

Article

Event-Triggered Adaptive Time Synchronization for Industrial Internet of Things

Zhaowei Wang  and Lei Zhou *

School of Electrical and Information Engineering, Jiangsu University, Zhenjiang 212013, China;
wangzhaowei@ujs.edu.cn

* Correspondence: 2222307047@stmail.ujs.edu.cn

Abstract

Time synchronization plays a critical role in enabling coordinated control and accurate data fusion in the Industrial Internet of Things (IIoT). However, most existing time-triggered synchronization protocols rely on periodic information exchange, which leads to considerable communication and energy consumption, particularly in large-scale and resource-constrained deployments. To address these limitations, this study proposes an adaptive event-triggered time synchronization scheme that eliminates the need for periodic communication. Unlike conventional approaches that employ fixed or predefined time-varying thresholds, the proposed method constructs a fully distributed triggering mechanism based on both local clock evolution and synchronization discrepancies observed from neighboring nodes. The triggering threshold evolves automatically according to the network synchronization state and does not require additional coordination messages. Theoretical analysis shows that the logical clock skews asymptotically converge to a common value, while the logical clock offset disagreement is ultimately bounded within an explicitly characterized neighborhood. Simulation results demonstrate that the proposed scheme achieves a more effective balance between synchronization accuracy and communication overhead, while producing more evenly distributed triggering events than several representative event-triggered synchronization methods.

Keywords: Industrial Internet of Things (IIoT); time synchronization; event-triggered synchronization; adaptive threshold; load balancing

1. Introduction

As a core enabler of Industry 4.0 and smart manufacturing, the Industrial Internet of Things (IIoT) is driving the comprehensive transformation of traditional manufacturing toward intelligence and networking through deep interconnection and real-time data exchange among physical devices, sensors, and control systems [1]. In this transformative process, time synchronization is not only the foundation for effective integration of massive heterogeneous data but also a key prerequisite for achieving distributed cooperative control, accurate event ordering, and system-wide scheduling consistency [2,3]. Indeed, a robust and efficient time synchronization mechanism serves as the cornerstone for unlocking the potential of IIoT systems and realizing leaps in system-level efficiency, quality, and productivity. For large-scale, highly dynamic IIoT networks, maintaining global time consistency has evolved from a supporting technology into indispensable infrastructure for achieving cooperative industrial operations [4].



Academic Editor: Alessandro Lo Schiavo

Received: 21 May 2026

Revised: 1 July 2026

Accepted: 9 July 2026

Published: 11 July 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

However, with the rapid expansion of IIoT network scale and the increasing complexity of application scenarios, achieving reliable time synchronization has become increasingly challenging. Traditional centralized synchronization mechanisms, which typically rely on point-to-point master–slave communication, suffer from limited scalability and poor energy efficiency in large-scale IIoT networks [5]. To overcome these limitations, various time synchronization mechanisms have been investigated for IIoT systems. Among them, distributed synchronization protocols, particularly those based on consensus theory, have attracted considerable attention. These protocols achieve time convergence through peer-to-peer interactions among nodes and offer advantages such as decentralization, high scalability, and strong robustness, making them well suited for large-scale and dynamically evolving IIoT networks [6–10]. Nevertheless, in IIoT networks, which often operate under strict energy constraints and limited communication resources, the overhead introduced by synchronization procedures may compete with bandwidth required for monitoring and control tasks. Therefore, designing synchronization mechanisms that maintain the required synchronization accuracy while reducing communication and energy overhead remains a critical research problem in large-scale IIoT networks.

To improve synchronization efficiency in resource-constrained IIoT networks, recent studies have explored several approaches, including protocol optimization, low-power synchronization platforms, energy consumption modeling, and intelligent energy management frameworks. In terms of protocol optimization, Elsharief et al. proposed the Intelligent Neighbor Knowledge-based Synchronization (INKS) algorithm, which reduces redundant message transmissions by selectively forwarding synchronization messages only to unsynchronized nodes [11]. In addition, low-power synchronization platforms, energy consumption models, and energy-aware management frameworks have also been investigated to enhance network efficiency [12–14]. These studies provide complementary perspectives for improving energy efficiency in IIoT networks.

However, despite these improvements, the communication overhead associated with synchronization operations remains a critical concern in resource-constrained IIoT networks, since most existing time synchronization approaches still rely on periodic communication frameworks. Such schemes cannot fundamentally eliminate the inherent overhead caused by continuous synchronization exchanges. To address this issue, event-triggered control has been introduced as an alternative communication scheduling paradigm, where information transmission or controller updates are performed only when a prescribed triggering condition is satisfied. This mechanism has been widely investigated in the consensus control of multi-agent systems. In particular, beyond basic static event-triggered rules, adaptive and dynamic event-triggered mechanisms have been developed to reduce communication requirements, avoid continuous monitoring, and improve scalability in distributed consensus problems [15–18]. Nevertheless, time synchronization is not a direct instance of general multi-agent state consensus. In IIoT time synchronization, node states are determined by hardware clocks, logical clock skews, and logical clock offsets, and relative clock information is usually obtained indirectly through timestamp exchanges. Therefore, event-triggered mechanisms designed for general multi-agent systems cannot be directly applied or compared without considering the specific clock model, synchronization objective, and communication structure of time synchronization systems. This distinction indicates that, although event-triggered ideas provide a promising way to reduce communication overhead, their application to clock synchronization requires problem-specific reformulation rather than a straightforward extension of existing consensus-based triggering rules.

Motivated by this observation, event-triggered concepts have gradually been incorporated into time synchronization protocols for wireless sensor networks, which form a

fundamental networking infrastructure for IIoT networks. Early studies introduced event-based distributed clock synchronization and event-triggered communication mechanisms for reducing unnecessary synchronization exchanges in WSNs [19,20]. García et al. [21] further formulated distributed clock synchronization from an event-triggered consensus perspective. Subsequently, Berneburg et al. [22] proposed a strongly non-Zeno dynamic event-triggered wireless clock synchronization algorithm, in which an auxiliary dynamic variable is introduced to guarantee a designable positive minimum inter-event time for each node. This work mainly focuses on improving the implementability of event-triggered clock synchronization. Event-triggered strategies have also been investigated in related discrete-time synchronization and more complex clock synchronization scenarios. For example, Jia et al. [23] developed an observer-based output feedback event-triggered control strategy using time-varying thresholds to achieve discrete-time synchronization under disturbances. Jia et al. [24] further proposed an asynchronous broadcast-based event-triggered control scheme for discrete-time clock synchronization, which reduces packet exchanges by avoiding unnecessary synchronous communication among nodes. Recent studies have further considered event-triggered time synchronization under communication delays [25], switching topologies and noise disturbances [26], time-varying delays [27], and varying clock skews [28]. These studies demonstrate the potential of event-triggered mechanisms for improving synchronization efficiency, while also indicating that adaptive threshold design specifically tailored to logical clock evolution and neighboring synchronization discrepancies remains insufficiently explored in IIoT time synchronization.

Despite these developments, existing event-triggered synchronization schemes still exhibit several critical limitations. In most existing studies, triggering thresholds are designed as fixed constants or predefined time-varying functions, such as exponentially decaying or inverse-time thresholds, and are therefore largely decoupled from the real-time synchronization state of the network as well as the dynamic interactions among neighboring nodes. As a result, constant thresholds [19] may cause nodes to remain inactive for long periods once synchronization stabilizes, thereby delaying the correction of clock skew, whereas decaying thresholds, including exponential [19] and inverse-time forms [25], may reduce communication effectively in the initial stage but tend to induce excessive triggering and redundant transmissions in later stages. Such locally isolated triggering behavior may further aggravate channel contention, cause uneven energy consumption across nodes, and even increase the risk of premature node failures. Moreover, the threshold design in many existing studies lacks a rigorous connection with Lyapunov stability analysis [19,25], so parameter selection often relies on empirical rules or extensive simulation-based tuning rather than systematic theoretical guidance. Fundamentally, these limitations stem from the lack of an adaptive threshold adjustment mechanism that can simultaneously account for synchronization dynamics, neighboring states, and stability requirements, making it difficult to achieve an effective balance between synchronization accuracy and resource efficiency in practical IIoT networks.

To overcome these limitations, this study proposes an adaptive threshold event-triggered time synchronization scheme based on neighbor-cooperative states. The core innovation of this research is to abandon the static or globally preset threshold paradigm and instead design a fully distributed dynamic triggering mechanism driven jointly by local and neighbor synchronization errors. This mechanism ensures that the trigger threshold for each node is no longer an isolated preset value but, rather, a dynamic variable that is updated according to the overall synchronization deviation of the network. The main contributions of this paper are summarized as follows:

- An adaptive threshold mechanism linked to the clock states of neighboring nodes is proposed. Based on the negative definiteness of the Lyapunov function deriva-

tive with respect to the logical clock skew, a novel distributed event-triggered condition is derived. The triggering threshold is dynamically constructed from the node's locally accumulated error and neighbor state information, enabling adaptive responses to network synchronization variations without introducing additional communication overhead.

- The proposed adaptive threshold mechanism produces a more balanced triggering pattern with moderate communication overhead, automatically adjusting the triggering frequency according to the synchronization state. This leads to faster convergence under large deviations while suppressing redundant transmissions near synchronization, thereby improving load balance and energy efficiency in the network.
- Comprehensive theoretical convergence analysis and systematic simulation evaluations are conducted. We rigorously prove that, under the proposed adaptive threshold mechanism, the logical clock skew achieves asymptotic uniform convergence, while the logical clock offset converges to a bounded neighborhood. Furthermore, extensive simulations compare the proposed scheme with representative event-triggered synchronization methods, including constant and exponential thresholds [19] and inverse-time thresholds [25], across multiple performance dimensions, demonstrating its superior performance in synchronization accuracy, communication efficiency, and triggering stability.

The remainder of this paper is organized as follows. Section 2 describes the adopted system and network models. Section 3 presents the proposed Event-Triggered Adaptive Time Synchronization (ETATS) for Industrial Internet of Things. Section 4 provides the convergence and performance analysis. Section 5 evaluates the performance of the proposed method through simulations. Finally, Section 6 concludes the paper and discusses future research directions.

2. System Model

2.1. Network Model

Consider an IIoT network consisting of n nodes. Its communication topology can be modeled as an undirected graph $G = (V, E)$, where $V = 1, 2, \dots, n$ is the set of nodes and $E \subseteq V \times V$ is the set of edges. If $(i, j) \in E$, then nodes i and j can communicate directly. The set of neighbors of node i is denoted by $N_i = \{j | (i, j) \in E, j \neq i\}$, and its size $|N_i| = d_i$ is called the degree of node i . The graph structure can be described by the adjacency matrix $\mathbf{A}$ (where $\mathbf{A}_{ij} = 1$ if $(i, j) \in E$; otherwise, $\mathbf{A}_{ij} = 0$), the degree matrix $\mathbf{D} = \text{diag}(d_1, \dots, d_n)$, and the Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{A}$. To ensure information can propagate throughout the network and achieve cooperative synchronization, we assume the graph G is connected. This assumption is a necessary condition for the convergence of distributed consensus algorithms [8]. If G is connected, then for any two nodes $i, j \in \mathcal{V}$, there exists a path connecting them.

2.2. Time Synchronization Model

Each node i is equipped with a quartz oscillator and a counter, forming its local hardware clock. Its reading $\tau_i(t)$ has a linear relationship with the absolute time t [8]:

$$\tau_i(t) = a_i t + b_i, \quad (1)$$

where a_i represents the hardware clock skew (i.e., frequency) and b_i denotes the hardware clock offset. Due to manufacturing variations and environmental factors, a_i and b_i differ among nodes and are typically assumed to be bounded; i.e., $a_i \in [a_{\min}, a_{\max}]$, $b_i \in [b_{\min}, b_{\max}]$, where $a_{\min}, a_{\max}, b_{\min}, b_{\max}$ are positive constants.

Since nodes cannot directly access absolute time t and should not modify their hardware clocks, each node maintains an adjustable logical clock $L_i(t)$ as its time reference for participating in cooperative tasks. This logical clock is obtained by compensating the hardware clock [8]:

$$L_i(t) = \alpha_i(t)\tau_i(t) + \beta_i(t), \quad (2)$$

where $\alpha_i(t)$ is the logical clock skew compensation parameter for correcting frequency differences and $\beta_i(t)$ is the logical clock offset compensation parameter for correcting phase differences. Substituting Equation (1) into Equation (2), the logical clock can be rewritten as a function of absolute time t :

$$L_i(t) = x_i(t)t + y_i(t), \quad (3)$$

where $x_i(t) = \alpha_i(t)a_i$ and $y_i(t) = \alpha_i(t)b_i + \beta_i(t)$ represent the node's logical clock skew and logical clock offset, respectively.

The goal of distributed time synchronization is to collaboratively update each node's compensation parameters $\alpha_i(t)$ and $\beta_i(t)$ through limited information exchange among nodes, such that the logical clocks of the entire network achieve asymptotic consensus in logical clock skew and practical synchronization in logical clock offset. Specifically, the desired synchronization objective can be expressed as

$$\lim_{t \rightarrow \infty} (x_i(t) - x_j(t)) = 0, \quad \lim_{t \rightarrow \infty} |y_i(t) - y_j(t)| \leq \varepsilon_y, \quad \forall i, j \in \mathcal{V}, \quad (4)$$

where $\varepsilon_y \geq 0$ denotes the ultimate bound of the logical clock offset disagreement. In the ideal case, $\varepsilon_y = 0$ corresponds to exact offset consensus, whereas under the proposed event-triggered implementation, ε_y is explicitly characterized by the triggering parameters and the network topology in the convergence analysis.

Unlike existing threshold design approaches that employ constant or monotonically decaying functions, such as those in [19,25], this work focuses on improving the communication efficiency and load balancing of the event-triggered mechanism itself. This enables a more adaptive triggering strategy that better matches the synchronization dynamics of the network while reducing communication overhead. The adopted network model and consensus-based synchronization framework remain consistent with existing synchronization research [7], while the offset synchronization objective is relaxed to practical synchronization under event-triggered communication. This setting preserves the generality and comparability of the research context while reflecting the bounded-error nature of the proposed scheme.

3. Event-Triggered Time Synchronization

This section presents an adaptive event-triggered time synchronization scheme for IIoT networks. The proposed approach integrates a distributed consensus-based logical clock update rule with a state-dependent event-triggered communication mechanism, aiming to reduce redundant message exchanges while maintaining high synchronization accuracy. Unlike conventional time-triggered synchronization protocols that rely on periodic information exchange regardless of the network state, the proposed method enables each node to autonomously determine when to communicate based on its local synchronization error and the inconsistency observed with neighboring nodes. Consequently, communication resources are utilized only when necessary, which is particularly beneficial for resource-constrained IIoT networks.

3.1. Compensation of Logical Clock Skew and Offset

Let $T > 0$ be the nominal update period measured by the local hardware clock, and let t_k^i denote the absolute instant at which node i performs its k -th update. Node i determines its update instants according to its local hardware clock, i.e., $\tau_i(t_k^i) = kT$. According to the hardware clock model in (1), the corresponding absolute update instant satisfies $t_k^i = \frac{kT - b_i}{a_i}$. Therefore, although all nodes update every T units measured by their own hardware clocks, their absolute update instants are generally not aligned because a_i and b_i differ across nodes. The resulting update process is thus asynchronous and pseudo-periodic. Moreover, $\tilde{\alpha}_i(t_k^i)$ and $\tilde{\beta}_i(t_k^i)$ represent the most recently broadcast skew and offset compensation parameters of node i , respectively. Between two consecutive update instants, these broadcast values are held constant; i.e.,

$$\begin{aligned}\alpha_i(t) &= \tilde{\alpha}_i(t_k^i), \\ \beta_i(t) &= \tilde{\beta}_i(t_k^i), \quad t \in [t_k^i, t_{k+1}^i).\end{aligned}\quad (5)$$

Other clock-related information stored at each node follows a similar piecewise-constant behavior. Using the neighbor information collected within an update interval, we adopt an update rule similar to [7], with the key difference that some neighbor quantities are estimated under event-triggered communications. The update laws for skew and offset compensation at node i are given by:

$$\alpha_i(t_k^i) = \tilde{\alpha}_i(t_k^{i-}) + \frac{1}{|N_i|} \sum_{j \in N_i} (a_{ij} \tilde{\alpha}_j(t_k^{j-}) - \tilde{\alpha}_i(t_k^{i-})), \quad (6)$$

$$\beta_i(t_k^i) = \beta_i(t_k^{i-}) + \frac{1}{|N_i| + 1} \sum_{j \in N_i} (\tilde{\alpha}_j(t_k^{j-}) \hat{\tau}_j(t_k^i) + \tilde{\beta}_j(t_k^{j-}) - \tilde{\alpha}_i(t_k^{i-}) \tau_i(t_k^i) - \tilde{\beta}_i(t_k^{i-})). \quad (7)$$

Denote by t_k^{i-} the instant right before node i performs its k -th update. The notation $\tilde{\alpha}_j(\cdot)$ and $\tilde{\beta}_j(\cdot)$ represent the neighbor-related parameters available at node i (received from the most recent broadcast of node j).

Let a_{ij} denote the relative clock skew (the ratio between the hardware clock skews of nodes i and j). Although the hardware skew is not directly measurable, a_{ij} can be estimated using two consecutive broadcast timestamps from node j :

$$a_{ij} = \frac{\tilde{\tau}_j(t_l^j) - \tilde{\tau}_j(t_k^j)}{\tau_i(t_l^j) - \tau_i(t_k^j)}. \quad (8)$$

In (8), $\tau_i(\cdot)$ is the hardware clock reading of node i recorded at the moments when node j broadcasts at t_k^j and t_l^j , while $\tilde{\tau}_j(\cdot)$ denotes the corresponding hardware clock readings of node j . Note that (7) requires the value of node j 's current hardware clock at node i 's update instant t_k^i . Under event-triggered communications, this value is generally unavailable in real time, and thus we estimate it by:

$$\hat{\tau}_j(t_k^i) = \tau_j(\tilde{t}_k^j) + a_{ij} (\tau_i(t_k^i) - \tau_i(\tilde{t}_k^j)). \quad (9)$$

where $\tilde{t}_k^j$ denotes the latest broadcast instant of node j known to node i , and $\hat{\tau}_j(t_k^i)$ is the estimated hardware clock reading of node j at t_k^i .

3.2. Event-Triggered Strategy

To reduce the periodic communication overhead in time-triggered synchronization, we design an adaptive event-triggered mechanism that responds to local synchronization state variations and suppresses unnecessary transmissions.

The mechanism consists of two triggering conditions. The first triggering condition is derived from a Lyapunov function constructed based on the logical clock skew and the update law (6). This Lyapunov function characterizes the deviation between the logical clock skew of the current node and the mean value of the sum of the initial logical clock skews of all nodes. By differentiating the Lyapunov function and applying Young's inequality [15], a sufficient condition ensuring the non-increasing property of the Lyapunov function can be obtained, which leads to the following triggering condition:

$$e_i^2(t_k^i) - \frac{\sigma q(1-q)}{|N_i|} \sum_{j \in N_i} \left(a_{ij} \tilde{\alpha}_j(t_k^{i-}) - \tilde{\alpha}_i(t_k^{i-}) \right)^2 \geq 0. \quad (10)$$

In (10), $e_i(t_k^i) = \alpha_i(t_k^{i-}) - \tilde{\alpha}_i(t_k^{i-})$ denotes the measurement error of node i at time t_k^i ; i.e., the accumulated variation of the skew compensation parameter since the last broadcast. When the triggering condition is satisfied, the measurement error of node i is reset to zero and begins accumulating again; otherwise, it continues to accumulate until the threshold is reached. If a node does not trigger for several update periods, the measurement error continues to accumulate until the triggering condition is satisfied. Meanwhile, the term $\sum_{j \in N_i} (a_{ij} \tilde{\alpha}_j(t_k^{i-}) - \tilde{\alpha}_i(t_k^{i-}))^2$ captures the neighbor state discrepancy, reflecting the inconsistency between node i and its neighbors in terms of logical clock skew. The parameter q is introduced through Young's inequality and can be selected as a design parameter. To maximize the dynamic threshold and thus reduce communication frequency, we maximize the concave quadratic function $q(1-q)$, which attains its maximum at $q = 0.5$. To ensure stable triggering behavior, we introduce the threshold sensitivity parameter σ , which should satisfy $0 < \sigma < 1$. The condition limits the allowable measurement error relative to the neighbor state discrepancy. In practice, σ controls the triggering sensitivity: smaller values improve convergence and accuracy with higher communication overhead, whereas larger values reduce communication at the cost of slower convergence. Thus, σ should be selected to balance synchronization performance and communication cost.

To prevent a potential reduction in convergence speed during the initial synchronization stage, an auxiliary triggering condition is introduced to enhance the responsiveness of the mechanism in the early phase:

$$|\beta_i(t_k^i) - \tilde{\beta}_i(t_k^{i-})| \geq c. \quad (11)$$

However, relying solely on the condition in (11) may lead to persistent oscillations in synchronization accuracy, since the logical clock error may fluctuate around the order of c due to the presence of the absolute time t . To address this issue, the threshold defined in (10) is introduced to adaptively regulate the triggering behavior. Meanwhile, the constant c also serves as a lower bound to prevent an excessive increase in triggering events during the later stage of synchronization.

After node i updates α_i and β_i according to (6) and (7), it broadcasts the updated clock parameters to all neighbors as long as either (10) or (11) is satisfied. The measurement error describes the accumulated deviation since the last broadcast, while the neighbor discrepancy essentially represents the average logical skew mismatch between node i and its neighbors. The complete procedure is summarized in Algorithm 1.

Algorithm 1 Event-Triggered Adaptive Time Synchronization (ETATS) at node i **Require:** Update period T ; initial compensation parameters $\alpha_i(0)$ and $\beta_i(0)$.**Ensure:** Logical clock synchronization.

```

1: Initialize  $\tilde{\alpha}_i \leftarrow \alpha_i(0)$  and  $\tilde{\beta}_i \leftarrow \beta_i(0)$ .
2: Set  $k \leftarrow 0$ .
3: while the synchronization process is active do
4:   Wait until the local hardware clock reaches  $t_k^i$ .
5:   for all  $j \in \mathcal{N}_i$  do
6:     Compute  $a_{ij}(t_k^i)$  according to (8).
7:     Estimate  $\hat{t}_j(t_k^i)$  according to (9).
8:   end for
9:   Update  $\alpha_i(t_k^i)$  according to (6).
10:  Update  $\beta_i(t_k^i)$  according to (7).
11:  Compute the adaptive triggering condition in (10).
12:  Compute the auxiliary triggering condition in (11).
13:  if (10) is satisfied or (11) is satisfied then
14:    Broadcast  $\alpha_i(t_k^i)$  and  $\beta_i(t_k^i)$  to all neighbors.
15:     $\tilde{\alpha}_i \leftarrow \alpha_i(t_k^i)$ .
16:     $\tilde{\beta}_i \leftarrow \beta_i(t_k^i)$ .
17:  end if
18:   $k \leftarrow k + 1$ .
19: end while

```

4. Convergence and Performance Analysis

This section analyzes the convergence properties of the proposed adaptive-threshold event-triggered synchronization scheme. According to (3), network-wide synchronization is achieved when both the skew-related state and the offset-related state reach steady agreement. Under the proposed protocol, we show that the logical clock skew achieves asymptotic consensus, whereas the logical clock offset converges to a bounded neighborhood whose radius can be explicitly characterized by the design parameters.

4.1. Matrix-Form System Model and Preliminaries

We first rewrite the update laws (6) and (7) into compact forms. For node i , the skew-related update reads

$$x_i(t_k^i) = \tilde{x}_i(t_k^{i-}) + \frac{1}{|N_i|} \sum_{j \in N_i} \tilde{x}_j(t_k^{i-}) - \tilde{x}_i(t_k^{i-}), \quad (12)$$

and the offset-related update is given by

$$y_i(t_k^i) = y_i(t_k^{i-}) + \frac{1}{|N_i| + 1} \sum_{j \in N_i} (\tilde{y}_j(t_k^{i-}) - \tilde{y}_i(t_k^{i-})) + \left(\frac{b_i}{(|N_i| + 1)a_i} + \frac{t_k^i}{|N_i| + 1} \right) \sum_{j \in N_i} (\tilde{x}_j(t_k^{i-}) - \tilde{x}_i(t_k^{i-})). \quad (13)$$

To facilitate the analysis, the asynchronous local-clock-driven updates are mapped into a global discrete-time sequence indexed by $k \in \mathbb{Z}_+$. Define the global state vectors

$$x(k) = [x_1(k), x_2(k), \dots, x_n(k)]^\top, \quad y(k) = [y_1(k), y_2(k), \dots, y_n(k)]^\top. \quad (14)$$

and the vectors of the most recently broadcast states

$$\tilde{x}(k) = [\tilde{x}_1(k), \tilde{x}_2(k), \dots, \tilde{x}_n(k)]^\top, \quad \tilde{y}(k) = [\tilde{y}_1(k), \tilde{y}_2(k), \dots, \tilde{y}_n(k)]^\top. \quad (15)$$

The corresponding measurement-error vectors are

$$e_x(k) = \tilde{x}(k) - x(k), \quad e_y(k) = \tilde{y}(k) - y(k). \quad (16)$$

Based on (12) and (13), we obtain the matrix iteration model

$$x(k+1) = (I - N_x L_k) \tilde{x}(k) = P_k \tilde{x}(k), \quad (17)$$

$$y(k+1) = (I - N_y L_k) y(k) - N_y L_k e_y(k) - (N_y R + N_y t_k) L_k (x(k) + e_x(k)). \quad (18)$$

Here, I is the identity matrix,

$$N_x = \text{diag}\left(\frac{1}{|N_1|}, \dots, \frac{1}{|N_n|}\right), \quad N_y = \text{diag}\left(\frac{1}{|N_1|+1}, \dots, \frac{1}{|N_n|+1}\right),$$

and L_k is a time-varying Laplacian-like matrix that activates exactly one node at each global time k (the activated node updates according to the underlying topology, while all other nodes keep their states unchanged). Moreover, $R = \text{diag}\left(\frac{b_1}{a_1}, \frac{b_2}{a_2}, \dots, \frac{b_n}{a_n}\right)$ is a diagonal matrix collecting the ratios between hardware-clock offset and skew. The representation in (17) and (18) converts the original asynchronous event-triggered updates into a switched discrete-time model. At each global step, the update matrix differs only in the row associated with the activated node, while the remaining rows keep the corresponding node states unchanged. Therefore, the effect of asynchronous triggering is captured by the switching sequence of $\{L_k\}$, or equivalently $\{P_k\}$, rather than by assuming that all nodes are updated synchronously. This form is convenient for convergence analysis because the evolution of the global state can be characterized by products of row-stochastic matrices.

To proceed, we introduce the notions of the ergodicity coefficient and the maximum disagreement (also called maximum discrepancy), together with a lemma similar to that in [19].

Definition 1 (Ergodicity coefficient). For a row-stochastic matrix $\mathbf{F}$, its ergodicity coefficient is defined as

$$\gamma(\mathbf{F}) = 1 - \min_{i,j} \sum_{k=1}^n \min\{f_{ik}, f_{jk}\}. \quad (19)$$

Definition 2 (Maximum disagreement). For a vector $\mathbf{v}$, define

$$S(\mathbf{v}) = \max_{i,j} |\mathbf{v}_i - \mathbf{v}_j|. \quad (20)$$

Lemma 1. The function $S(\cdot)$ satisfies:

(i) for any vectors $\mathbf{v}, \mathbf{w}$,

$$S(\mathbf{v} + \mathbf{w}) \leq S(\mathbf{v}) + S(\mathbf{w}). \quad (21)$$

(ii) for any row-stochastic matrix $\mathbf{F}$ and any vector $\mathbf{v}$,

$$S(\mathbf{F}\mathbf{v}) \leq \gamma(\mathbf{F}) S(\mathbf{v}). \quad (22)$$

Since each node updates at least once within any sufficiently long interval and the underlying graph is connected, there exists an integer $M > 0$ such that

$$P_{k+M} \cdots P_k > 0, \quad \forall k \in \mathbb{Z}_+. \quad (23)$$

Consequently, there exists a constant $\lambda \in (0, 1)$ such that

$$\gamma(P_{k+M} \cdots P_k) \leq 1 - \lambda, \quad \forall k \in \mathbb{Z}_+. \quad (24)$$

Equations (23) and (24) provide the key contraction property used in the subsequent analysis. They follow from the connected communication topology and the assumption that each node is activated at least once within a finite update window. The constant λ quantifies the worst-case mixing strength of every M -step matrix product. Hence, the convergence bounds derived below depend on the network connectivity, the maximum update window length, and the contraction capability of the corresponding stochastic matrix products.

4.2. Convergence of the Logical Clock Skew

By construction, the broadcast state $\tilde{x}(k)$ is updated only when a triggering event occurs. Specifically, if a trigger occurs at time k , then

$$\tilde{x}(k+1) = P_k \tilde{x}(k), \quad (25)$$

otherwise $\tilde{x}(k+1) = \tilde{x}(k)$. Sampling every M steps yields

$$\tilde{x}(kM) = [P_{kM-1} \cdots P_{(k-1)M}] \cdots [P_{M-1} \cdots P_0] \tilde{x}(0). \quad (26)$$

Using Lemma 1 and (24), there exists $\lambda_x \in (0, 1)$ such that

$$S(\tilde{x}(kM)) \leq (1 - \lambda_x)^k S(\tilde{x}(0)), \quad (27)$$

which implies asymptotic consensus of the broadcast skew states. Moreover, from (17), the instantaneous skew state $x(k+1)$ is generated by applying the row-stochastic matrix P_k to the latest broadcast state $\tilde{x}(k)$. Therefore, $x(k+1)$ is a convex combination of the broadcast skew states and does not enlarge their maximum disagreement. Consequently, the consensus of $\tilde{x}(k)$ also implies the consensus of the instantaneous skew state:

$$\lim_{k \rightarrow \infty} |x_i(k) - x_j(k)| = 0. \quad (28)$$

4.3. Convergence of the Logical Clock Offset

The convergence analysis of the logical-clock offset follows a similar framework as that of the skew, but it requires additional treatment of the coupling terms introduced by the skew component and the absolute time. By iterating (18) and applying Lemma 1 together with the contraction constant λ_y , we obtain the following bound:

$$\begin{aligned} S(y(kM)) &\leq (1 - \lambda_y)^k S(y(0)) \\ &\quad + \sum_{l=0}^{kM-1} (1 - \lambda_y)^{[k - \frac{l+1}{M}]} S(N_y L_l e_y(l)) \\ &\quad + \sum_{l=0}^{kM-1} (1 - \lambda_y)^{[k - \frac{l+1}{M}]} \left(S(N_y R L_l x(l)) + S(N_y R L_l e_x(l)) \right. \\ &\quad \left. + S(N_y t_l L_l x(l)) + S(N_y t_l L_l e_x(l)) \right). \end{aligned} \quad (29)$$

To derive an upper bound for the logical clock offset, we decompose the right-hand side of (29) into three components: (i) effect of the initial condition, (ii) accumulated effect

of measurement errors, and (iii) the coupling terms induced by skew and absolute time. Each component is analyzed separately in the following paragraphs.

(i) Effect of the initial condition.

The first term is handled analogously to the skew analysis. As iterations progress, it decays to zero:

$$\lim_{k \rightarrow \infty} (1 - \lambda_y)^k S(y(0)) = 0. \quad (30)$$

(ii) Accumulated effect of measurement errors.

We bound the second term in (29). First, using the Definition 2 and the fact that $N_y = \text{diag}(1/(|N_i| + 1))$, we obtain

$$S(N_y L_l e_y(l)) \leq \frac{2}{d_{\min}} \|L_l e_y(l)\|_{\infty}. \quad (31)$$

Next,

$$\|L_l e_y(l)\|_{\infty} = \max_i \left| \sum_{j \in N_i} (e_{y,i}(l) - e_{y,j}(l)) \right| \leq 2d_{\max} \|e_y(l)\|_{\infty}. \quad (32)$$

Combining (31) and (32) yields

$$S(N_y L_l e_y(l)) \leq \frac{4d_{\max}}{d_{\min}} \|e_y(l)\|_{\infty}. \quad (33)$$

Moreover, for each component,

$$\begin{aligned} |e_{y,i}(l)| &= |\tilde{y}_i(l) - y_i(l)| \\ &= |\tilde{\beta}_i(l) - \beta_i(l) + b_i(\tilde{\alpha}_i(l) - \alpha_i(l))| \\ &\leq |\tilde{\beta}_i(l) - \beta_i(l)| + |b_i| |\tilde{\alpha}_i(l) - \alpha_i(l)|. \end{aligned} \quad (34)$$

Since the triggering rule (11) enforces $|\tilde{\beta}_i(l) - \beta_i(l)| \leq c$, and $\tilde{\alpha}_i(l) - \alpha_i(l)$ vanishes as a consequence of the skew consensus (28), it follows that $\|e_y(l)\|_{\infty}$ is ultimately bounded. Consequently, the second term in (29) is ultimately bounded by a constant determined by c and the network degrees. In particular, we obtain the ultimate bound

$$\limsup_{l \rightarrow \infty} S(N_y L_l e_y(l)) \leq \frac{4d_{\max}c}{d_{\min}}. \quad (35)$$

Consequently, Part II does not necessarily converge to zero; instead, it converges to a uniformly bounded neighborhood whose radius is explicitly controlled by c and node degrees in the network. This is exactly the reason why the offset reaches agreement up to a finite accuracy bound.

(iii) Coupling terms induced by skew and absolute time.

By the skew consensus (28), we have

$$\lim_{k \rightarrow \infty} (x_i(k) - x_j(k)) = 0. \quad (36)$$

Since the time-varying Laplacian-like matrix satisfies $L_l \mathbf{1} = 0$ and $x(l)$ asymptotically approaches the consensus subspace, $L_l x(l) \rightarrow 0$. In addition, the skew measurement error $e_x(l)$ is constrained by the skew-triggering condition and decreases with the skew disagreement; hence, $L_l e_x(l) \rightarrow 0$. It should be noted that the last two coupling terms in (29) also contain the absolute-time factor t_l . Under the bounded hardware-clock skew assumption and the fixed nominal update period, t_l grows at most linearly with respect to

the global update index l . In contrast, the skew disagreement established in (27) decays geometrically. Therefore, the geometric decay of $L_l x(l)$ and $L_l e_x(l)$ dominates the linear growth of t_l , which gives

$$\lim_{l \rightarrow \infty} \left(S(N_y R L_l x(l)) + S(N_y R L_l e_x(l)) + S(N_y t_l L_l x(l)) + S(N_y t_l L_l e_x(l)) \right) = 0. \quad (37)$$

Combining Parts I–III, from (29) and (35) we obtain

$$\limsup_{k \rightarrow \infty} S(y(kM)) \leq \frac{4d_{\max}c}{\lambda_y d_{\min}}. \quad (38)$$

Finally, for any $m \in \{0, 1, \dots, M-1\}$, an analogous bound holds:

$$\limsup_{k \rightarrow \infty} S(y(kM + m)) \leq \frac{4Md_{\max}c(1 + \lambda_y)}{\lambda_y d_{\min}}. \quad (39)$$

Combining Parts (i)–(iii) yields the ultimate bound in (39). Therefore, the logical clock offset achieves agreement within a finite neighborhood whose radius is explicitly determined by the constant threshold c , the network topology parameters (e.g., $d_{\max}$ and $d_{\min}$), and the contraction factor λ_y . This bound is computable and time-invariant, which guarantees ultimate boundedness of the logical clock offset disagreement.

4.4. Performance Analysis and Discussion with Other Synchronization Paradigms

Beyond theoretical convergence, implementation complexity is also an important consideration for resource-constrained IIoT nodes. This subsection discusses the practical deployment cost of ETATS from three aspects; namely, computational overhead, storage requirement, and communication overhead. It then provides a qualitative comparison between ETATS and several other synchronization paradigms for industrial networks.

(a) Computational overhead.

Let $d_i = |\mathcal{N}_i|$ denote the number of one-hop neighbors of node i . In each synchronization update, ETATS mainly performs relative clock skew estimation, neighbor hardware clock estimation, logical clock skew compensation update, logical clock offset compensation update, and triggering condition evaluation. These operations are all based on one-hop neighbor information and therefore require a linear traversal over $\mathcal{N}_i$. Although the proposed adaptive triggering condition introduces an additional neighbor state discrepancy term, this term also depends only on one-hop neighbor information and can share the same neighbor traversal with the synchronization update. Therefore, the per-node computational complexity of ETATS in each update round is $O(d_i)$.

(b) Storage requirement.

Each node only needs to store its current and most recently broadcast clock compensation parameters, the most recently broadcast state information from neighboring nodes, and the timestamp information required for relative clock skew estimation. Thus, the storage requirement of ETATS increases linearly with the number of one-hop neighbors, and the per-node storage complexity is $O(d_i)$. In addition, the proposed adaptive triggering mechanism only introduces a few local scalar variables, such as the threshold sensitivity parameter, measurement error, and neighbor state discrepancy. It does not require global topology information, centralized scheduling information, or long-term historical buffers.

(c) Communication overhead.

Unlike periodic time synchronization mechanisms, ETATS does not require each node to broadcast its state in every synchronization period. A node broadcasts only when the adaptive triggering condition or the auxiliary triggering condition is satisfied. Let $I_i(l)$ denote whether node i triggers a broadcast at the l -th update, where $I_i(l) = 1$ if node i broadcasts and $I_i(l) = 0$ otherwise. After k synchronization update rounds, the total number of broadcasts is

$$C_{\text{ETATS}}(k) = \sum_{l=1}^k \sum_{i=1}^n I_i(l) \quad (40)$$

In contrast, a periodic synchronization mechanism usually requires nk broadcasts. Therefore, the communication overhead of ETATS is determined by the actual number of triggering events rather than by the fixed synchronization period. Since each triggered transmission only contains the clock-related information required for synchronization, the payload of each broadcast is of constant order. No additional centralized coordination, global scheduling, or threshold negotiation message is required.

Furthermore, compared with existing event-triggered threshold methods, the advantage of ETATS does not lie in simply pursuing the minimum number of triggers, but in improving the effectiveness of each communication event. A constant-threshold method cannot perceive the real-time synchronization state. If the threshold is too large, a node may remain silent for a long time, and its neighbors may continuously use stale information, which limits synchronization accuracy. If the threshold is too small, the triggering frequency increases. Exponential and inverse-time thresholds vary with time, but their threshold evolution is governed by predefined functions and lacks a direct connection with the actual synchronization discrepancy among nodes. In contrast, ETATS incorporates neighbor synchronization discrepancy into the triggering condition, linking the broadcast behavior to the real-time synchronization demand. Therefore, its triggering events are generally more meaningful for synchronization correction: nodes can broadcast in time when their discrepancies from neighbors are large, while redundant broadcasts are suppressed as the network approaches synchronization. In this way, ETATS improves the contribution of each communication event to synchronization accuracy and triggering-load balance under the same single-broadcast payload, thereby making more effective use of limited communication resources.

(d) Discussion with other synchronization paradigms.

Other studies in industrial networks have also explored several representative synchronization paradigms, including digital-twin-assisted approaches [29], TSN/PTP-based mechanisms [30], and AI-assisted methods [31], each with different design principles and system assumptions. Digital-twin-based methods rely on virtual system models for prediction and optimization, but typically introduce additional modeling and maintenance overhead. TSN/PTP-based methods achieve high-precision synchronization via standardized protocols and hardware support, but depend on dedicated network infrastructure. AI-assisted methods improve adaptability through learning-based estimation or parameter tuning, but usually require training data and additional computational resources. In contrast, the proposed ETATS scheme is a fully distributed event-triggered mechanism that relies only on local clock information and neighbor interactions. It does not require digital twins, specialized hardware, or offline training. The triggering rule is derived from a Lyapunov-based design, ensuring lightweight implementation with theoretical guarantees. Therefore, ETATS differs from these paradigms in terms of communication model, implementation requirements, and theoretical design, making it suitable for resource-constrained IIoT synchronization scenarios.

5. Simulation Results and Analysis

In this section, numerical simulations are conducted to evaluate the performance of ETATS. The proposed method is evaluated from the perspectives of convergence performance, triggering behavior, communication efficiency, and communication load distribution. To highlight the advantages of the proposed strategy, ETATS is compared with several representative event-triggered synchronization schemes. Furthermore, scalability and robustness are examined under different network sizes and adverse communication conditions.

5.1. Simulation Setup

A random industrial internet of things consisting of 10 sensor nodes is considered, as shown in Figure 1. The network topology remains fixed during the simulation, and the underlying communication graph is undirected and connected. Each node is equipped with an imperfect hardware clock characterized by an individual clock skew and clock offset.

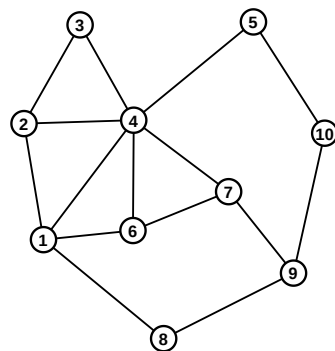


Figure 1. Random topology of the 10-node Industrial Internet of Things used in the simulations.

To ensure fairness, all synchronization schemes are evaluated under identical network topology, hardware clock model, and consensus-based update rules, differing only in their triggering mechanisms. Specifically, the proposed ETATS scheme is compared with the constant-threshold and exponentially decaying threshold schemes in [19], as well as the inverse-time threshold scheme in [25].

Hardware clock skews are randomly generated within the interval $a_i \in [0.9999, 1.0001]$, while the hardware clock offsets are randomly initialized within $b_i \in [0, 2 \times 10^{-6}]$. The logical clock compensation parameters are initialized as $\alpha_i(0) = 1$ and $\beta_i(0) = 0$. The update period is fixed at $T = 1$ s, and the total simulation time is 100 s. The parameters involved in the proposed ETATS triggering mechanism are selected as $\sigma = 0.05$, $c = c_\alpha = 2 \times 10^{-6}$, and $q = 0.5$.

For all comparison schemes, the triggering condition associated with the logical clock offset parameter β_i follows (11). For the constant-threshold scheme in [19], the triggering condition for the skew compensation parameter α_i is $|\alpha_i(t_k^i) - \tilde{\alpha}_i(t_k^{i-})| \geq c_\alpha$. For the exponentially decaying threshold scheme in [19], the triggering condition is defined as $|\alpha_i(t_k^i) - \tilde{\alpha}_i(t_k^{i-})| \geq 0.02e^{-w\tau_i(t_k^i)}$, where $w = 0.5$ determines the decay rate. For the inverse-time threshold scheme in [25], the triggering condition is given by $|\alpha_i(t_k^i) - \tilde{\alpha}_i(t_k^{i-})| \geq \frac{c_\alpha}{\tau_i(t_k^i)}$, where $\tau_i(t_k^i)$ represents elapsed local time used to generate the inverse-time threshold.

5.2. Convergence Performance

We first evaluate the convergence performance of the proposed ETATS scheme by plotting the evolution of the global maximum errors of the logical clock skew. As shown in Figure 2a, it can be observed that the logical clock skew converges within approximately 10 s for all four triggering strategies. This rapid convergence is mainly attributed to the auxiliary

triggering condition in (11), which ensures sufficient information exchange among nodes during the initial synchronization stage. Among the compared schemes, ETATS achieves the fastest reduction of the maximum skew error in the early stage. This improvement results from the triggering condition in (10), which captures neighbor state variations and enables more effective triggering decisions, thereby accelerating convergence.

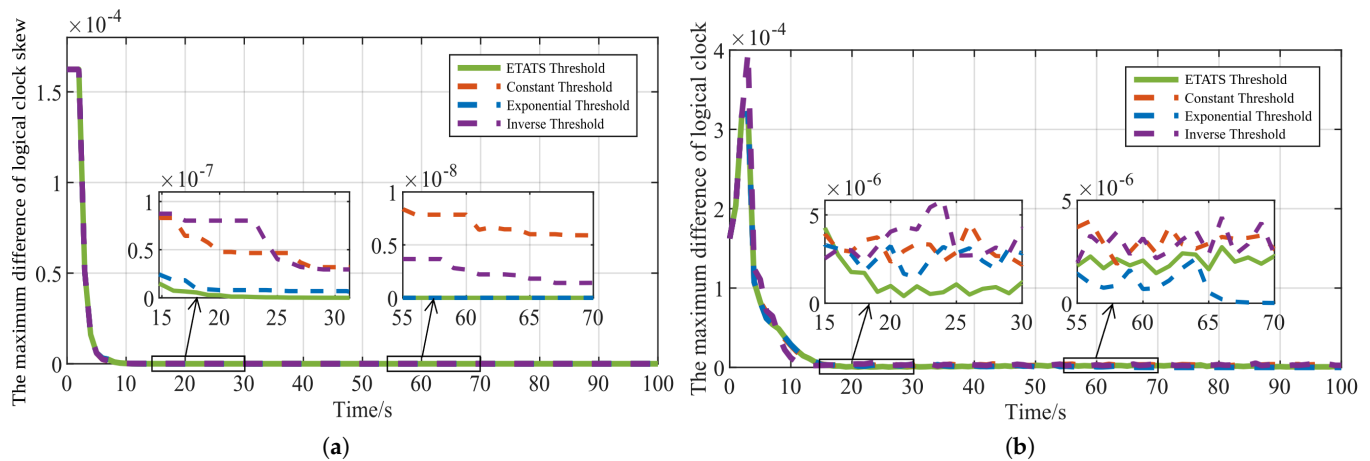


Figure 2. The logical clock synchronization error versus time under different schemes: (a) Logical clock skew; (b) Logical clock.

In the later stage, the constant-threshold and inverse-time threshold schemes exhibit lower precision than ETATS and the exponential-threshold scheme. This is because these schemes rely on prescribed thresholds that do not adapt to real-time neighbor state discrepancies, making them less sensitive to measurement error variations and therefore triggering fewer communications. The inverse-time threshold achieves slightly higher accuracy than the constant threshold due to the decay term in its triggering condition, which increases the communication frequency.

Figure 2b illustrates the asynchronous convergence behavior of the evaluated schemes. The logical clock offset converges after the skew stabilizes, and the entire network reaches synchronization at approximately 20 s. Compared with the constant-threshold, exponentially decaying threshold, and inverse-time threshold schemes, ETATS achieves a faster decay of the maximum logical clock error in the initial stage. This improvement stems from its neighbor-aware dynamic threshold mechanism, which more effectively captures local synchronization deviations and accelerates convergence.

Meanwhile, the exponentially decaying threshold scheme behaves similarly to a time-triggered mechanism in the later stage (around 65 s) because the threshold becomes extremely small and causes persistent triggering. Although this reduces the logical clock error rapidly, it significantly increases communication overhead. In contrast, ETATS maintains a better balance between synchronization accuracy and communication cost.

5.3. Triggering Behavior

To investigate the triggering behavior of different event-triggered synchronization schemes, the triggering instants of all nodes are shown in Figure 3. Triggering events caused by the dynamic threshold condition and the auxiliary triggering condition are marked with stars and circles, respectively. For the constant-threshold and inverse-time threshold schemes, prolonged communication silence occurs once the local synchronization error falls below the predefined threshold. During these intervals, nodes update their clock parameters using outdated neighbor information, which may delay error correction and lead to uneven energy consumption.

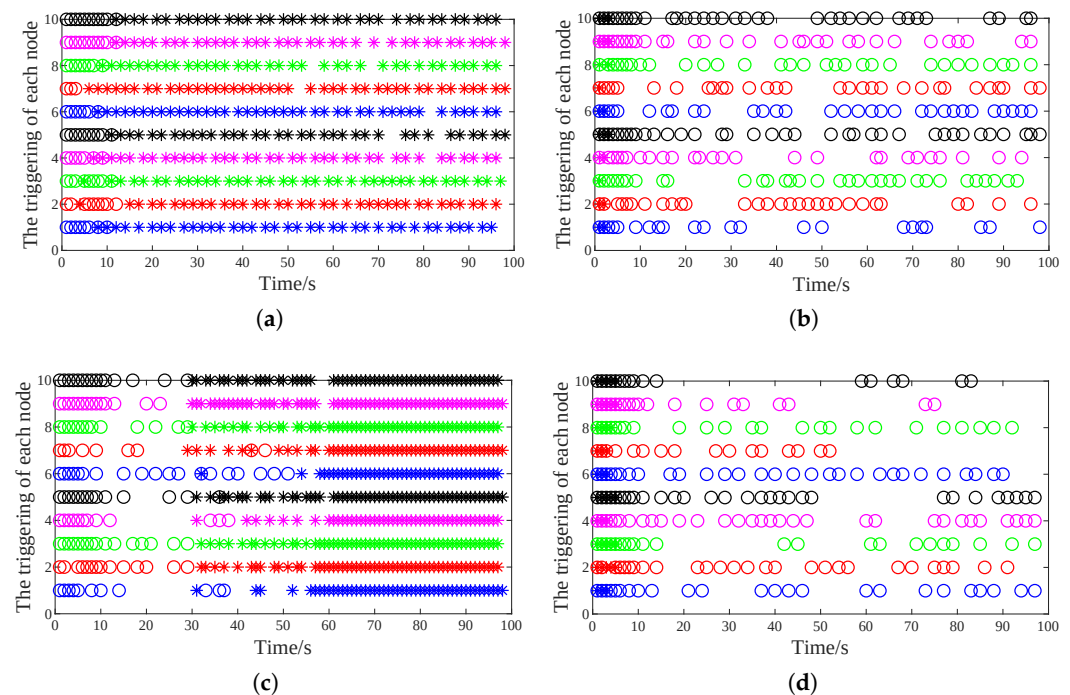


Figure 3. Triggering in these images. instants of all nodes under different event-triggered synchronization schemes: (a) ETATS. (b) Constant-threshold scheme. (c) Exponential-threshold scheme. (d) Inverse-time-threshold scheme.

In contrast, the exponential-threshold scheme exhibits frequent triggering in the later stage because the threshold decays rapidly to a value far below the measurement error. As a result, the communication pattern becomes similar to periodic synchronization, causing unnecessary communication overhead. Compared with these schemes, ETATS shows a more balanced triggering pattern. As illustrated in Figure 3a, the auxiliary triggering condition dominates in the initial stage, producing dense triggering events that accelerate convergence. After approximately 10 s, the dynamic threshold becomes dominant and triggering events are more evenly distributed over time, avoiding both prolonged silence and excessive triggering.

5.4. Communication Load Balancing

To quantitatively evaluate the impact of triggering behavior on network load balancing, we analyze the total number of broadcasts generated by each node during the simulation.

The distribution of broadcasts reflects the effectiveness of each algorithm in balancing network energy consumption. Figure 4 presents a histogram of broadcast counts for each node under different schemes over the entire simulation duration. For the comparison schemes, noticeable differences in broadcast counts among nodes can be observed, indicating uneven network load distribution. In contrast, the proposed ETATS scheme produces the most concentrated and uniform broadcast distribution across nodes.

To further assess the uniformity of triggering behavior, ten independent simulations were conducted, and the average number of broadcasts per node together with the corresponding standard deviation was computed for each scheme, as shown in Figure 5. Under the ETATS scheme, the broadcast counts exhibit the smallest standard deviation among all compared methods, with an average of approximately 45 broadcasts per node over 100 update periods.

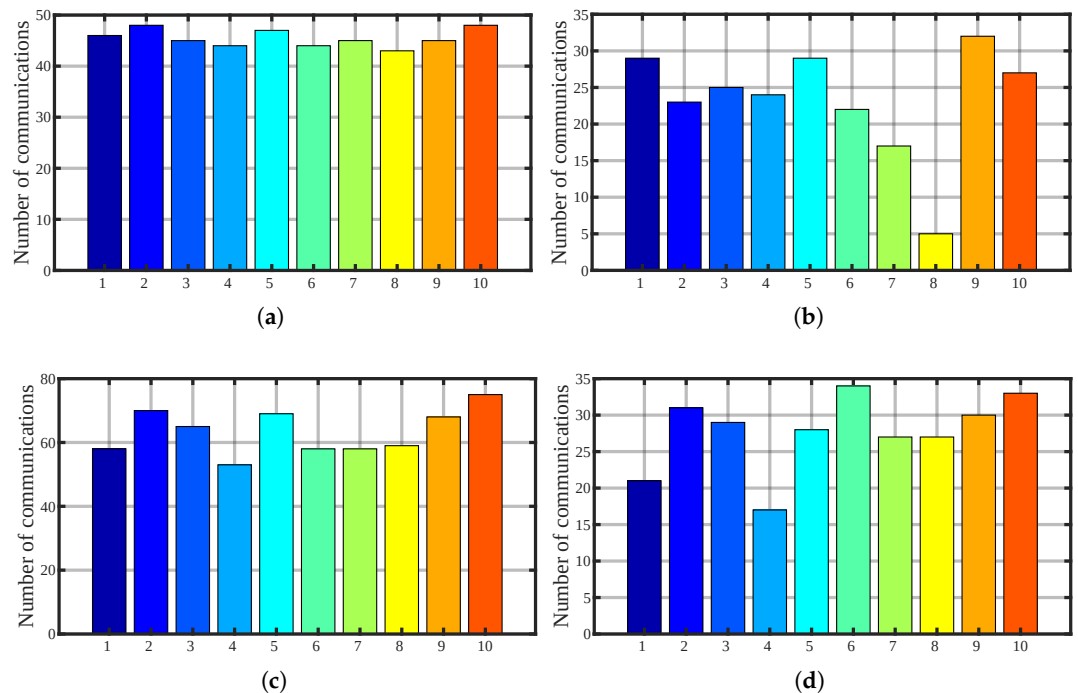


Figure 4. Distribution of broadcast counts per node under different schemes during the entire simulation period: (a) ETATS. (b) Constant-threshold scheme. (c) Exponential-threshold scheme. (d) Inverse-time-threshold scheme.

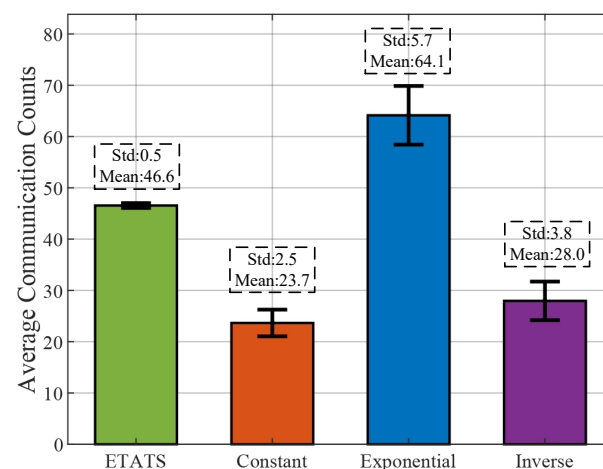


Figure 5. Average broadcast count per node over ten independent simulations.

Such balanced triggering behavior promotes more uniform energy consumption across the network and may contribute to extending the overall network lifetime. In addition, the evenly distributed broadcast counts reduce the probability of simultaneous channel access by multiple nodes. The simulation results show that the proposed ETATS scheme produces more balanced triggering patterns with moderate communication overhead.

5.5. Communication Efficiency

For a more comprehensive assessment of communication efficiency, synchronization accuracy under a fixed communication budget is considered. After running the simulations for 100 s, the synchronization accuracy achieved by different schemes under the same number of broadcast events is shown in Figure 6.

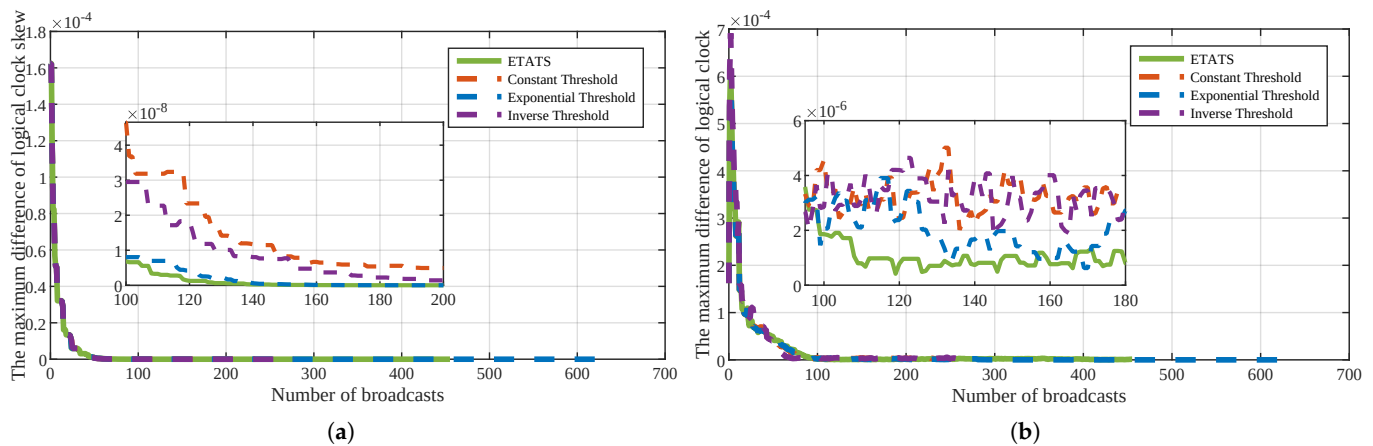


Figure 6. The logical clock synchronization error under an equal communication budget: (a) Logical clock skew. (b) Logical clock.

As illustrated in Figure 6a, the proposed ETATS scheme achieves lower logical clock skew errors than the constant-threshold and inverse-time threshold schemes under the same communication budget. This improvement mainly results from the adaptive threshold mechanism, which dynamically adjusts the triggering sensitivity according to the synchronization state and discrepancies among neighboring nodes, allowing ETATS to utilize communication opportunities more effectively while avoiding unnecessary transmissions.

As shown in Figure 6b, a similar trend can be observed for the logical clock offset error. Compared with the constant-threshold and inverse-time threshold schemes, ETATS achieves better synchronization accuracy under the same communication budget. Although the exponentially decaying threshold scheme eventually attains slightly higher accuracy in the later stage, it requires substantially more communication events. This is because the exponential decay gradually reduces the threshold to a very small value, causing frequent triggering once the measurement errors become larger than the threshold. In contrast, ETATS incorporates neighbor state information into the triggering decision and dynamically adjusts the triggering threshold according to the synchronization state. This enables a balanced trade-off between synchronization accuracy and communication cost. The other schemes rely on fixed or purely time-varying thresholds and therefore lack the flexibility to adapt to dynamic network states, leading either to reduced accuracy due to conservative communication or excessive communication overhead to achieve higher precision.

To further quantify communication efficiency, the time and the number of broadcast transmissions required for each scheme to reach the same global maximum logical clock error of 2×10^{-6} s are compared. This target error is used as a common accuracy criterion, so that the convergence speed and communication cost of different triggering mechanisms can be evaluated under the same synchronization requirement. Ten independent simulations are performed under identical network topology and algorithmic parameters. The averaged results are shown in Figure 7a.

As illustrated in Figure 7a, under the same synchronization accuracy requirement, the proposed ETATS scheme achieves the best overall performance among all compared algorithms. The average convergence time and the total number of broadcasts are 15.8 s and 96, respectively, indicating that ETATS provides higher communication efficiency when achieving the desired synchronization accuracy. Moreover, the error bars obtained from ten independent simulation runs indicate that ETATS exhibits smaller standard deviations in both convergence time and total number of broadcasts compared with the benchmark methods. This suggests that the proposed method not only improves the average performance

but also provides better statistical stability across different simulation runs. In contrast, the larger variability observed in the benchmark methods indicates that their performance is more sensitive to stochastic triggering behavior and network dynamics.

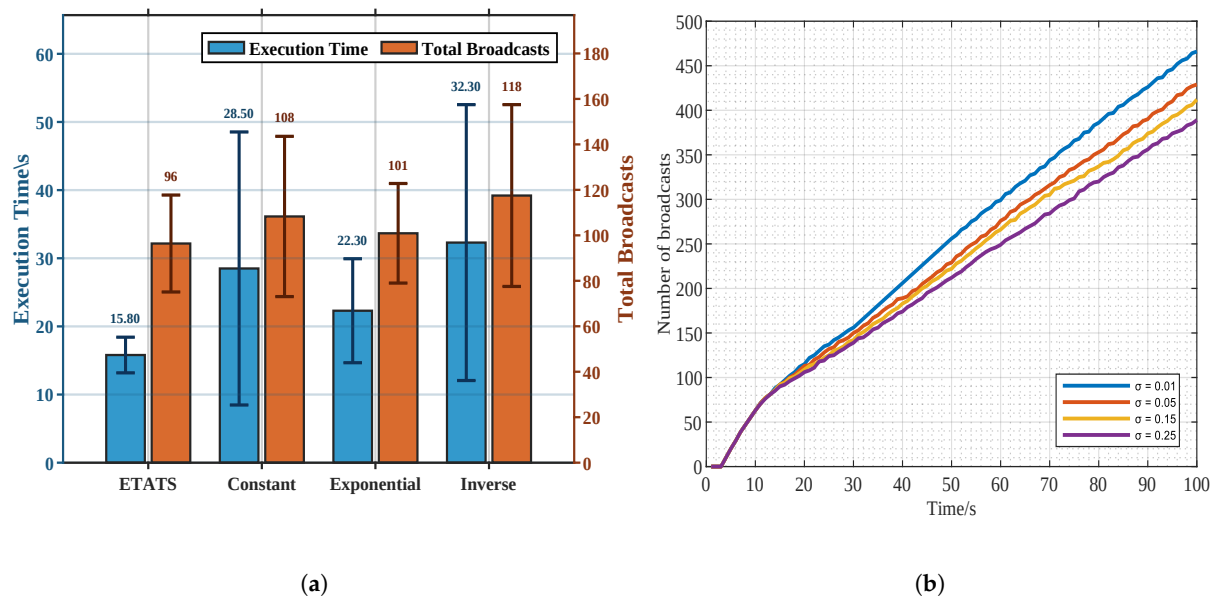


Figure 7. Additional performance comparison: (a) Number of broadcasts and time required to achieve the same global logical clock error under different schemes. (b) Impact of threshold sensitivity parameters on communication overhead.

Finally, the influence of the threshold sensitivity parameter σ on the performance of ETATS is investigated. As illustrated in Figure 7b, smaller sensitivity parameters increase communication overhead, whereas larger parameters significantly reduce communication load. Specifically, when the sensitivity parameter σ is set to 0.25, the number of broadcasts is 391. In contrast, when the sensitivity parameter is reduced to 0.01, the number of broadcasts increases to 465. This demonstrates negative correlation between the threshold sensitivity parameter and the communication load. These results are consistent with the theoretical analysis and confirm that ETATS provides explicit tunability to balance synchronization performance and communication efficiency, allowing for fine-tuned adjustments based on specific application requirements.

5.6. Scalability Analysis

To evaluate the scalability of the proposed ETATS scheme, additional simulations were conducted under random connected topologies with different network sizes. In addition to the original 10-node network, random connected networks with 30, 50, and 100 nodes were generated. For all network sizes, the hardware clock model, initialization ranges, update period, simulation duration, and triggering parameters were kept consistent with those in Section 5.1, so that the influence of network scale could be evaluated under comparable conditions.

Figure 8a shows the evolution of the maximum logical clock skew error under different network sizes. Although the initial synchronization deviation varies with the generated random topology and hardware clock initialization, the skew error decreases rapidly in all cases. This indicates that the proposed adaptive event-triggered mechanism is not restricted to the small-scale 10-node topology used in the baseline simulations, but can also maintain stable synchronization behavior when the number of nodes increases.

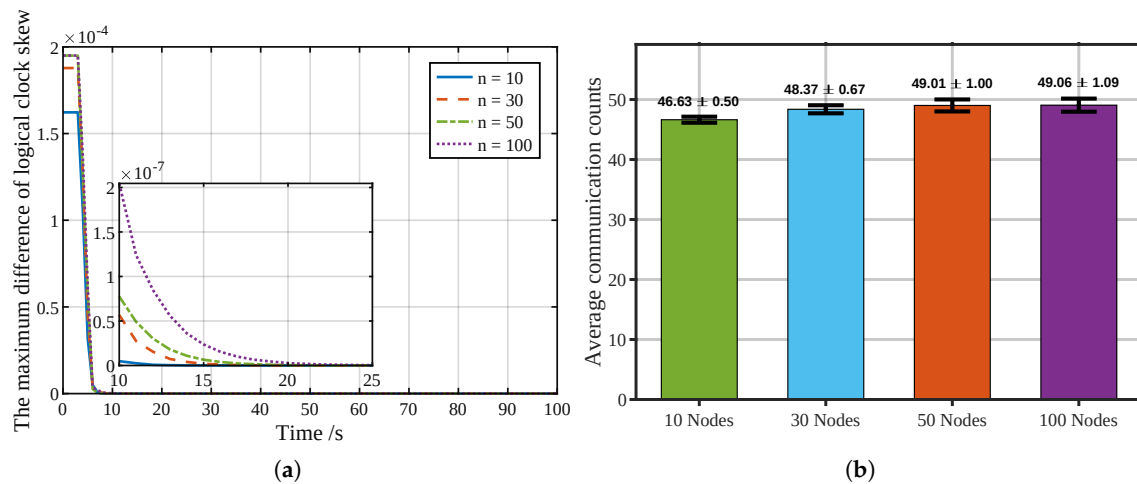


Figure 8. Scalability of ETATS with different network sizes: (a) Maximum logical clock skew error. (b) Average broadcast count per node.

Figure 8b shows the average broadcast count per node over ten independent simulations. The average numbers of broadcasts for 10, 30, 50, and 100 nodes remain close to each other, and the corresponding standard deviations are relatively small. These results indicate that increasing the network size does not lead to a severe concentration of triggering events on a small subset of nodes. Therefore, ETATS preserves a relatively balanced communication load across nodes while maintaining effective synchronization performance in larger random connected IIoT networks.

5.7. Robustness Analysis

To further evaluate the robustness of the proposed method under more realistic industrial communication conditions, several non-ideal scenarios were considered, including node failures, topology switching, random packet loss, and communication delay with measurement noise. Unless otherwise stated, the simulations were conducted on the 10-node network used in Section 5.1 with a total simulation time of 100 s.

First, a node failure scenario was considered. In this simulation, nodes 2, 5, and 8 left the network at different times and stopped broadcasting their clock parameters. As shown in Figure 9, after the faulty nodes leave the network, the remaining normal nodes still maintain stable synchronization. The maximum errors of logical clock skew do not exhibit sustained growth or divergence. This result indicates that ETATS can tolerate partial node failures under the considered connected residual-network condition.

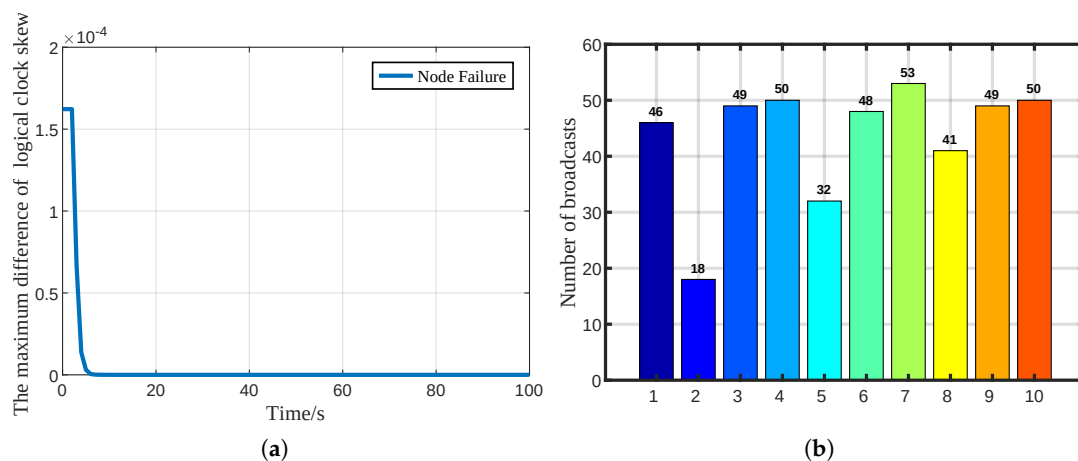


Figure 9. Robustness of ETATS under node failures: (a) Maximum logical clock skew error. (b) Distribution of broadcast counts per node.

Then, a topology switching scenario was designed to simulate temporary link failures and recoveries caused by obstacles, electromagnetic interference, or link quality fluctuations in industrial wireless environments. An initial connected topology with 10 nodes was first generated. The topology was then switched at 20 s, 45 s, 80 s, and 85 s by randomly removing a proportion of links from the original edge set while checking that the resulting topology remained connected. The link removal ratios at the four switching instants were set to 25, 35, 40, and 45%, respectively, to emulate relatively strong link-quality fluctuations.

As shown in Figure 10, the logical clock synchronization error may exhibit short-term fluctuations around the topology switching instants, indicating that changes in neighboring relationships perturb the local synchronization update process. Meanwhile, the broadcast counts may slightly increase around some switching intervals, suggesting that several nodes need additional broadcasts to readjust their synchronization relationships with neighbors. Nevertheless, because the topology remains connected in each stage, the error curves return to stable convergence after a short adjustment period, and no persistent divergence or long-term loss of synchronization is observed. The broadcast counts among nodes also show no obvious long-term concentrated triggering or long-term silence, demonstrating that the proposed event-triggered synchronization method has a certain degree of adaptability under strongly disturbed connected topology switching.

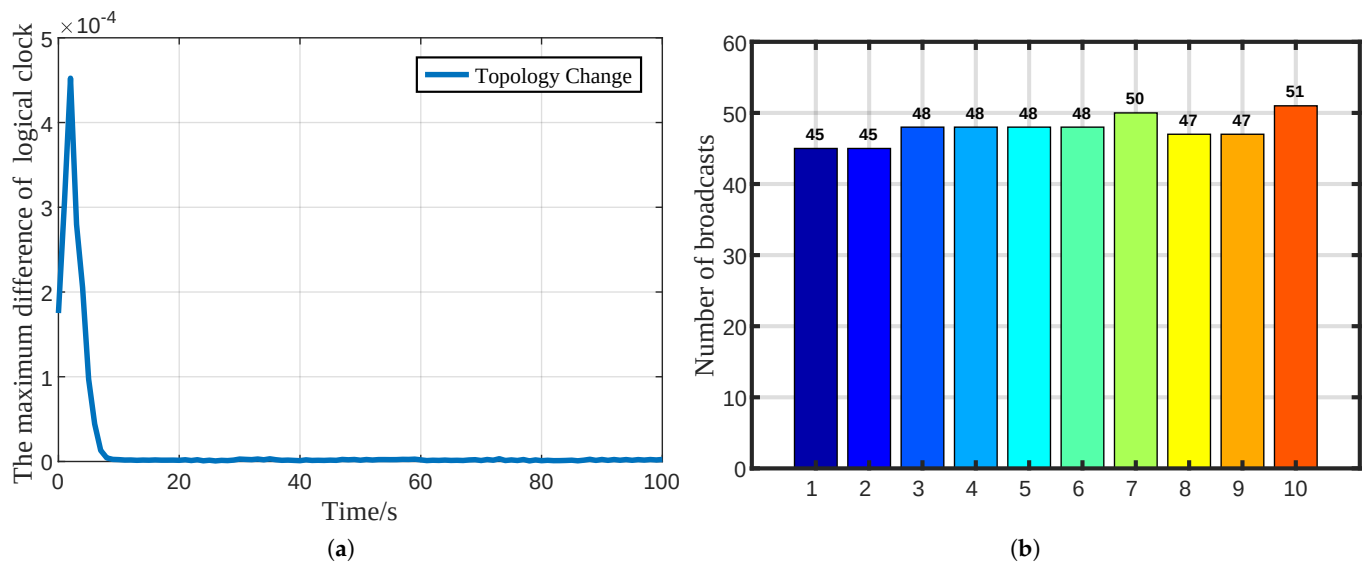


Figure 10. Robustness of ETATS under topology change: (a) Maximum logical clock error. (b) Distribution of broadcast counts per node.

For packet loss scenario, packet-loss probabilities of 20, 40, and 60% were considered. As shown in Figure 11a, both convergence speed and steady-state synchronization accuracy are affected as the packet-loss probability increases. Nevertheless, the network can still maintain synchronization within the tested packet-loss range. Figure 11b further shows the broadcast count of each node when the packet-loss probability is 40%. The communication load remains relatively balanced among nodes, indicating that random packet loss does not cause severe triggering-load concentration in the tested scenario. These results show that ETATS has a certain degree of robustness to medium-to-high random packet loss, although reduced communication reliability inevitably degrades synchronization performance.

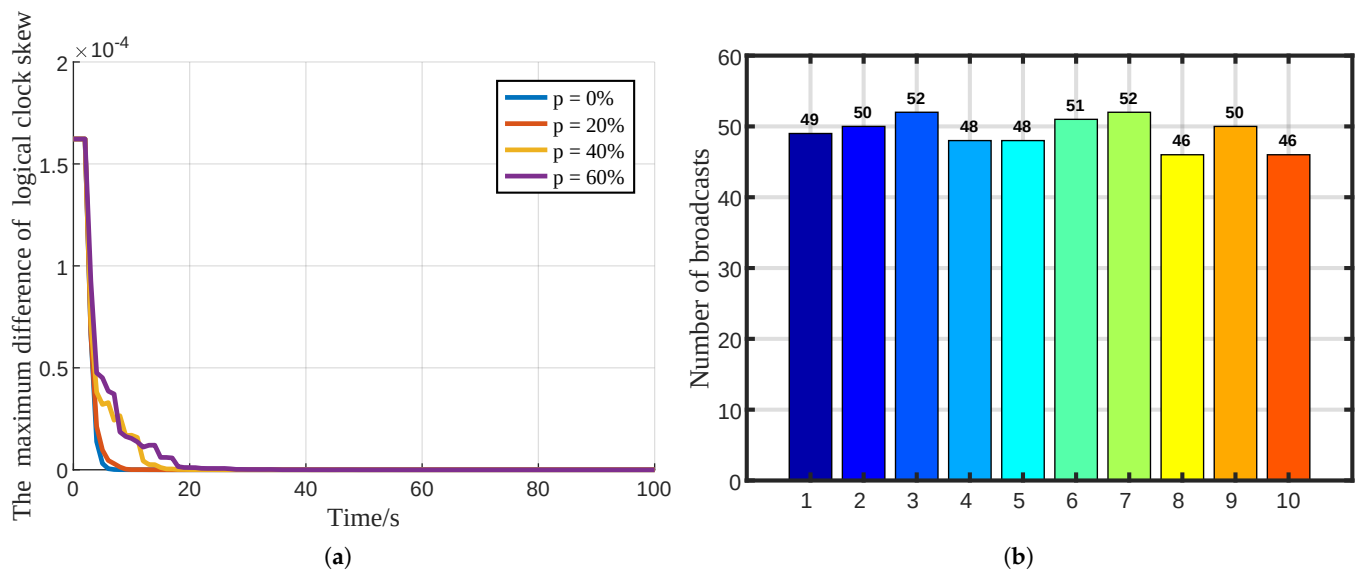


Figure 11. Robustness of ETATS under random packet loss: (a) Maximum logical clock skew error under different packet loss probabilities. (b) Distribution of broadcast counts per node when the packet loss probability is 40%.

Finally, communication delay was investigated. In the simulation, the communication delay is modeled as an independent and identically distributed Gaussian random variable ($d \sim \mathcal{N}(\mu_d, \sigma_d^2)$), with $\mu_d = 2.5 \times 10^{-4}$ s and $\sigma_d^2 = 10^{-8}$ s². Since communication delay directly perturbs timestamp-based relative clock skew estimation, accumulated estimation errors may degrade synchronization performance if they are not properly handled. Therefore, a stochastic-approximation-based estimation method was adopted to reduce the influence of delay, i.e., $\hat{a}_{ij}(t_k^i) = \left[\left(\tau_j(t_k^i) - \tau_j(t_0^i) \right) / \left(\tau_i(t_k^{ij}) - \tau_i(t_0^{ij}) \right) + (k-1)\hat{a}_{ij}(t_{k-1}^i) \right] / k$, where $\tau_j(t_0^i)$ denotes the time at which node j first broadcast to node i . $\hat{a}_{ij}(t_k^i)$ denotes the modified estimated value at t_k^i .

As shown in Figure 12, compared with ETATS, the stochastic-approximation-based implementation achieves more stable synchronization under communication delays, indicating that the large-span timestamp estimation method can effectively reduce the influence of delay-induced estimation errors.

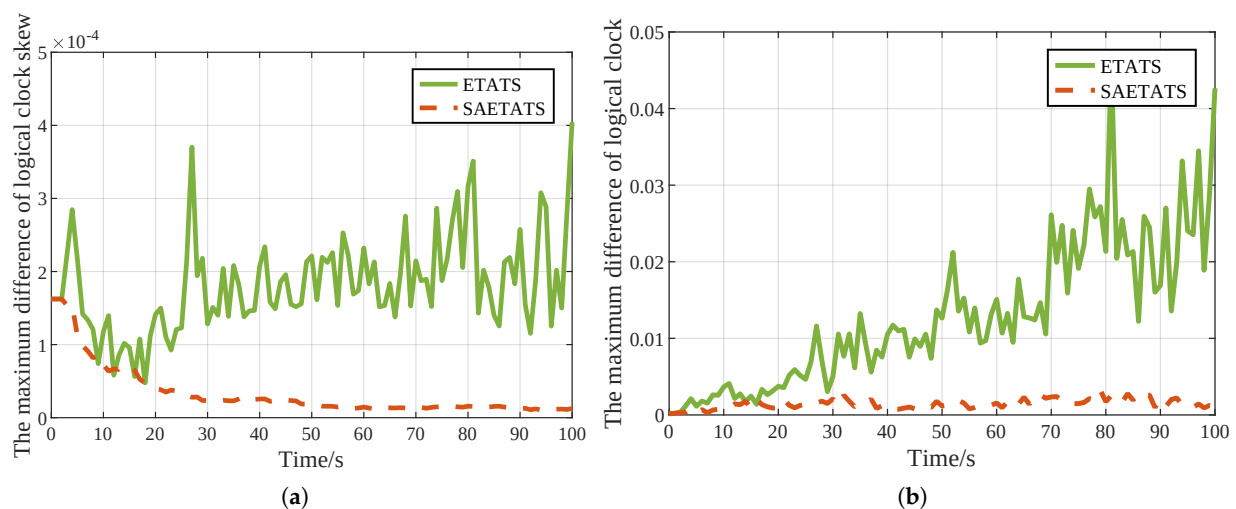


Figure 12. Synchronization performance of ETATS under communication delay: (a) maximum logical clock skew error; (b) maximum logical clock error.

Overall, the above robustness experiments further verify the performance of ETATS under several representative non-ideal industrial communication conditions. More complicated scenarios, such as persistent disconnection, strongly correlated or bursty packet loss, non-Gaussian delay, network partitioning, and frequent large-scale node joining or leaving, will be studied in future work.

6. Conclusions

This study addressed the problem of communication-efficient time synchronization for Industrial Internet of Things (IIoT) networks and proposed an adaptive-threshold event-triggered synchronization scheme termed ETATS. Unlike conventional event-triggered approaches that rely on fixed or purely time-varying thresholds, the proposed strategy introduces a neighbor-aware dynamic triggering mechanism that jointly exploits locally accumulated clock deviations and synchronization discrepancies observed from neighboring nodes. This mechanism allows the triggering process to adapt to the instantaneous synchronization state of the network and improves the efficiency of communication utilization. Theoretical analysis shows that the proposed scheme guarantees consensus of logical clock skews and ensures that the logical clock offset disagreement converges to a bounded neighborhood. Simulation results demonstrate that ETATS achieves faster convergence and significantly improves communication efficiency compared with several representative event-triggered synchronization schemes. In particular, the proposed method effectively reduces redundant broadcast transmissions while maintaining competitive synchronization accuracy, which is highly desirable for resource-constrained IIoT systems. Future work will extend the proposed framework to more realistic industrial network environments, including communication delays, packet losses, and dynamically changing network topologies.

Author Contributions: Conceptualization, Z.W. and L.Z.; methodology, Z.W. and L.Z.; software, L.Z.; validation, Z.W. and L.Z.; formal analysis, L.Z.; investigation, L.Z.; resources, Z.W.; data curation, L.Z.; writing—original draft preparation, Z.W. and L.Z.; writing—review and editing, Z.W. and L.Z.; visualization, L.Z.; supervision, Z.W.; project administration, Z.W.; funding acquisition, Z.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Pond Ecological Aquaculture Green Intelligent Equipment Technology and Control System Integration Demonstration Project, grant number JSYTH03.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgments: The authors would like to thank the reviewers for their constructive comments.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Qiu, F.; Kumar, A.; Hu, J.; Sharma, P.; Tang, Y.B.; Xu, X.Y.; Hong, J. A review on integrating IoT, IIoT, and Industry 4.0: A pathway to smart manufacturing and digital transformation. *IET Inf. Secur.* **2025**, *2025*, 9275962. [[CrossRef](#)]
2. Wang, X.; Jia, P.; Shen, X.; Poor, H.V. Intelligent and low overhead network synchronization for large-scale industrial IoT systems in the 6G era. *IEEE Netw.* **2023**, *37*, 76–84. [[CrossRef](#)]
3. Wang, H.; Zhong, Y.; Qian, P.; Wang, P. A cross-network clock synchronization scheme with low complexity for industrial Internet of Things. In *Proceedings of the 2018 Chinese Automation Congress (CAC), Xi'an, China, 30 November–2 December 2018*; IEEE: Piscataway, NJ, USA, 2018; pp. 1758–1763.

4. Tang, L.; Li, F.; Yu, Z.; Zeng, H. Noise-robust-based clock parameter estimation and low-overhead time synchronization in time-sensitive industrial Internet of Things. *Entropy* **2025**, *27*, 927. [[CrossRef](#)] [[PubMed](#)]
5. Shi, F.; Tuo, X.; Yang, S.X.; Lu, J.; Li, H. Rapid-flooding time synchronization for large-scale wireless sensor networks. *IEEE Trans. Ind. Inform.* **2020**, *16*, 1581–1590. [[CrossRef](#)]
6. Shi, F.; Li, H.; Yang, S.X.; Tuo, X.; Lin, M. Novel maximum likelihood estimation of clock skew in one-way broadcast time synchronization. *IEEE Trans. Ind. Electron.* **2020**, *67*, 9948–9957. [[CrossRef](#)]
7. Sommer, P.; Wattenhofer, R. Gradient clock synchronization in wireless sensor networks. In *Proceedings of the International Conference on Information Processing in Sensor Networks, San Francisco, CA, USA, 13–16 April 2009*; IEEE Computer Society: Washington, DC, USA, 2009; pp. 37–48.
8. Schenato, L.; Fiorentin, F. Average timesynch: A consensus-based protocol for clock synchronization in wireless sensor networks. *Automatica* **2011**, *47*, 1878–1886. [[CrossRef](#)]
9. He, J.; Cheng, P.; Shi, L.; Chen, J.; Sun, Y. Time synchronization in WSNs: A maximum-value-based consensus approach. *IEEE Trans. Autom. Control* **2014**, *59*, 660–675. [[CrossRef](#)]
10. Wang, Z.; Yong, T.; Song, X.; Sun, D. Maximum likelihood estimation of relative clock skew for distributed time synchronization in industrial wireless sensor networks. *IEEE Sens. J.* **2023**, *23*, 31359–31367. [[CrossRef](#)]
11. Elsharief, M.; Emran, A.; Hassan, H.; Sabuj, S.R.; Jo, H.-S. Energy-efficient synchronization in industrial Internet of Things: An intelligent neighbor-knowledge approach. *IEEE Trans. Ind. Inform.* **2024**, *20*, 8548–8558. [[CrossRef](#)]
12. Cappelle, J.; Goossens, S.; De Strycker, L.; Van der Perre, L. Low-power synchronization for multi-IMU WSNs. *IEEE Embed. Syst. Lett.* **2024**, *16*, 210–213. [[CrossRef](#)]
13. Portillo, C.; Martinez-Bauset, J.; Pla, V.; Casares-Giner, V. Energy consumption modeling for heterogeneous Internet of Things wireless sensor network devices. *Telecom* **2024**, *5*, 723–746. [[CrossRef](#)]
14. Kumar, R.; Irfan, B.M.; Soni, U.; BK, M.; Maranan, R. Smart energy management leveraging twin adaptive pulse coupled networks for dynamic energy optimization in IoT-based electrical WSN. In *Proceedings of the International Conference on Trends in Material Science and Inventive Materials*; IEEE: Piscataway, NJ, USA, 2025; pp. 401–407.
15. Nowzari, C.; Cortés, J. Distributed event-triggered coordination for average consensus on weight-balanced digraphs. *Automatica* **2016**, *68*, 237–244. [[CrossRef](#)]
16. Meng, X.; Chen, T. Event-based agreement protocols for multi-agent networks. *Automatica* **2013**, *49*, 2125–2132. [[CrossRef](#)]
17. Ding, L.; Han, Q.-L.; Ge, X.; Zhang, X.-M. An overview of recent advances in event-triggered consensus of multiagent systems. *IEEE Trans. Cybern.* **2018**, *48*, 1110–1123. [[CrossRef](#)] [[PubMed](#)]
18. Cheng, B.; Li, Z. Fully distributed event-triggered protocols for linear multiagent networks. *IEEE Trans. Autom. Control* **2019**, *64*, 1655–1662. [[CrossRef](#)]
19. Kadowaki, Y.; Ishii, H. Event-based distributed clock synchronization for wireless sensor networks. In *Proceedings of the 52nd IEEE Conference on Decision and Control, Florence, Italy, 10–13 December 2013*; IEEE: Piscataway, NJ, USA, 2013; pp. 6747–6752.
20. Chen, Z.; Li, D.; Huang, Y.; Tang, C. Event-triggered communication for time synchronization in WSNs. *Neurocomputing* **2016**, *177*, 416–426. [[CrossRef](#)]
21. García, E.; Mou, S.; Cao, Y.; Casbeer, D.W. An event-triggered consensus approach for distributed clock synchronization. In *Proceedings of the American Control Conference, Seattle, WA, USA, 24–26 May 2017*; IEEE: Piscataway, NJ, USA, 2017; pp. 279–284.
22. Berneburg, J.A.; García, E.; Gerlach, A.R.; Casbeer, D.W.; Nowzari, C. Strongly non-zeno event-triggered wireless clock synchronization. *IFAC-PapersOnLine* **2020**, *53*, 2745–2750. [[CrossRef](#)]
23. Jia, Z.; Dai, X.; Cui, D. Observer-based output feedback event-triggered control for discrete-time synchronization. In *Proceedings of the 2022 China Automation Congress (CAC), Xiamen, China, 25–27 November 2022*; IEEE: Piscataway, NJ, USA, 2022; pp. 6104–6109.
24. Jia, Z.; Dai, X.; Cui, D.; Hu, Y. Asynchronous broadcast-based event-triggered control for discrete-time clock synchronization. *IET Control Theory Appl.* **2023**, *17*, 1543–1551. [[CrossRef](#)]
25. Wang, H.; Zhang, N.; Liu, X.; Zou, Y.; Li, M. A simple event-based average consensus clock synchronization scheme in industrial wireless sensor networks under communication delays. *IEEE Trans. Ind. Inform.* **2024**, *20*, 10113–10122. [[CrossRef](#)]
26. Zhao, H.; Dai, X.; Zhao, Y. An event-triggered synchronization strategy for high-precise clock systems with switching topologies. *Heliyon* **2024**, *10*, e36755. [[CrossRef](#)] [[PubMed](#)]
27. Lou, G.; Hong, Y.; Gu, W.; Xie, J. Event-triggered time synchronization strategy for distributed secondary control considering time-varying delays. *CSEE J. Power Energy Syst.* **2025**, *11*, 1118–1130.
28. Wang, H.; Zhu, X.; Liu, X.; Zou, Y.; Tan, T.; Li, M. Event-triggered time synchronization with varying skews for wireless sensor networks based on average consensus. *IEEE Trans. Netw. Sci. Eng.* **2025**, *12*, 2895–2906. [[CrossRef](#)]

29. Jia, P.; Wang, X.; Shen, X. Digital-twin-enabled intelligent distributed clock synchronization in industrial IoT systems. *IEEE Internet Things J.* **2021**, *8*, 4548–4559. [[CrossRef](#)]
30. Rodrigues, S.; Lv, J. Synchronization in time-sensitive networking: An introduction to IEEE Std 802.1AS. *IEEE Commun. Stand. Mag.* **2022**, *6*, 14–20. [[CrossRef](#)]
31. Abakasanga, E.; Shlezinger, N.; Dabora, R. Unsupervised deep-learning for distributed clock synchronization in wireless networks. *IEEE Trans. Veh. Technol.* **2023**, *72*, 12234–12247. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.