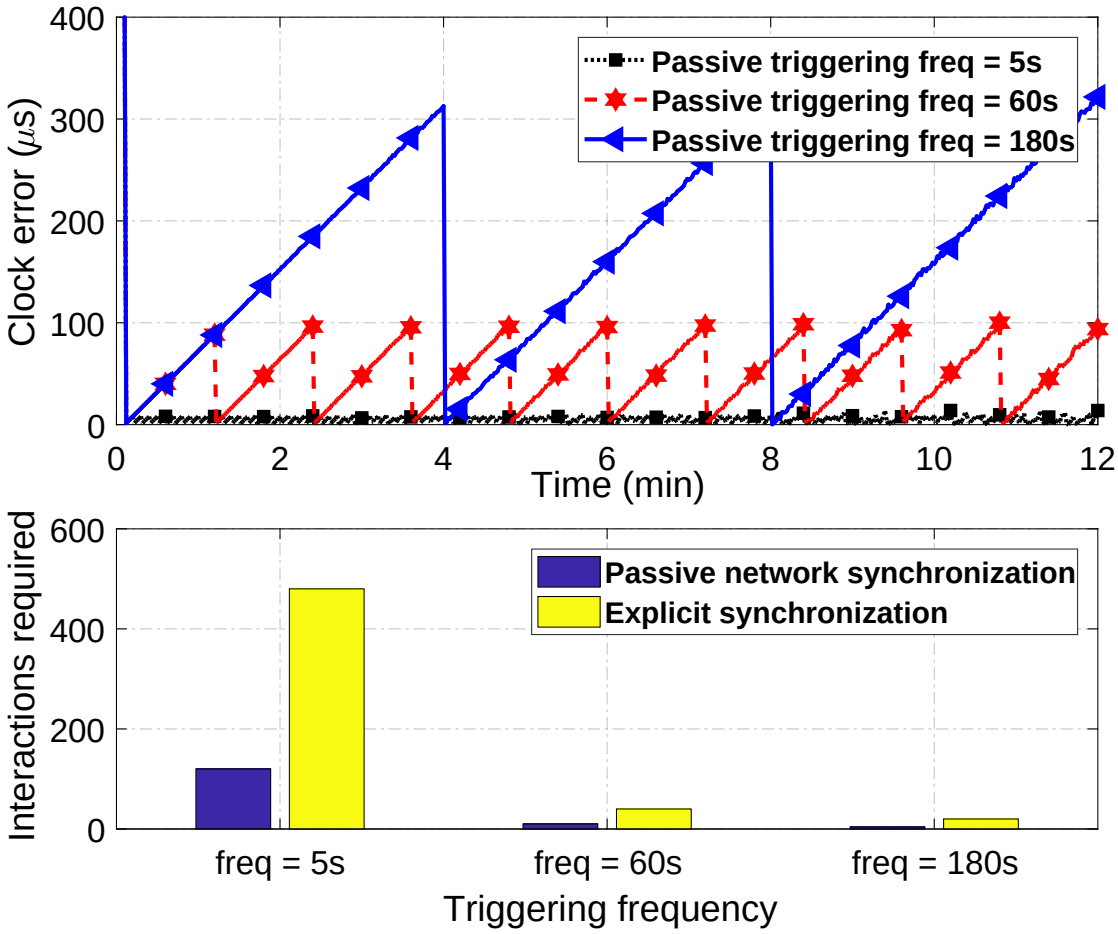


Figure 5.7: The evolution of clock errors after adopting PANSO with three different triggering frequencies. Compared to the conventional active synchronization methods, the passive one only requires a very small number of interactions for various triggering frequencies.

for further the clock calibration.

After adopting the proposed RSS-based distance compensation method, the clock calibration can be achieved among the first group of devices based on the reference time provided by CN. As shown in Fig. 5.7, clock errors can be accurately calibrated and almost eliminated after triggering the passive synchronization actions. Meanwhile, a higher triggering frequency can result in a better synchronization performance in terms of the achievable accuracy in long-term operations. For example, with a triggering frequency of $5s$, sub-microsecond synchronization performance is attainable for a group of devices, while a synchronization frequency of $180s$ can stop the averaged clock error exceeding $150\ \mu s$. Therefore, application-oriented synchronization can be provisioned with controllable clock accuracy to avoid overwhelming synchroniza-

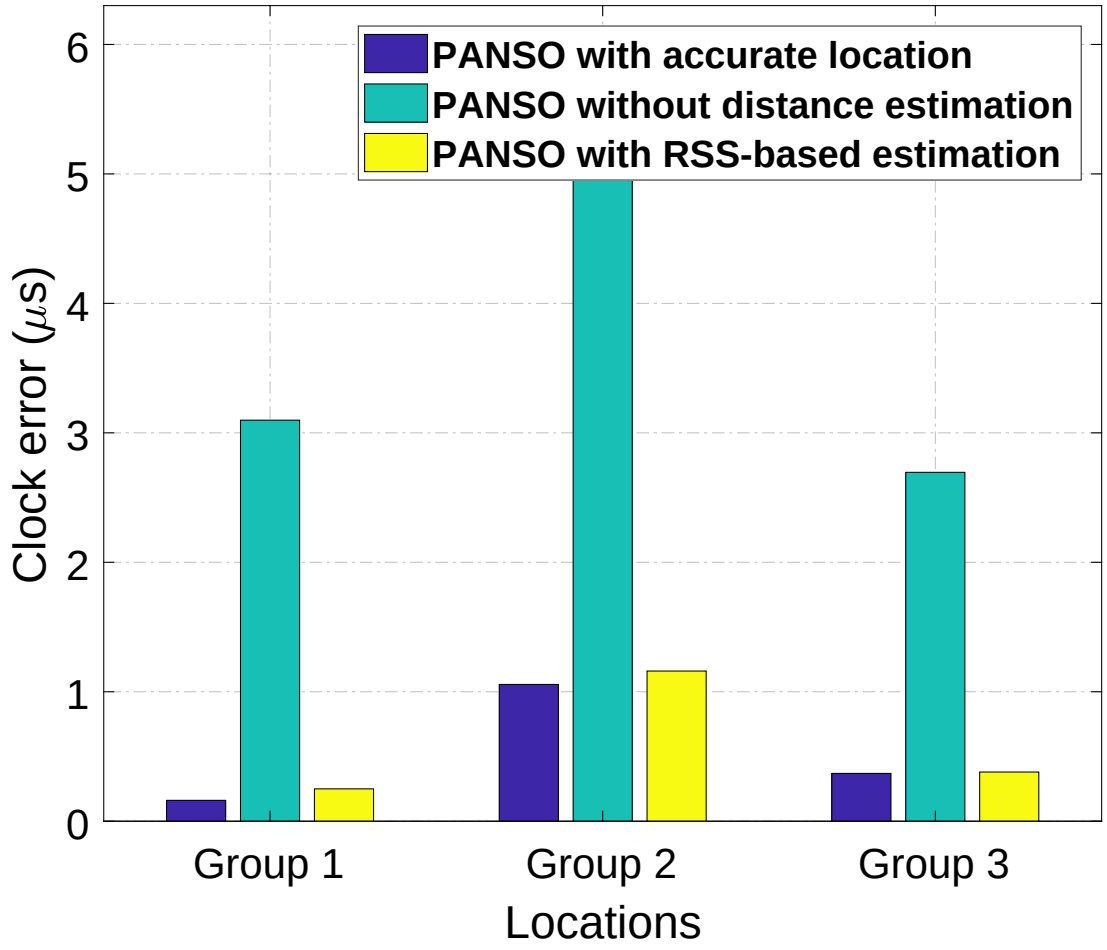


Figure 5.8: The comparison for the averaged achievable clock accuracy among each group of devices in three situations considering different distance-related strategies. RSS-based method can greatly enhance the clock accuracy.

tion processes. Moreover, compared to the conventional synchronization methods that hinge on frequent explicit interactions, e.g., PTP, PANSO can achieve accurate clock calibration with significantly reduced interactions for all IIoT devices. As a direct consequence, the uncertain network conditions will have a less significant effect on the synchronization performance, while more network resources can be saved for other critical industrial applications.

5.5.3 Reference Time Expansion

After achieving a consensual clock for a group of devices, global synchronization should be subsequently performed with the assistance of the relay nodes. For each group of devices, the

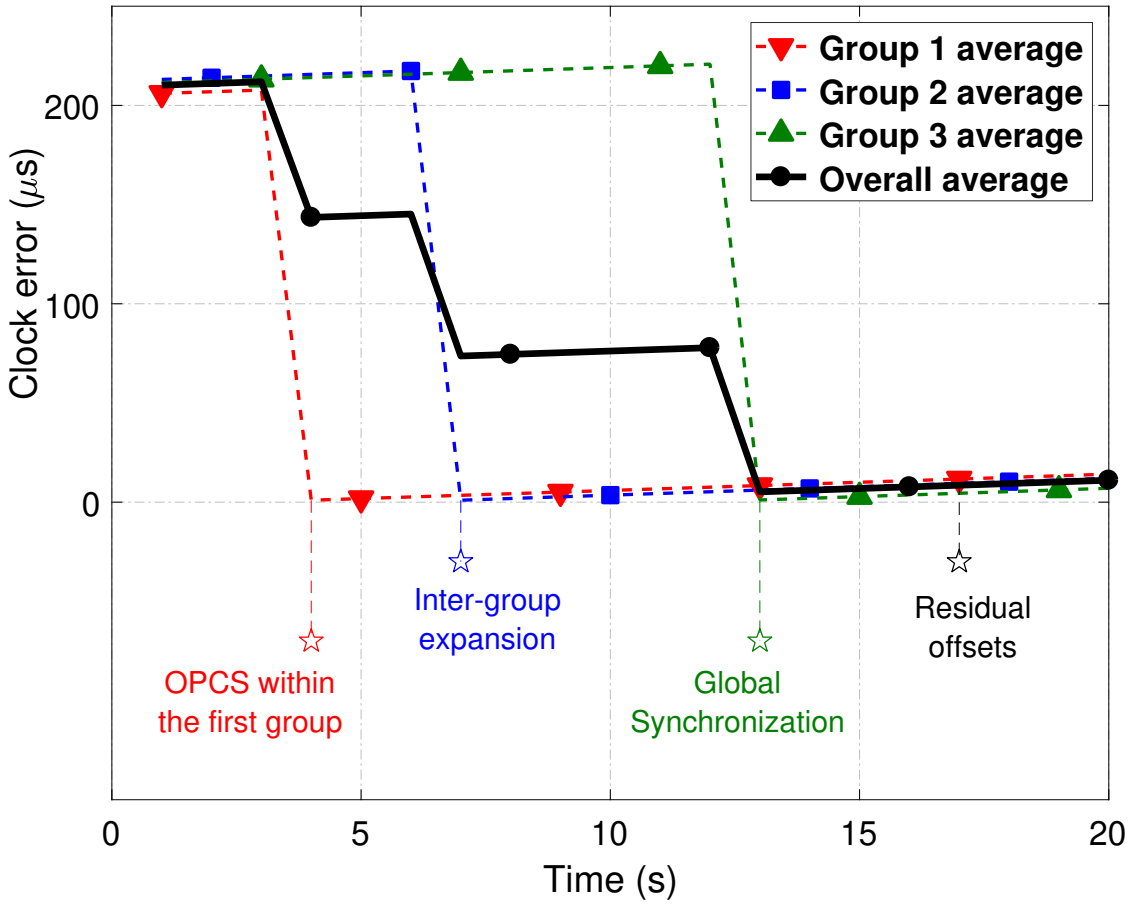


Figure 5.9: The expansion of time reference for three areas assisted by the relay nodes. Global synchronization is achieved gradually with minor residual offset after synchronization.

proposed RSS-based distance compensation scheme should be adopted in advance to enhance the offset estimation accuracy. As shown in Fig. 5.8, a simulation is conducted to compare the performance of the proposed PANSO scheme with three different types of location-related situations, namely, aware of accurate location of each device, RSS-based distance compensation method, and without considering the effect of propagation time. It can be observed that, for all three groups of devices, the clock errors for the case with accurate location information is the smallest, while PANSO adopting RSS-based distance compensation approach can reduce the clock error by a few microseconds, which is a significant improvement for accurate synchronization. Moreover, it indicates that with more steady communication environments or GPS-embedded devices in outdoor situations, the achievable clock errors can be further reduced since the distance estimation accuracy will be enhanced.

With the distance information estimated, global synchronization can be achieved accordingly, which is illustrated in Fig. 5.9. Since there are three neighboring areas required to be synchronized, the network synchronization is achieved gradually with the reference time expansion. Initially, the PANSO is conducted in the first group where CN is selected, while the averaged clock error within the first group will dramatically drop after clock calibration. Then the inter-group time reference expansion is conducted by the selected RNs, and other groups of devices will be synchronized accordingly. Therefore, the averaged clock errors for the three groups will reduce in different time instants and the overall averaged clock error will appear a decrement three times. The accomplishment of global synchronization throughout the IIoT network is achieved at the end, where all devices are accurately synchronized based on the clock offsets locally calculated. After global synchronization, there will be a minor clock error remaining and increasing, which is due to the calculation error and the effect of the clock skew. This residual issue will increase the clock errors in the long-term operation so that periodic clock calibrations are necessary to limit the residual offset within the application-oriented expectations.

5.5.4 Unreliable Node Detection

The reliability and security during synchronization can be affected by a wide range of issues, as illustrated in Table 5.2. Both the internal and external issues will lead to the anomalous timestamps during information exchange, causing unexpected synchronization errors especially when the relay nodes are unreliable.

PCA-assisted clock information analysis mechanism is adopted in the cloud center for both the abnormal nodes detection and reliable RNs selection. Any nodes with an uncommon time information will be regarded as unreliable, while only reliable INs will be selected as RNs for reference time expansion. The performance of the proposed PCA-assisted scheme is demonstrated in Fig. 5.10 based on the temporal information uploaded from the local devices. Specifically, the effectiveness of the detection scheme is denoted by true positive rate (TPR), which is the ratio of the correct detection r_c and the overall anomalous instants M_p. TPR for each local

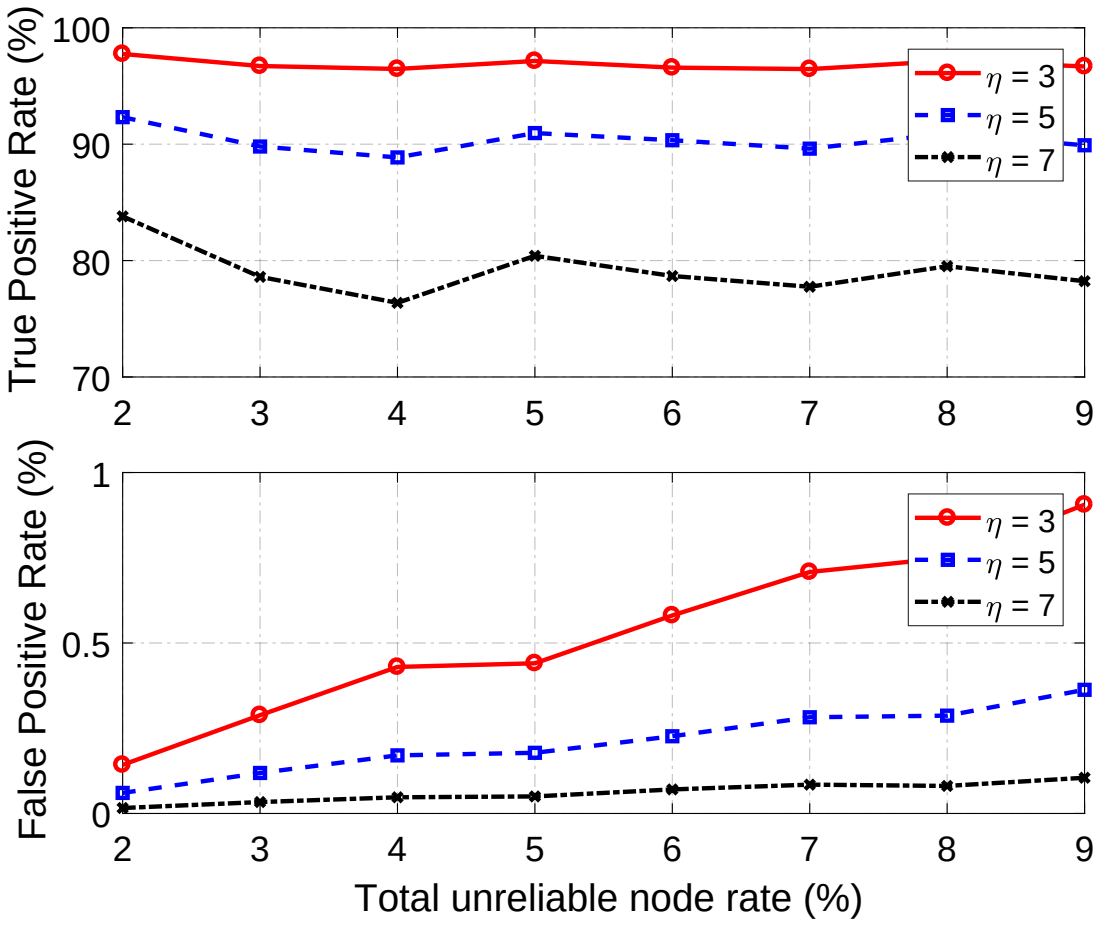


Figure 5.10: The performance of the proposed PCA-assisted unreliable node detection. With more unreliable nodes and higher detection threshold, the detection precision is increased with slightly increased false positive cases.

device can be calculated by

$$TPR = \frac{r_c}{M_p} = \frac{r_c}{\sum_{i=1,j=1}^{N_p,S_p} C_i(t_j)} \tag{5.21}$$

where $C_i(t_j)$ is the clock generation at the instant j of the device i. N_p and S_p are the total number of positive cases, i.e., anomalous devices and instants, respectively. It can be observed from Fig. 5.10 that, with the increment of the detection threshold ρ, the detection precision is enhanced accordingly. In other words, more unreliable devices can be successfully discovered by setting a stricter condition when the PCA-assisted scheme is adopted. However, a stricter condition will lead to false positive (FP) cases, i.e., normal devices incorrectly detected as anomalies. The false positive rate (FPR) in the proposed study is defined as the ratio of the FP

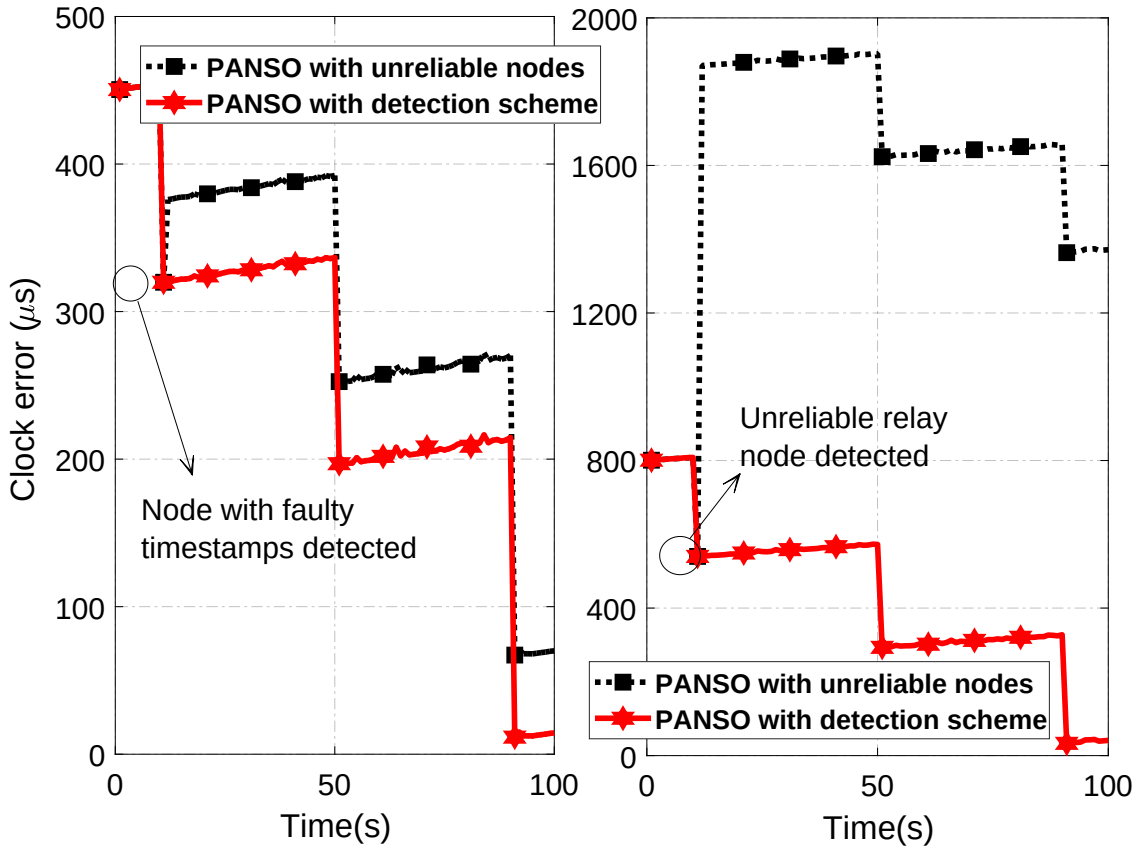


Figure 5.11: The improvement of clock accuracy after adopting the PCA-assisted unreliable node detection. The enhancement can be more significant if unreliable relay nodes are successfully detected.

reports r_e and the total number of normal instants M_n during the simulation period S_t, given by

$$FPR = \frac{r_e}{M_n} = \frac{r_t - r_c}{N_t S_t - M_p} \tag{5.22}$$

where r_t and N_t and the total number of anomaly reports and total number of devices in the network, respectively. It can be observed that the FPR is slightly increased with a stricter detection threshold, while the largest FPR is still lower than 0.5%, indicating that a reliable detection accuracy without overwhelming false alarms is achieved. Therefore, in this scheme, a stricter detection is preferred so that more unreliable devices can be successfully detected and the synchronization performance can be enhanced accordingly.

By utilizing the PCA-assisted detection scheme according to the historically recorded ob-

servation instants, various unreliable devices can be filtered so that future synchronization can be achieved with trustworthy nodes. The improvement of the synchronization performance is shown in Fig. 5.11, where it can be seen that an isolated node with error message will lead to the increment of the clock inaccuracy while unreliable RNs will cause significant clock error since the accurate reference time cannot be successfully expanded to the neighboring groups. However, by adopting the proposed unreliable detection scheme, both unreliable local devices and RNs can be effectively detected so that the clock errors are dramatically reduced. It is worth noting that if any of the initially assigned RNs are reported as unreliable after performing the PCA-assisted detection scheme in the cloud center, other reliable INs should be further selected as RNs to ensure the accomplishment of time information expansion and network-wide synchronization.

5.6 Conclusion

In this chapter, a passive network synchronization scheme based on concurrent observation was designed for large-scale industrial IoT systems to achieve synchronization without dedicated resources consumption during timestamps information exchange. To be more specific, by processing the commonly observed physical phenomenon in a group of IIoT devices, passive clock calibration was achieved at each local device with the alignment of the observation instants. RSS-based distance estimation method was adopted to compensate for the effect of propagation latency, while several relay nodes were selected to expand the time reference between the neighboring groups throughout the entire network. In addition, a PCA-assisted detection scheme was designed at the cloud center to investigate unreliable devices based on the historically uploaded observation instants, so that the synchronization performance under various uncertainties and external attacks was enhanced substantially. Extensive simulations were carried out to evaluate the improvement of the clock accuracy, synchronization efficiency, and the effectiveness of the unreliable device detection scheme. The results illustrated that the proposed scheme could achieve accurate clock calibrations in a passive manner and accomplish the efficient network-wide synchronization with the assistance of the relay devices selected. Furthermore, the synchronization performance was further enhanced by adopting the

proposed PCA-assisted scheme, which can effectively detect unreliable timestamps resulted form various kinds of abnormalities.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

With the extensive development of the IoT systems and their associated enabling technologies in industrial environments, industrial IoT (IIoT) systems are formed as one of the cornerstones in the accomplishment of industry 4.0. By integrating IIoT systems with traditional industrial systems in various application scenarios, the novel systems established can benefit from the ubiquitous connections among the involved IIoT devices as well as the pervasive deployment of sensing devices, which can provide exhaustive real-time information that can be applied to a wide range of applications. Diversified subsystems, including smart supply chain, advanced manufacturing system, and wireless sensor and actuator networks, can be established accordingly with the involved IIoT devices and the communication and interconnection among them. More complicated applications will hinge on the coordination and cooperation among different subsystems, leading to the stringent requirement on seamless connection, coherent collaboration, and accurate synchronization, for any devices assigned with collaborative tasks. As one of the prerequisites to support cohesion and coherence within the entire IIoT system, accurate clock synchronization can ensure the temporal alignment of the information transmitted so that the accuracy of data analysis and task scheduling can be enhanced. Traditional end-to-end clock synchronization protocols typically achieve clock calibration by frequently exchanging packets containing timestamps, while their intrinsic drawback in terms of resource-consuming and latency-sensitive will become exacerbated by the involvement of massive IIoT devices in

large-scale networks. Moreover, the design of clock synchronization protocols in IIoT systems is even more challenging due to the more complicated operating environment, intricate cross-standard communications, and demand for guaranteed security. This thesis investigated the design of IIoT-oriented clock synchronization protocols that can: 1). reduce resource consumption during clock synchronization without sacrificing the synchronization accuracy by analyzing timestamps generated at each clock, 2). obtain a better understanding of the inherent characteristics of each clock so that the situation-awareness can be achieved, and 3). secure the involved devices and their associated timestamps to tackle external malicious attacks and inner malfunctions during clock synchronization. Meanwhile, a digital twin-assisted platform is established to improve the efficiency of information exchange and the overall synchronization performance.

The primary contributions of this thesis and the corresponding conclusions are summarized as follows:

In Chapter 2, a comprehensive review of clock synchronization in IIoT systems has been conducted. Some synchronization-related preliminaries, including clock model and traditional synchronization procedures, were introduced first as some of the technical backgrounds. Different from conventional end-to-end synchronization techniques, there are multiple novel challenges in industrial environments, e.g., the cost-efficient and secure synchronization protocol design as well as the clock uncertainty induced from the operating environment. The effect of the diverse operating environment on the clock synchronization behavior was analyzed and summarized as well. According to these more stringent requirements, extensive literature reviews are conducted to investigate the existing methods in addressing each of the issues. Afterward, as one of the most promising enablers to achieve accurate and intelligent clock synchronization in IIoT systems, an introduction to digital twin and the survey of the relevant applications that are related to our research are given at the end.

In Chapter 3, an intelligent clustering-based distributed synchronization protocol is proposed for local area industrial IoT systems to achieve reliable, packet efficient, and secure clock synchronization. To be specific, the overall synchronization process consists of two phases. The first phase is responsible for initial skews correction and offsets compensation. After the first phase, all clocks will operate in the same frequency with different drift rates.

In the second phase, all nodes in the network are grouped into several clusters according to their VRS values, which is relevant to their clock quality and the environment. One of the nodes is selected as the chief cluster head to provide reference time while several cluster heads are elected, acting as connectors between the CCH node and cluster members. Different from simultaneous synchronization, in the proposed scheme, nodes with a high-quality clock are synchronized less frequently than those with an unstable clock. Meanwhile, according to their VRS values, a fault detection algorithm is also developed to overcome malicious attacks during synchronization processes. Simulation results proved that the proposed clock synchronization protocol could achieve a better performance in terms of accuracy and resource consumption. Furthermore, the proposed fault detection algorithm is capable of detecting harmful attacks and improving clock accuracy.

A digital-twin-enabled intelligent clock skew estimation and distributed synchronization approach had been proposed in this chaper to compensate for the effect of complicated environments on the heterogeneous oscillators in industrial IoT systems. A comprehensive digital twin model of each clock had been established by modeling the effect of external operating environments on the clock drift. By adopting the established virtual model at each device, the distributed local clock skews were compensated adaptively. Compared to the traditional clock synchronization approaches where clock skews are calculated frequently during network operation, the proposed method can avoid unnecessary and excessive packet exchange benefiting from a better understanding of clock behavior under different operating environments. The proposed digital-twin-enabled synchronization approach accomplished much higher clock accuracy with fewer packets required during network operation. Meanwhile, the performance improvements were even more significant under challenging network conditions, indicating that the proposed method is less sensitive to the packet delay variation and the dynamic operating environment for maintaining higher clock accuracy.

In this chapter, a passive clock synchronization scheme based on concurrent observations was designed to achieve undetectable synchronization without dedicated resources consumption during timestamps information exchange. The proposed synchronization scheme consists of two parts, namely, commonly observed signal processing to achieve passive clock synchronization and unreliable device detection to enhance synchronization performance against

security issues. To be more specific, a common signal was defined by the coordinator node to provide an opportunity for distributed devices to observe and compare their observation instants so that clock calibration can be achieved accordingly. RSS-based distance estimation method was adopted to compensate for the effect of propagation latency, while interactive nodes were selected to expand the time reference within the whole network. Moreover, a PCA-assisted detection scheme was designed at the cloud center to investigate unreliable devices, so that the synchronization performance can be enhanced substantially. Extensive simulations were carried out to evaluate the improvement of the clock accuracy and the effectiveness of unreliable device detection. The results illustrated that the concurrent observation-based approach can achieve accurate synchronization with the assistance of the interactive nodes. Furthermore, the synchronization performance can be further enhanced by adopting the proposed PCA-assisted scheme, which can detect various kinds of unreliable behaviors effectively.

6.2 Future Work

The technical issues on distributed clock synchronization in industrial IoT systems have been addressed in the thesis, where several intelligent-enhanced mechanisms are adopted to improve the synchronization performance from different perspectives. Many other challenges are still required to be investigated and addressed to further enhance the performance not only for the clock synchronization but also for the entire IIoT systems. The current research proposed in this thesis can be extended from several aspects, while some of the future research directions are identified and summarized in this section, including secured clock synchronization in IIoT systems, cross-standard clock synchronization, synchronization-assisted accurate data analytics, and digital twin platform-enabled collaborations. Details of these topics are given as follows:

- Security issues in distributed clock synchronization are still worth to be investigated based on the current studies proposed in this thesis. There are a wide variety of security requirements to be addressed, including clock identification and authorization during clock information dissemination, encryption of control command, hop-by-hop security guarantee, as well as security association. Different physical layer security and autho-

rization methods can be utilized during clock synchronization with the assistance of previously recorded information. Meanwhile, supporting security mechanisms at devices with limited computing and communication resources is also a challenging topic, which can be considered as a potential topic.

- Generally, resource-constrained devices are densely deployed in large-scale IIoT systems for ubiquitous sensing and seamless interconnections. The temporal consistency of the sensed data is necessary for the subsequent data analytics, but some of the devices may lack power or opportunity to exchange timestamps for accurate clock calibration. Based on this observation, novel synchronization schemes can be designed without posing burdens on the end-devices. A potential topic can be conducted based on the idea that a remote location, e.g., edge devices, can analyze the delivered data and the associated timestamps, based on which the clock inaccuracy among distributed devices can be predicted in advance so that the data can be corrected accordingly. The accuracy of the data analysis and network overhead will be significantly reduced by adopting this scheme.

- Multi-hop and cross-standard clock synchronization are always challenging due to the involvement of a large number of heterogeneous devices and the accumulated network latency. For a large scale IoT system, both the synchronization accuracy and efficiency will be dramatically reduced by simply adopting the traditional point-to-point synchronization methods. Although this thesis proposed three different network-wide clock synchronization schemes, their practicality might be degraded due to the lack of considering dedicated communication standard, e.g., IEEE 802.11ac and IEEE 802.15.4. A generic synchronization scheme that can be used in and between different protocols is required to be designed in the short future.

- Due to the dynamic inherent to the industrial IoT systems, there might be devices adding or disappearing from the current network, leading to the change of network topology. Meanwhile, for typical wireless sensor networks, broadcasting is a very commonly used communication method to transmit information throughout the network, which can help to reduce the network overhead. However, the current scheme proposed is achieved with packet-switching-based method without considering the dynamic of the topology.

Considering the design of topology discovery by utilizing the power of broadcasting might be beneficial to further enhanced the synchronization performance.

- In this thesis, a digital twin-enabled platform is established for achieving clock synchronization. However, such a platform will be even more useful for complex systems with heterogeneous devices and different applications. Networked control systems, which are enabled by accurate clock synchronization, can benefit from the establishment of the digital twin platform. The seamless collaboration and coordination among distributed devices and plants will be enabled by the involvement of their virtual counterparts so that the overall performance can be enhanced. The integration of digital twin platform and controller design will be one of the interesting topics.

- Accurate network synchronization plays a critical role in a wide variety of research fields, for example, simultaneous localization and mapping (SLAM) in multi-agent systems [113]. The precisely timestamped data packets are essential in supporting mutual comprehension, while the temporal misalignments among the involved robots will inevitably induce localization and mapping error during data processing. With the accurately synchronized agents throughout the large-scale systems, the distributed data packets delivered can be effectively fused for constructing the entire map efficiently. Moreover, in virtue of the accurate temporal alignment and improved consistency among the distributed agents, their local timelines can be utilized as the unified reference to track the distributed events. Therefore, a more informative construction with dynamically updated location and event knowledge can be achieved to support holistic situation-awareness in the long-term operation.

Bibliography

[1] S. Jeschke, C. Brecher, T. Meisen, D. Özdemir, and T. Eschert, "Industrial Internet of Things and Cyber Manufacturing Systems," in *Industrial internet of things*. Switzerland, Springer, 2017, pp. 3–19.

[2] M. Wollschlaeger, T. Sauter, and J. Jasperneite, "The Future of Industrial Communication: Automation Networks in the Era of the Internet of Things and Industry 4.0," *IEEE Industrial Electronics Magazine*, vol. 11, no. 1, pp. 17–27, March 2017.

[3] S. Raza, M. Faheem, and M. Guenes, "Industrial Wireless Sensor and Actuator Networks in Industry 4.0: Exploring Requirements, Protocols, and ChallengesA MAC Survey," *International Journal of Communication Systems*, vol. 32, no. 15, p. e4074, Aug. 2019.

[4] A. Haleem and M. Javaid, "Additive Manufacturing Applications in Industry 4.0: A Review," *Journal of Industrial Integration and Management*, vol. 4, no. 04, p. 1930001, Aug. 2019.

[5] M. ur Rehman, I. Yaqoob, K. Salah, M. Imran, P. Jayaraman, and C. Perera, "The Role of Big Data Analytics in Industrial Internet of Things," *Future Generation Computer Systems*, vol. 99, pp. 247–259, Apr. 2019.

[6] M. Ferrantino and E. Koten, "Understanding Supply Chain 4.0 and its Potential Impact on Global Value Chains," *GLOBAL VALUE CHAIN DEVELOPMENT REPORT 2019*, p. 103, 2019.

[7] M. Ben-Daya, E. Hassini, and Z. Bahroun, "Internet of Things and Supply Chain Management: A Literature Review," *International Journal of Production Research*, vol. 57, no. 15-16, pp. 4719–4742, 2019.

[8] K. B. Lee, S. Cheon, and C. O. Kim, "A Convolutional Neural Network for Fault Classification and Diagnosis in Semiconductor Manufacturing Processes," *IEEE Transactions on Semiconductor Manufacturing*, vol. 30, no. 2, pp. 135–142, May 2017.

[9] R. Zhao, D. Wang, R. Yan, K. Mao, F. Shen, and J. Wang, "Machine Health Monitoring Using Local Feature-Based Gated Recurrent Unit Networks," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 2, pp. 1539–1548, Feb. 2018.

[10] K. Židek, J. Pitel', M. Adámek, P. Lazorík, and A. Hošovskỳ, "Digital Twin of Experimental Smart Manufacturing Assembly System for Industry 4.0 Concept," *Sustainability*, vol. 12, no. 9, p. 3658, May 2020.

[11] R. Luo, N. Hua, X. Zheng, and B. Zhou, "High-Reliability Sub-Nanosecond Network Time Synchronization Method Enabled by Double-Frequency Distributed Time Synchronization," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 11, no. 1, pp. A40–A51, Jan. 2019.

[12] P. Jia, X. Wang, and X. Shen, "Digital Twin Enabled Intelligent Distributed Clock Synchronization in Industrial IoT Systems," *IEEE Internet of Things Journal*, pp. 1–12, 2020.

[13] L. Sliwczynski, P. Krehlik, J. Kolodziej, H. Imlau, H. Ender, H. Schnatz, D. Piester, and A. Bauch, "Fiber-Optic Time Transfer for UTC-Traceable Synchronization for Telecom Networks," *IEEE Communications Standards Magazine*, vol. 1, no. 1, pp. 66–73, March 2017.

[14] M. Rizzi, A. Depari, P. Ferrari, A. Flammini, S. Rinaldi, and E. Sisinni, "Synchronization Uncertainty Versus Power Efficiency in LoRaWAN Networks," *IEEE Transactions on Instrumentation and Measurement*, vol. 68, no. 4, pp. 1101–1111, Apr. 2019.

[15] P. Dass, S. Misra, and C. Roy, "T-Safe: Trustworthy Service Provisioning for IoT-Based Intelligent Transport Systems," *IEEE Transactions on Vehicular Technology*, pp. 1–1, Sept. 2020.

[16] T. Qiu, X. Liu, M. Han, H. Ning, and D. O. Wu, "A Secure Time Synchronization Protocol Against Fake Timestamps for Large-Scale Internet of Things," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1879–1889, Dec 2017.

[17] K. R. Choo, S. Gritzalis, and J. H. Park, "Cryptographic Solutions for Industrial Internet-of-Things: Research Challenges and Opportunities," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3567–3569, May 2018.

[18] G. T. 22.104, "Service Requirements for Cyber-Physical Control Applications in Vertical Domains," March 2019.

[19] S. Schriegel and J. Jasperneite, "Investigation of industrial environmental influences on clock sources and their effect on the synchronization accuracy of ieee 1588," in *2007 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication*, Vienna, Austria, 2007, pp. 50–55.

[20] Y. Zou, H. Liu, and Q. Wan, "Joint Synchronization and Localization in Wireless Sensor Networks Using Semidefinite Programming," *IEEE Internet of Things J.*, vol. 5, no. 1, pp. 199–205, Feb 2018.

[21] A. Mahmood, M. I. Ashraf, M. Gidlund, and J. Torsner, "Over-the-Air Time Synchronization for URLLC: Requirements, Challenges and Possible Enablers," in *Proc. 15th Int. Symp. Wireless Commun. Syst. (ISWCS)*, Lisbon, Portugal, Aug 2018, pp. 1–6.

[22] A. Mahmood, M. I. Ashraf, M. Gidlund, J. Torsner, and J. Sachs, "Time Synchronization in 5G Wireless Edge: Requirements and Solutions for Critical-MTC," *IEEE Communications Magazine*, vol. 57, no. 12, pp. 45–51, Dec. 2019.

[23] X. Xiao, B. Tang, and L. Deng, "High Accuracy Synchronous Acquisition Algorithm of Multi-hop Sensor Networks for Machine Vibration Monitoring," *Measurement*, vol. 102, pp. 10–19, Jan. 2017.

[24] A. Westenberger, T. Huck, M. Fritzsche, T. Schwarz, and K. Dietmayer, "Temporal Synchronization in Multi-Sensor Fusion for Future Driver Assistance Systems," in *2011 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication*, Munich, Germany, 2011, pp. 93–98.

[25] F. Tirado-Andrés, A. Rozas, and A. Araujo, "A Methodology for Choosing Time Synchronization Strategies for Wireless IoT Networks," *Sensors*, vol. 19, no. 16, p. 3476, Aug. 2019.

[26] Y. Geng, S. Liu, Z. Yin, A. Naik, B. Prabhakar, M. Rosenblum, and A. Vahdat, "Exploiting a Natural Network Effect for Scalable, Fine-Grained Clock Synchronization," in *15th USENIX Symposium on Networked Systems Design and Implementation*, Renton, WA, USA, 2018, pp. 81–94.

[27] H. A. Khattak, H. Farman, B. Jan, and I. U. Din, "Toward Integrating Vehicular Clouds with IoT for Smart City Services," *IEEE Network*, vol. 33, no. 2, pp. 65–71, Mar. 2019.

[28] X. Wang, "System, Method, and Apparatus for End-to-End Synchronization, Adaptive Link Resource Reservation and Data Tunnelling," Oct. 2019, uS Patent App. 16/312,884.

[29] S. Bhandari and X. Wang, "Prioritized Clock Synchronization for Event Critical Applications in Wireless IoT Networks," *IEEE Sensors J.*, vol. 19, no. 16, pp. 7120–7128, Aug. 2019.

[30] D. Sullivan, D. Allan, D. Howe, and F.Walls, *Characterization of Clocks and Oscillators*. National Institute of Standards and Technology, Technical Note 1337, 1990.

[31] F. Sivrikaya and B. Yener, "Time Synchronization in Sensor Networks: A Survey," *IEEE Network*, vol. 18, no. 4, pp. 45–50, July 2004.

[32] V. Paxson, "End-to-End Internet Packet Dynamics," *IEEE/ACM Transactions on Networking*, vol. 7, no. 3, pp. 277–292, June 1999.

[33] D. Shrestha, Z. Pang, and D. Dzung, "Precise Clock Synchronization in High Performance Wireless Communication for Time Sensitive Networking," *IEEE Access*, vol. 6, pp. 8944–8953, Feb. 2018.

[34] D. Fontanelli and D. Macii, "Accurate time synchronization in PTP-based industrial networks with long linear paths," in *2010 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication*, Portsmouth,USA, 2010, pp. 97–102.

[35] J. J. Pérez-Solano and S. Felici-Castell, "Adaptive Time Window Linear Regression Algorithm for Accurate Time Synchronization in Wireless Sensor Networks," *Ad Hoc Networks*, vol. 24, pp. 92–108, Jan. 2015.

[36] D. Zhou and T. H. Lai, "An Accurate and Scalable Clock Synchronization Protocol for IEEE 802.11-Based Multihop Ad Hoc Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 12, pp. 1797–1808, Dec. 2007.

[37] J. He, X. Duan, P. Cheng, L. Shi, and L. Cai, "Accurate Clock Synchronization in Wireless Sensor Networks with Bounded Noise," *Automatica*, vol. 81, pp. 350–358, July 2017.

[38] M. Akhlaq and T. R. Sheltami, "Rtsp: An accurate and energy-efficient protocol for clock synchronization in wsns," *IEEE Transactions on Instrumentation and Measurement*, vol. 62, no. 3, pp. 578–589, Mar. 2013.

[39] C. Benzad, M. Bagaa, and M. Younis, "Efficient Clock Synchronization for Clustered Wireless Sensor Networks," *Ad Hoc Networks*, vol. 56, pp. 13 – 27, March 2017.

[40] K. Noh, Q. Chaudhari, E. Serpedin, and B. Suter, "Power-Efficient Clock Synchronization using Two-Way Timing Message Exchanges in Wireless Sensor Networks," in *MILCOM 2006 - 2006 IEEE Military Communications conference*, Washington,USA, 2006, pp. 1–7.

[41] J. Chen, Q. Yu, Y. Zhang, H. Chen, and Y. Sun, "Feedback-based clock synchronization in wireless sensor networks: A control theoretic approach," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 6, pp. 2963–2973, July 2010.

[42] T. Mizrahi, "Security Requirements of Time Protocols in Packet Switched Networks," in *RFC 7384*, 2014.

[43] Y. Kikuya, S. M. Dibaji, and H. Ishii, "Fault-Tolerant Clock Synchronization Over Unreliable Channels in Wireless Sensor Networks," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 4, pp. 1551–1562, Dec. 2018.

[44] W. Dong and X. Liu, "Robust and secure time-synchronization against sybil attacks for sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 6, pp. 1482–1491, Dec 2015.

[45] K. Fan, S. Wang, Y. Ren, K. Yang, Z. Yan, H. Li, and Y. Yang, "Blockchain-Based Secure Time Protection Scheme in IoT," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4671–4679, June 2019.

[46] K. Sun, P. Ning, and C. Wang, "Secure and Resilient Clock Synchronization in Wireless Sensor Networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 2, pp. 395–408, Feb. 2006.

[47] R. Annessi, J. Fabini, and T. Zseby, "Securetime: Secure multicast time synchronization," *arXiv preprint arXiv:1705.10669*, May 2017.

[48] R. Sugihara and R. K. Gupta, "Clock Synchronization with Deterministic Accuracy Guarantee," in *Wireless Sensor Networks*, Heidelberg, Germany, 2011, pp. 130–146.

[49] C. Na, D. Obradovic, R. Scheiterer, G. Steindl, and F. Goetz, "Synchronization Performance of the Precision Time protocol," in *2007 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication*, Vienna,Astria, 2007, pp. 25–32.

[50] W. Zhou, H. Zhou, Z. Xuan, and W. Zhang, "Comparison Among Precision Temperature Compensated Crystal Oscillators," in *Proceedings of the 2005 IEEE International Frequency Control Symposium and Exposition, 2005.*, Vancouver,Canada, 2005, pp. 5 pp.–.

[51] F. Gong and M. L. Sichitiu, "Temperature Compensated Kalman Distributed Clock Synchronization," *Ad-Hoc Networks*, vol. 62, pp. 88 – 100, July 2017.

[52] M. Xu and W. Xu, "TACO: Temperature-Aware Compensation for Time Synchronization in Wireless Sensor Networks," in *2013 IEEE 10th International Conference on Mobile Ad-Hoc and Sensor Systems*, Hangzhou,China, 2013, pp. 122–130.

[53] T. Schmid, Z. Charbiwala, R. Shea, and M. B. Srivastava, "Temperature compensated time synchronization," *IEEE Embedded Systems Letters*, vol. 1, no. 2, pp. 37–41, Aug. 2009.

[54] G. Cena, S. Scanzio, and A. Valenzano, "A Neural Network Clock Discipline Algorithm for the RBIS Clock Synchronization Protocol," in *Proc. 14th IEEE Int. Workshop Factory Commun. Syst. (WFCS)*, Imperia,Italy, June 2018, pp. 1–10.

[55] Z. Yang, L. He, L. Cai, and J. Pan, "Temperature-Assisted Clock Synchronization and Self-Calibration for Sensor Networks," *IEEE Trans. Wireless Commun.*, vol. 13, no. 6, pp. 3419–3429, June 2014.

[56] T. Uhlemann, C. Schock, C. Lehmann, S. Freiberger, and R. Steinhilper, "The Digital Twin: Demonstrating the Potential of Real Time Data Acquisition in Production Systems," *Procedia Manufacturing*, vol. 9, pp. 113–120, 2017.

[57] F. Tao and M. Zhang, "Digital Twin Shop-Floor: A New Shop-Floor Paradigm Towards Smart Manufacturing," *Ieee Access*, vol. 5, pp. 20 418–20 427, Sep. 2017.

[58] Y. Xu, Y. Sun, X. Liu, and Y. Zheng, "A Digital-Twin-Assisted Fault Diagnosis Using Deep Transfer Learning," *IEEE Access*, vol. 7, pp. 19 990–19 999, Jan. 2019.

[59] S. Kumar, R. Madhumathi, P. Chelliah, L. Tao, and S. Wang, "A Novel Digital Twin-Centric Approach for Driver Intention Prediction and Traffic Congestion Avoidance," *Journal of Reliable Intelligent Environments*, vol. 4, no. 4, pp. 199–209, Oct. 2018.

[60] W. Serrano, "Digital Systems in Smart City and Infrastructure: Digital as a Service," *Smart cities*, vol. 1, no. 1, pp. 134–154, Nov. 2018.

[61] V. Havard, B. Jeanne, M. Lacomblez, and D. Baudry, "Digital Twin and Virtual Reality: A Co-Simulation Environment for Design and Assessment of Industrial Workstations," *Production & Manufacturing Research*, vol. 7, no. 1, pp. 472–489, Sep. 2019.

[62] A. El Saddik, "Digital twins: The convergence of multimedia technologies," *IEEE multimedia*, vol. 25, no. 2, pp. 87–92, Aug. 2018.

[63] H. Laaki, Y. Miche, and K. Tammi, "Prototyping a Digital Twin for Real Time Remote Control Over Mobile Networks: Application of Remote Surgery," *IEEE Access*, vol. 7, pp. 20 325–20 336, Feb. 2019.

[64] Y. .Liu, L. Zhang, Y. Yang, L. Zhou, L. Ren, F. Wang, R. Liu, Z. Pang, and M. Deen, "A Novel Cloud-based Framework for the Elderly Healthcare Services Using Digital Twin," *IEEE Access*, vol. 7, pp. 49 088–49 101, Apr. 2019.

[65] G. Hamilton, L. Swann, S. Kutty, G. Hearn, R. Nayak, J. Donovan, D. Polson, M. Rittenbruch, and R. Hellens, "Detecting Opportunities and Challenges for Australian Rural Industries: Final Report-February, 2018 (Publication No. 18/009)," 2018.

[66] C. C. Lin, D. J. Deng, Z. Y. Chen, and K. C. Chen, "Key Design of Driving Industry 4.0: Joint Energy-Efficient Deployment and Scheduling in Group-based Industrial Wireless Sensor Networks," *IEEE Communications Magazine*, vol. 54, no. 10, pp. 46–52, October 2016.

[67] S. Li, Q. Ni, Y. Sun, G. Min, and S. Al-Rubaye, "Energy-Efficient Resource Allocation for Industrial Cyber-Physical IoT Systems in 5G Era," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 6, pp. 2618–2628, June 2018.

[68] M. Lvesque and D. Tipper, "A Survey of Clock Synchronization Over Packet-Switched Networks," *IEEE Communications Surveys Tutorials*, vol. 18, no. 4, pp. 2926–2947, July 2016.

[69] R. Carli and S. Zampieri, "Network Clock Synchronization Based on the Second-Order Linear Consensus Algorithm," *IEEE Transactions on Automatic Control*, vol. 59, no. 2, pp. 409–422, Feb 2014.

[70] L. Narula and T. Humphreys, "Requirements for Secure Clock Synchronization," *IEEE Journal of Selected Topics in Signal Processing*, pp. 1–1, Aug. 2018.

[71] A. Mahmood, R. Exel, H. Trsek, and T. Sauter, "Clock Synchronization Over IEEE 802.11 - A Survey of Methodologies and Protocols," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 2, pp. 907–922, April 2017.

[72] "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems," *IEEE Std 1588-2008 (Revision of IEEE Std 1588-2002)*, pp. 1–300, 2008.

[73] M. J. Hajikhani, T. Kunz, and H. Schwartz, "A Recursive Method for Clock Synchronization in Asymmetric Packet Based Networks," *IEEE/ACM Transactions on Networking*, vol. 24, no. 4, pp. 2332–2342, Aug 2016.

[74] H. Wang, L. Shao, M. Li, B. Wang, and P. Wang, "Estimation of Clock Skew for Time Synchronization Based on Two-Way Message Exchange Mechanism in Industrial Wireless Sensor Networks," *IEEE Transactions on Industrial Informatics*, pp. 1–1, Nov. 2018.

[75] J. Wu, L. Zhang, Y. Bai, and Y. Sun, "Cluster-Based Consensus Time Synchronization for Wireless Sensor Networks," *IEEE Sensors Journal*, vol. 15, no. 3, pp. 1404–1413, March 2015.

[76] Z. Wang, P. Zhou, M. Li, and J. Wang, "Cluster-Based Maximum Consensus Time Synchronization for Industrial Wireless Sensor Networks," *Sensors*, vol. 17, no. 1, January 2017.

[77] H. Wang, D. Xiong, L. Chen, and P. Wang, "A consensus-based time synchronization scheme with low overhead for clustered wireless sensor networks," *IEEE Signal Processing Letters*, vol. 25, no. 8, pp. 1206–1210, Aug 2018.

[78] Z. Wang, P. Zeng, L. Kong, D. Li, and X. Jin, "Node-identification-based secure time synchronization in industrial wireless sensor networks," *Sensors*, vol. 18, no. 8, August 2018.

[79] T. Kohno, A. Broido, and K. C. Claffy, "Remote Physical Device Fingerprinting," *IEEE Transactions on Dependable and Secure Computing*, vol. 2, no. 2, pp. 93–108, April 2005.

[80] S. K. Bhatia, "Adaptive K-Means Clustering," *American Association for Artificial Intelligence*, pp. 1–5, 2004.

[81] Y. Kikuya, S. M. Dibaji, and H. Ishii, "Fault Tolerant Clock Synchronization over Unreliable Channels in Wireless Sensor Networks," *IEEE Transactions on Control of Network Systems*, pp. 1–1, July 2017.

[82] Y. Liao, E. de Freitas Rocha Loures, and F. Deschamps, "Industrial Internet of Things: A Systematic Literature Review and Insights," *IEEE Internet of Things J.*, vol. 5, no. 6, pp. 4515–4525, Dec 2018.

[83] A. Humayed, J. Lin, F. Li, and B. Luo, "Cyber-Physical Systems SecurityA Survey," *IEEE Internet of Things J.*, vol. 4, no. 6, pp. 1802–1831, Dec 2017.

[84] S. Vitturi, C. Zunino, and T. Sauter, "Industrial Communication Systems and Their Future Challenges: Next-Generation Ethernet, IIoT, and 5G," *Proc. IEEE*, vol. 107, no. 6, pp. 944–961, June 2019.

[85] M. Compare, P. Baraldi, and E. Zio, "Challenges to IoT-enabled Predictive Maintenance for Industry 4.0," *IEEE Internet of Things J.*, pp. 1–13, Dec. 2019.

[86] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial Internet of Things: Challenges, Opportunities, and Directions," *IEEE Trans. Ind. Informat.*, vol. 14, no. 11, pp. 4724–4734, Nov 2018.

[87] O. N. C. Yilmaz, Y. . E. Wang, N. A. Johansson, N. Brahmi, S. A. Ashraf, and J. Sachs, "Analysis of Ultra-Reliable and Low-Latency 5G Communication for a Factory Automation Use Case," in *Proc. IEEE Int. Conf. Commun.*, London, U.K., June 2015, pp. 1190–1195.

[88] L. Chettri and R. Bera, "A Comprehensive Survey on Internet of Things (IoT) Toward 5G Wireless Systems," *IEEE Internet Things J.*, vol. 7, no. 1, pp. 16–32, Jan. 2020.

[89] K. S. Yildirim and A. Kantarci, "Time Synchronization Based on Slow-Flooding in Wireless Sensor Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 1, pp. 244–253, Jan. 2014.

[90] F. Tao, H. Zhang, A. Liu, and A. Y. C. Nee, "Digital Twin in Industry: State-of-the-Art," *IEEE Trans. Ind. Informat.*, vol. 15, no. 4, pp. 2405–2415, April 2019.

[91] Y. He, J. Guo, and X. Zheng, "From Surveillance to Digital Twin: Challenges and Recent Advances of Signal Processing for Industrial Internet of Things," *IEEE Signal Process. Mag.*, vol. 35, no. 5, pp. 120–129, Sep. 2018.

[92] M. Leng and Y. Wu, "Low-Complexity Maximum-Likelihood Estimator for Clock Synchronization of Wireless Sensor Nodes Under Exponential Delays," *IEEE Trans. Signal Process.*, vol. 59, no. 10, pp. 4860–4870, Oct. 2011.

[93] B. Wang and Y.-P. Tian, "Time Synchronization in WSNs with Random Communication Delays: A Constant Gain Design," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 657 – 662, July 2017.

[94] F. Tao, J. Cheng, Q. Qi, M. Zhang, H. Zhang, and F. Sui, "Digital Twin-Driven Product Design, Manufacturing and Service with Big Data," *The International Journal of Advanced Manufacturing Technology*, vol. 94, no. 9-12, pp. 3563–3576, Feb. 2018.

[95] M. Zhou, J. Yan, and D. Feng, "Digital Twin Framework and its Application to Power Grid Online Analysis," *CSEE J. Power Energy Syst.*, vol. 5, no. 3, pp. 391–398, Sep. 2019.

[96] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile Edge Computing: A Survey," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 450–465, Feb. 2018.

[97] T. Yu, X. Wang, J. Jin, and K. McIsaac, "Cloud-Orchestrated Physical Topology Discovery of Large-Scale IoT Systems Using UAVs," *IEEE Trans. Ind. Informat.*, vol. 14, no. 5, pp. 2261–2270, May 2018.

[98] Y. Wu, Q. Chaudhari, and E. Serpedin, "Clock Synchronization of Wireless Sensor Networks," *IEEE Signal Processing Magazine*, vol. 28, no. 1, pp. 124–138, Jan. 2011.

[99] P. Jia, X. Wang, and K. Zheng, "Distributed Clock Synchronization Based on Intelligent Clustering in Local Area Industrial IoT Systems," *IEEE Trans. Ind. Informat.*, pp. 1–1, Aug. 2019.

[100] R. Veisllari, S. Bjornstad, J. P. Braute, K. Bozorgebrahimi, and C. Raffaelli, "Field-Trial Demonstration of Cost Efficient Sub-Wavelength Service Through Integrated Packet/Circuit Hybrid Network [Invited]," *IEEE J. Opt. Commun. Netw.*, vol. 7, no. 3, pp. A379–A387, March 2015.

[101] B. K. J. Al-Shammari, N. Al-Aboody, and H. S. Al-Raweshidy, "IoT Traffic Management and Integration in the QoS Supported Network," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 352–370, Feb. 2018.

[102] C. Callegari, R. G. Garroppo, S. Giordano, C. C. Labrozzo, G. Procissi, G. Minissale, and S. Topazzi, "Experimental Analysis of ViLTE Service," *IEEE Access*, vol. 6, pp. 21 129–21 139, Apr. 2018.

[103] H. Wang, H. Zeng, M. Li, B. Wang, and P. Wang, "Maximum Likelihood Estimation of Clock Skew in Wireless Sensor Networks With Periodical Clock Correction under Exponential Delays," *IEEE Trans. Signal Process.*, vol. 65, no. 10, pp. 2714–2724, May 2017.

[104] F. Naeem, M. Tariq, and H. V. Poor, "SDN-enabled Energy-Efficient Routing Optimization Framework for Industrial Internet of Things," *IEEE Transactions on Industrial Informatics*, pp. 1–1, July 2020.

[105] H. Fang, A. Qi, and X. Wang, "Fast Authentication and Progressive Authorization in Large-Scale IoT: How to Leverage AI for Security Enhancement," *IEEE Network*, vol. 34, no. 3, pp. 24–29, June 2020.

[106] T. Yu, X. Wang, and A. Shami, "A Novel Fog Computing Enabled Temporal Data Reduction Scheme in IoT Systems," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, Singapore, Singapore, 2017, pp. 1–5.

[107] F. H. Bijarbooneh, W. Du, E. C. . Ngai, X. Fu, and J. Liu, "Cloud-Assisted Data Fusion and Sensor Selection for Internet of Things," *IEEE Internet of Things Journal*, vol. 3, no. 3, pp. 257–268, June 2016.

[108] H. H. A. Arao, "Clock Synchronization Algorithm Between Wireless Sensor Nodes without Additional Control Message Exchanges," *WSEAS Transactions on Communications*, vol. 18, pp. 8–16, 2019.

[109] Y. Hu and G. Leus, "Robust Differential Received Signal Strength-Based Localization," *IEEE Transactions on Signal Processing*, vol. 65, no. 12, pp. 3261–3276, June 2017.

[110] S. C. Chan, H. C. Wu, and K. M. Tsui, "Robust Recursive Eigendecomposition and Subspace-Based Algorithms With Application to Fault Detection in Wireless Sensor Networks," *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 6, pp. 1703–1718, June 2012.

[111] V. Chatzigiannakis and S. Papavassiliou, "Diagnosing Anomalies and Identifying Faulty Nodes in Sensor Networks," *IEEE Sensors Journal*, vol. 7, no. 5, pp. 637–645, May 2007.

[112] D. Huang, W. Teng, C. Wang, H. Huang, and J. M. Hellerstein, "Clock Skew Based Node Identification in Wireless Sensor Networks," in *IEEE GLOBECOM 2008 - 2008 IEEE Global Telecommunications Conference*, New Orleans, LO, USA, 2008, pp. 1–5.

[113] Q. Shi, X. Cui, S. Zhao, S. Xu, and M. Lu, "BLAS: Broadcast Relative Localization and Clock Synchronization for Dynamic Dense Multi-Agent Systems," *IEEE Transactions on Aerospace and Electronic Systems*, Oct. 2020.

Curriculum Vitae

Name:	Pengyi Jia
Post-Secondary Education and Degrees:	2016 - Present, Ph.D. Electrical and Computer Engineering The University of Western Ontario London, Ontario, Canada 2015 - 2016, M.Eng Electrical and Computer Engineering The University of Western Ontario London, Ontario, Canada 2010 - 2014, B.Eng Electronic Information Science and Technology Hebei University Baoding, Hebei, China
Honours and Awards:	2019, Best Presentation at the UWO ECE Symposium
Related Work Experience:	Teaching Assistant The University of Western Ontario 2016 - 2020 Research Assistant The University of Western Ontario 2016 - Present

Publications:

1. **P. Jia** and X. Wang, "Nested Markov Chain - A Novel Approach to Model Network-Induced Constraints," in *Proc. 8th IEEE Annu. Inf. Technol., Electron. Mobile Commun. Conf. (IEMCON)*, Vancouver, BC, Canada, Oct. 2017, pp. 400-404.

2. **P. Jia**, X. Wang, and K. Zheng, "Distributed Clock Synchronization based on Intelligent Clustering in Local Area Industrial IoT Systems," *IEEE Trans. Ind. Informat.*, vol. 16, no. 6, pp. 3697-3707, June 2019.

3. **P. Jia**, X. Wang, and X. Shen, "Digital Twin Enabled Intelligent Distributed Clock Synchronization in Industrial IoT Systems," *IEEE Internet Things J.*, pp. 1-12, 2020 (Early Access).

4. **P. Jia**, X. Wang, and X. Shen, "Passive Network Synchronization based on Concurrent Observations in Industrial IoT Systems" (submitted to *IEEE Internet of Things Journal*).

5. **P. Jia** and X. Wang, "Intelligent and Low Overhead Network Synchronization over Large-Scale Industrial IoT Systems" (magazine paper, to be submitted).