Wireless Clock Synchronization: A Comprehensive Survey and Taxonomy

By MOHAMED SELIEM, *Member IEEE*, UTZ ROEDIG, *Member IEEE*, MAHIN AHMED, *Member IEEE*, RAHEEB MUZAFFAR, *Member IEEE*, DAMIR HAMIDOVIC, *Member IEEE*, ARMIN HADZIAGANOVIC, *Member IEEE*, CORMAC J. SREENAN, *Senior Member IEEE*, AND DIRK PESCH, *Senior Member IEEE*

ABSTRACT | Precise clock synchronization underpins deterministic operation in wireless systems spanning industrial automation, vehicular networks, distributed extended reality (XR), smart infrastructure, and wide-area precision agriculture. Wireless links introduce variable propagation delays, channel asymmetry, interference, clock drift, and scalability constraints that make sub-microsecond alignment difficult. This article provides a comprehensive survey and tutorial on wireless clock synchronization. We introduce a five-dimension taxonomy covering: system architecture, synchronization mechanism, correction strategy, delay and uncertainty modeling, and resource and deployment class, and apply it to eight canonical protocol families and to synchronization as realized across IEEE 802.15.4, ZigBee, Bluetooth low energy (BLE), LoRa, Wi-Fi, Ultrawide band (UWB), and 4G/5G/6G systems. We examine solutions across five application domains: industrial automation and Industrial Internet of Things (IIoT), vehicular V2X, distributed XR and metaverse, infrastructure monitoring, and wide-area Internet of Things (IoT) and precision agriculture; alongside tools, testbeds, and datasets supporting evaluation. Open challenges addressed include scalability and mobility, ultralow-jitter determinism, robust clock parameter estimation under non-Gaussian delay distributions, distributed and consensus-based synchronization for infrastructure-free networks, secure and resilient synchronization against wireless-specific threats, and cross-domain convergence encompassing time-sensitive networking (TSN)–5G/6G interoperability and joint communication, sensing, and timing as an emerging 6G design paradigm. Together, these contributions provide the first unified cross-technology framework connecting fundamentals, protocol families, application domains, and open research challenges in wireless clock synchronization.

KEYWORDS | 5G/6G; clock synchronization; distributed systems; time-sensitive networking (TSN); wireless sensor networks (WSNs); wireless synchronization.

Received 15 October 2025; revised 25 March 2026; accepted 15 June 2026. This work was supported by Taighde Éireann—Research Ireland under Grant 13/RC/2077_P2. *(Corresponding author: Mohamed Seliem.)*

Mohamed Seliem, **Utz Roedig**, **Cormac J. Sreenan**, and **Dirk Pesch** are with the School of Computer Science and Information Technology, Western Gateway Building, University College Cork, T12 XF62 Cork, Ireland (e-mail: mseliem@ucc.ie; u.roedig@ucc.ie; cormac.sreenan@ucc.ie; dirk.pesch@ucc.ie).
Mahin Ahmed, **Raheeb Muzaffar**, **Damir Hamidovic**, and **Armin Hadziaganovic** are with Silicon Austria Labs, 4040 Linz, Austria (e-mail: mahin.ahmed@silicon-austria.com; raheeb.muzaffar@silicon-austria.com; damir.hamidovic@silicon-austria.com; Armin.Hadziaganovic@silicon-austria.com).

Digital Object Identifier 10.1109/JPROC.2026.3705899

Nomenclature

6TiSCH	IPv6 over the TSCH.
AGV	Automated guided vehicle.
AI	Artificial intelligence.
AP	Access point.
AR	Augmented reality.
BLE	Bluetooth low energy.
BSS	Basic service set.

C-V2X	Cellular V2X.
CSMA/CA	Carrier sense multiple access with collision avoidance.
DS-TT	Device-side TSN translator.
FTSP	Flooding time synchronization protocol.
GCL	Gates control list.
GM	Grandmaster.
GNSS	Global Navigation Satellite System.
GPS	Global Positioning System.
gPTP	Generalized precision time protocol.
GTSP	Gradient time synchronization protocol.
HARQ	Hybrid automatic repeat request.
IIoT	Industrial Internet of Things.
IoT	Internet of Things.
ISAC	Integrated sensing and communication.
JCST	Joint communication sensing and timing.
LPWANs	Low-power wide-area networks.
LTE	Long term evolution.
ML	Machine learning.
NR	New radio.
NTP	Network time protocol.
NW-TT	Network-side TSN translator.
OFDMA	Orthogonal frequency-division multiple access.
PCO	Pulse-coupled oscillator.
PMU	Phasor measurement unit.
PPM	Parts per million.
PPS	Pulse per second.
PTP	Precision time protocol.
RBS	Reference broadcast synchronization.
RL	Reinforcement learning.
TDoA	Time difference of Arrival.
TPSN	Timing-sync protocol for sensor networks.
TSC	Time-sensitive communication.
TSCH	Time-slotted channel hopping.
TSF	Timing synchronization function.
TSN	Time-sensitive networking.
TWR	Two-way ranging.
UE	User equipment.
URLLCs	Ultrareliable low-latency communications.
UTC	Universal time coordinated.
UWB	Ultrawide band.
V2V	Vehicle to vehicle.
V2X	Vehicle to everything.
VR	Virtual reality.
WSN	Wireless sensor network.
XR	Extended reality.

I. INTRODUCTION

Time synchronization allows devices in a distributed system to agree on a common notion of time. This shared reference is essential for coordinating transmissions, combining sensor measurements, scheduling control actions, and enabling coherent sensing across distributed nodes. In wired networks, protocols like the NTP [9] and the PTP [10] have provided widely used synchronization solutions, enabling reliable coordination in enterprise infrastructures, financial systems, and industrial automation. Wireless networks, however, present a fundamentally different set of challenges: variable propagation delays, multipath fading, interference, link asymmetry, and energy constraints collectively prevent direct application of wired synchronization approaches. At scale, these effects make it far more difficult to maintain sub-microsecond time accuracy across distributed wireless devices, as demanded by modern real-time applications spanning industrial control, vehicular safety, and immersive communications.

The demand for precise wireless synchronization is expanding across a broad set of domains. IIoT deployments require tightly synchronized sensors and actuators to sustain scheduled communication and real-time control; a 10-μs timing error, for instance, can cause collisions in a TSN gate schedule and jeopardize robotic actuation on a factory floor. Vehicular and cooperative robotic systems rely on shared timing for safety-critical maneuvers such as platooning and intersection coordination, where intervehicle synchronization within 1 ms is a baseline requirement. In parallel, 5G and emerging 6G networks are extending TSN concepts into the wireless domains, where deterministic sub-microsecond synchronization is a design prerequisite for URLLC and TSC [8], [11]. Beyond these established verticals, new domains are asserting equally stringent requirements: distributed XR and haptic tele-operation demand interdevice alignment below 100 μs to prevent perceptible timing artifacts; edge AI pipelines, spanning multicamera inspection, predictive maintenance, and neuromorphic sensing, require millisecond-to-microsecond coherence across heterogeneous sensor streams; wide-area IoT deployments for precision agriculture and environmental monitoring demand lightweight, energy-efficient timing over kilometer-scale links; and civil or utility infrastructure monitoring, including structural health monitoring and distributed power-grid sensing via PMUs, depends on synchronized measurements across geographically dispersed nodes. In each case, the gap between the sub-microsecond precision achievable over wired TSN and the microsecond-to-millisecond accuracy typical of current wireless deployments remains significant, underscoring the need for further advances in wireless synchronization.

Wireless clock synchronization has consequently emerged as a cross-cutting enabler across diverse application domains, as illustrated in Fig. 1. In WSNs, IIoT, vehicular systems, XR, edge AI, smart grids, wide-area sensing, and infrastructure monitoring, requirements for precision, scalability, robustness, and security are all intensifying. Yet the field remains fragmented: existing surveys address individual technology families or narrow design dimensions, and none provides a unified framework connecting fundamentals, protocol families, application domains, open challenges, and emerging paradigms such as distributed consensus-based synchronization and JCST. Table 1 highlights this gap by mapping eight representative prior surveys across 11 coverage dimensions. For example, works such as [7] and [8] provide thorough treatment of

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Table 1 Comparison of Prior Survey Articles on Clock Synchronization. • = Full Coverage; ○ = Partial Coverage; – = Not Addressed

Reference	Dimension of Coverage										
	WSN / IoT sync.	*IIoT / TSN*	*5G/6G TSC*	*UWB*	*Unified taxonomy*	*Consensus / distributed*	*Robust estimation*	*ML-assisted sync.*	*JCST*†	*Security*	*Tools / testbeds*
Sundararaman et al. (2005) [1]	•	–	–	–	–	–	–	–	–	–	○
Dong & Liu (2012) [2]	•	–	–	–	–	–	–	–	–	–	–
Ji et al. (2018) [3]	○	–	–	○	–	–	○	–	–	–	–
Wang et al. (2022) [4]	•	○	–	–	–	–	–	•	–	–	–
Dang et al. (2023) [5]	○	•	○	–	–	–	–	–	–	–	–
Balakrishnan et al. (2023) [6]	○	•	○	–	–	–	–	–	–	–	○
Morato et al. (2024) [7]	–	•	•	–	–	–	–	–	–	–	○
Sasiain et al. (2024) [8]	–	•	•	–	–	–	–	–	–	–	○
This article (2026)	•	•	•	•	•	•	•	•	•	•	•

†JCST: Joint Communication, Sensing, and Timing (§VIII-E2).

5G–TSN integration but leave UWB, consensus algorithms, robust estimation, JCST, and security unaddressed. Wang et al. [4] introduced ML-assisted synchronization but lack a cross-technology taxonomy and omit IIoT and cellular domains. Likewise, Dang et al. [5] and Balakrishnan et al. [6] covered IIoT in depth but do not extend to UWB, ML-assisted control, or emerging 6G concepts. This article fills these gaps by providing a unified, cross-technology survey and taxonomy.

This article focuses specifically on wireless clock synchronization protocols and techniques, with an emphasis on their underlying principles, architectures, and performance tradeoffs. We review relevant standards and frameworks, including PTP, gPTP, and the broader IEEE 802.1 TSN framework, alongside experimental tools, testbeds, and datasets. We do not provide a full treatment of wired synchronization protocols, oscillator hardware design, or localization methods, except where they are directly relevant to wireless clock synchronization. The main contributions of this article are as follows.

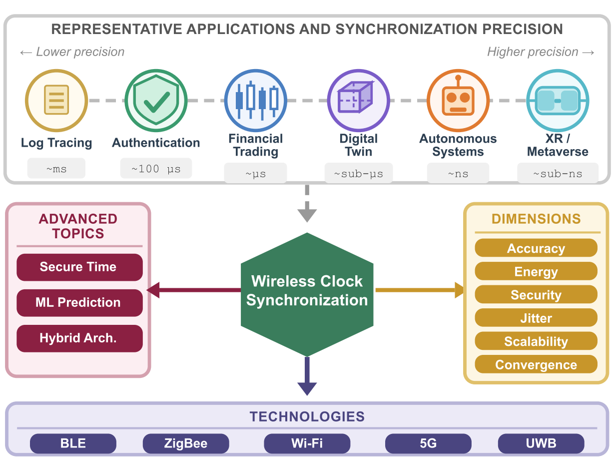

Fig. 1. *Survey coverage: application domains and precision requirements (top), wireless technologies examined (bottom), and key performance dimensions and advanced topics (center).*

A. Unified Cross-Technology Framework for Wireless Clock Synchronization

We propose a five-dimension taxonomy (system architecture, synchronization mechanism, correction strategy, delay and uncertainty modeling, and resource and deployment class) and apply it consistently across eight canonical protocol families and seven wireless technology domains (IEEE 802.15.4, ZigBee, BLE, LoRa/NB-IoT, Wi-Fi, UWB, and 4G/5G/6G TSC). The taxonomy exposes cross-protocol relationships that fragmented, technology-specific surveys obscure and provides a principled basis for protocol selection across heterogeneous deployments (see Sections III–V).

B. Cross-Domain Synthesis of Synchronization Requirements and Solutions

We demonstrate that timing requirements diverge fundamentally across application domains, from sub-100 μs haptic alignment in XR to second-level event timestamping in wide-area precision agriculture, and that no single protocol architecture satisfies all constraints simultaneously. By analyzing industrial automation and IIoT, vehicular V2X, distributed XR, infrastructure monitoring across smart grids and sensor networks, and wide-area IoT, we provide the first unified account of how protocol choices must be driven by application-specific timing budgets (see Section VI).

C. Distributed and Consensus-Based Synchronization for Infrastructure-Free Networks

We provide the first survey-level treatment of PCO schemes, average-consensus and diffusion algorithms, and

Byzantine-resilient protocols as a coherent family of infrastructure-free synchronization methods. No prior survey in Table 1 addresses this family, despite its critical role in mobile, ad hoc, and safety-critical deployments (see Section VIII-B).

D. Robust Clock Estimation and JCST as New Technical Frontiers

We analyze M-estimators, composite particle filters, and expectation–maximization methods for offset and skew estimation under the heavy-tailed and asymmetric delay distributions that characterize real wireless links, addressing a gap in both the literature and in this article's original submission. We further define JCST as a 6G design paradigm distinct from ISAC, one in which timing precision is co-optimized with sensing and communication from the outset rather than treated as a peripheral constraint (see Sections VIII-A4 and VIII-E2).

E. Structured Analysis of Open Challenges With an Opinionated Outlook

Rather than cataloging open problems, we argue that three challenges are structurally neglected across the field: security, which must become a first-class design constraint rather than a post hoc addition; benchmarking discipline, where evaluations must move beyond idealized simulations to reproducible hybrid testbeds; and architectural thinking, where rigid master–slave hierarchies must give way to federated, adaptive time fabrics capable of sustaining determinism across heterogeneous wired–wireless–satellite domains (see Section VIII).

The remainder of this article is organized as follows. Section II covers clock models, performance metrics, and generic synchronization architectures. Section III presents the five-dimension taxonomy. Section IV-A surveys canonical and representative protocol families. Section V examines synchronization across wireless technologies. Section VI covers five application domains. Section VII surveys tools, testbeds, and datasets. Section VIII analyses open challenges and future directions. Section IX concludes with an architectural outlook.

II. BACKGROUND AND FUNDAMENTALS

This section introduces the essential background required to understand wireless clock synchronization. We first describe the basic clock model and the common sources of deviation. Next, we present the key performance metrics used to evaluate synchronization protocols. Finally, we introduce the generic building blocks of synchronization systems as a conceptual foundation for the taxonomy developed in Section III.

A. Clock Models: Offset, Skew, and Perturbations

A clock comprises an oscillator and a counter. The oscillator generates a periodic signal with a nominal frequency that defines the time base, while the counter tracks the number of cycles elapsed. Using the cycle duration and the initial counter value, a clock can estimate the current UTC time by accumulating elapsed cycles. In practice, however, oscillator imperfections degrade frequency accuracy and stability. Aging, temperature variation, vibration, supply-voltage fluctuation, and counter quantization all contribute to deviations from ideal behavior [12], [13], [14]. Hence, regular synchronization is necessary to maintain the required timing performance.

Let t be the ideal reference time and $C(t)$ be the reading of a local clock with respect to that reference. A common first-order affine clock model is [15], [16], [17], [18]

$$C(t) = \theta + \rho t + \epsilon(t) \tag{1}$$

where θ is the clock offset, ρ is the clock skew, and $\epsilon(t)$ is a residual time-varying perturbation term. Fig. 2 illustrates these components: offset appears as a vertical displacement, skew as a slope mismatch, and $\epsilon(t)$ as a nonideal deviation from the nominal affine trajectory. The figure also distinguishes intrinsic clock behavior from extrinsic synchronization uncertainty by showing path-delay uncertainty during timestamp exchange as a separate shaded band.

1) **Offset** (θ) is the instantaneous time difference between a local clock and a reference clock. For example, if node A reads *10:00:00* while node B reads *10:00:05*, their offset is 5 s.
2) **Skew** (ρ) is the relative rate mismatch between the local clock and the reference clock. A skew of 50 PPM, typical of low-cost oscillators, causes a clock to gain or lose 50 μs/s, which accumulates to 50 ms over 1000 s.
3) **Residual perturbation** $[\epsilon(t)]$ captures deviations from the affine model due to oscillator instability, quantization effects, and environmental influences such as temperature and voltage fluctuations. Depending on the time scale, these deviations may appear as slow wander or drift-like behavior, or as short-term jitter.

Client or slave nodes compensate for offset and estimate skew in order to align their local clocks with that of the reference clock. This decomposition highlights why synchronization cannot be treated as a one-time offset correction. Even if two clocks are perfectly aligned at one instant, residual skew and time-varying perturbations will cause them to diverge again over time. To counteract this drift, a *synchronization event* at time t_s denotes an instantaneous correction applied to the local clock to realign it with the reference, as illustrated in Fig. 2. For example, two oscillators differing by only 20 PPM accumulate approximately 72 ms of error after 1 h if left unsynchronized, underscoring the need for periodic synchronization events. These effects directly govern the precision requirements introduced in Section II-B and the protocol design choices analyzed in Section III.

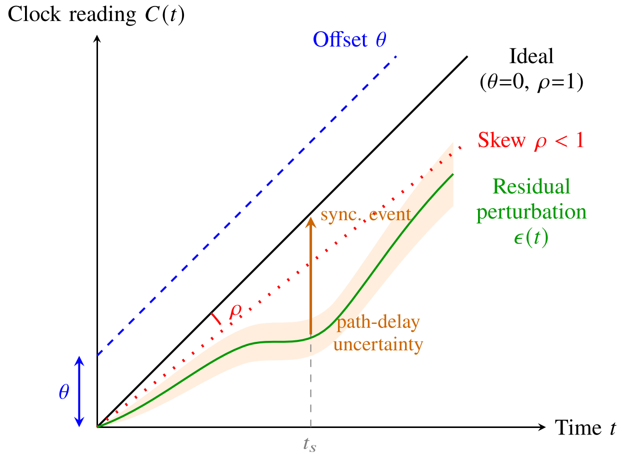

Fig. 2. ***Conceptual illustration of clock offset θ, skew ρ, and residual perturbation $\epsilon(t)$ relative to an ideal clock. A synchronization event at t_s denotes an instantaneous correction that realigns the local clock with the reference. The shaded band represents extrinsic path-delay uncertainty introduced during timestamp exchange.***

Beyond intrinsic clock imperfections, synchronization protocols must also contend with extrinsic timing uncertainty arising from the stochastic behavior of message propagation delays, whose statistical characteristics vary significantly across deployment environments and link types, as detailed in Section III [15], [17], [18], [19].

In wireless environments, the combination of intrinsic oscillator imperfections and extrinsic delay variability introduced by the network and channel makes synchronization particularly challenging. Effective protocols must therefore jointly address offset estimation, skew compensation, and robustness against stochastic timing perturbations and uncertain message delays [20], [21], [22].

B. Performance Metrics

Evaluating synchronization protocols requires a clear understanding of the performance metrics that determine their suitability for specific applications. In wireless environments, these metrics are particularly critical not only due to the inherent variability of the channel but also because diverse application domains such as industrial control, vehicular coordination, and large-scale IoT systems impose different constraints on accuracy, scalability, and energy consumption [5]. The most relevant metrics are defined as follows.

1) Accuracy: It measures how closely a local clock aligns with an ideal reference clock, such as UTC. Accuracy is often expressed as the mean or maximum offset over a given observation period, and is fundamentally measured in time units (ms, μs, ns, and ps). Financial trading systems may require millisecond-level accuracy to meet regulations, whereas industrial TSN deployments explicitly target sub-microsecond accuracy for deterministic scheduling.

2) Precision: It reflects the consistency of time agreement among a group of nodes, regardless of their absolute offset from an ideal reference clock. Precision is often characterized by the maximum pairwise clock difference or variance across nodes, and is fundamentally measured in time units (ms, μs, ns, and ps). High precision enables coordinated behaviors in cooperative robotics and distributed sensing [23].

3) Jitter: It describes short-term fluctuations in clock offset between synchronization events, and is fundamentally measured in time units (ms, μs, ns, and ps). Jitter is critical in time-slotted systems such as TSN, where even small variations may lead to missed deadlines or packet collisions [24].

4) Convergence Time: It denotes the time required for a network to reach the target synchronization accuracy following initialization or a disruption (e.g., node failure or mobility). It is fundamentally measured in time units (ms, μs, ns, and ps). Fast convergence is essential for mobile and vehicular networks [25].

5) Energy Consumption: It quantifies the amount of energy used by the protocol specifically due to synchronization-related activity, typically measured in joules or millijoules per node, per packet, per cycle, or per bit. It encompasses energy spent on synchronization transmissions as well as associated wake-ups and processing. Lower energy consumption is achieved through fewer or shorter sync. transmissions, more time in sleep states, and shifting processing to mains-powered nodes. Lower energy consumption means fewer/shorter sync. transmissions, more time in sleep, and/or shifting processing to mains-powered nodes [26].

6) Communication Overhead: It defines the incremental traffic load imposed on the network to establish and maintain a required synchronization performance, typically measured as packets or bits per synchronization event, per node or per unit time. Lower overhead means fewer or smaller additional messages for the same level of synchronization accuracy. The rate at which these messages are exchanged, the synchronization interval, is a configurable parameter whose tradeoffs are analyzed in Section III-F.

7) Scalability: It defines how well a protocol maintains its synchronization accuracy as network size increases. It is measured as the maximum number of nodes for which a target accuracy threshold (e.g., 95% of nodes within $\pm 1\,\mu s$) is maintained, or equivalently as the degradation in accuracy per order-of-magnitude increase in node count. Industrial-scale deployments may require thousands of synchronized devices with minimal performance degradation.

8) Robustness: It quantifies a protocol's ability to maintain synchronization accuracy within a specified tolerance under adverse conditions. It is measured as the maximum packet loss rate, number of faulty nodes, or magnitude of delay asymmetry under which the protocol maintains its target accuracy bound. For example, a protocol may

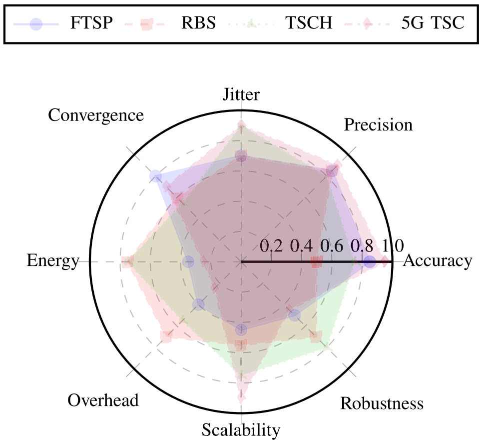

Fig. 3. ***Illustrative metric profiles for four protocols spanning distinct design paradigms across the eight performance dimensions defined in Section II-B. Axes are equally scaled for visual clarity; relative metric importance is application-dependent. Tradeoffs governing these profiles are analyzed in Section III.***

be characterized as robust if it maintains ±1-μs accuracy under 20% packet loss or up to three simultaneous faulty nodes.

To illustrate how these metrics combine into a multidimensional evaluation profile, Fig. 3 presents representative metric profiles for four protocols spanning distinct design paradigms: FTSP (flooding-based), RBS (receiver–receiver), TSCH (scheduled TDMA), and 5G TSC (infrastructure cellular). The profiles are drawn from reported performance characteristics in the literature and are intended to demonstrate that each protocol occupies a different region of the metric space, not to rank protocols against one another. Metric importance is application-dependent: an industrial TSN deployment weights accuracy and jitter most heavily, whereas a large-scale IoT deployment may prioritize energy efficiency and scalability above all others.

C. Generic Synchronization Systems

Classical synchronization protocols all share a set of recurring building blocks, even though their target environments and precision levels differ. At the core, every synchronization system can be abstracted into three main elements:

1) *Reference Source:* A time reference may be provided by a stratum 0 clock [e.g., atomic clocks or GPS], a GM node elected via clock quality metrics, or an external GNSS feed disciplining a local oscillator.
2) *Distribution Hierarchy:* Intermediate nodes propagate timing information through the network, compensating for accumulated delays and jitter along the way.
3) *End Devices:* Client or slave nodes apply offset and skew corrections to align their local oscillators with the distributed reference.

How these building blocks are instantiated across different structural configurations, and the tradeoffs each choice entails, is analyzed as part of the taxonomy in Section III.

III. TAXONOMY OF WIRELESS CLOCK SYNCHRONIZATION PROTOCOLS

A wide variety of wireless clock synchronization protocols have been proposed over the past two decades, each designed under different assumptions, performance goals, and deployment constraints. To avoid presenting these in an ad hoc manner, this section organizes them within a consistent framework. The proposed taxonomy (see Fig. 4) classifies wireless synchronization protocols along five complementary dimensions: *system architecture*, *synchronization mechanism*, *correction strategy*, *delay and uncertainty modeling*, and *resources and deployment class*. Each dimension highlights a distinct design choice: who synchronizes with whom, how timing information is obtained, what aspects of the clock are corrected, how uncertainty is handled, and how protocols scale or conserve resources. By organizing the landscape in this manner, the taxonomy offers a unified and structured lens for comparing protocols that differ fundamentally in assumptions, synchronization models, and performance objectives, extending rather than replacing prior classifications, which have typically focused on narrow subsets such as IEEE 1588-based systems or WSNs or emphasized only a single design axis such as hierarchy or message exchange.

A. System Architecture

A primary dimension for classifying synchronization protocols is their underlying architecture, i.e., how nodes are organized to establish and maintain temporal alignment. Three broad categories can be identified.

1) Master–Slave (Hierarchical): A single node, designated either statically or dynamically, acts as the master and disseminates timing information to all other nodes (slaves). Intermediate nodes limit error propagation by retiming packets at each hop. The authority structure is fixed and unambiguous, making this the simplest architecture to implement and reason about.

2) Peer-to-Peer: Nodes synchronize directly with one another without relying on a designated master. Timing information is exchanged between pairs or groups of nodes, often using relative offsets or reference broadcasts; no node holds singular time authority.

3) Fully Distributed (Consensus-Based): No node serves as a reference. Instead, all nodes iteratively converge to a common time through averaging or consensus algorithms, exchanging local estimates with neighbors until network-wide agreement is reached.

4) Hybrid Approaches: Centralized and distributed elements are combined within the same protocol. A common pattern uses centralized coordination for initial coarse alignment, followed by distributed peer exchanges for ongoing maintenance. Hybrid architectures do not fit cleanly into any single category above and are classified by their dominant coordination mode.

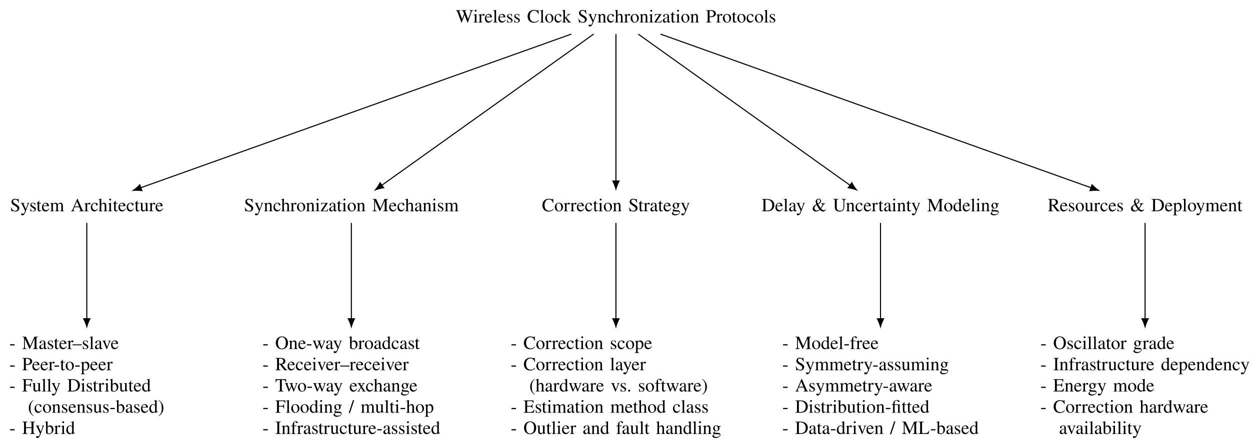

Fig. 4. ***Taxonomy of wireless clock synchronization protocols along five dimensions: 1) system architecture, who synchronizes with whom; 2) synchronization mechanism, how timing information is exchanged; 3) correction strategy, what clock parameters are estimated and how corrections are applied; 4) delay and uncertainty modeling, how propagation delay and channel uncertainty are handled; and 5) resource and deployment class, oscillator grade, infrastructure dependency, energy mode, and hardware constraints.***

B. Synchronization Mechanism

Another critical dimension of classification concerns the communication patterns used to exchange timing information. Five broad categories can be identified.

1) One-Way Broadcast: A reference node periodically transmits timing messages, which receivers timestamp upon arrival. Only the sender's timestamp and the receiver's arrival timestamp are used to estimate relative clock offset or skew. No return message is required, making this the lightest mechanism in terms of communication cost.

2) Receiver–Receiver: Synchronization is achieved using only receiver-side timestamps. A reference node transmits a broadcast packet, and two or more receivers exchange their recorded arrival times to estimate their relative clock offset. By eliminating sender-side timestamps entirely, this mechanism removes nondeterministic transmission delay as a source of error. Relative precision among receivers can be high even when absolute accuracy with respect to the reference is limited.

3) Two-Way Message Exchange: Timing messages flow in both directions between a master and a slave node, enabling explicit measurement of round-trip propagation delay. Both sender and receiver timestamps from each direction are used to jointly estimate clock offset and one-way delay. This mechanism not only places higher demands on communication resources but also provides the information needed to correct for both offset and propagation asymmetry.

4) Flooding and Multihop Propagation: A time reference periodically floods timestamped packets, which are rebroadcast by intermediate nodes to propagate timing information across the network. MAC-layer timestamping is typically used to capture transmission times as close to the physical layer as possible, reducing queuing-induced uncertainty at each hop.

5) Infrastructure-Assisted: Timing information is derived from signals already present in the wireless environment, periodic beacons, ranging frames, or broadcast control messages, rather than from dedicated synchronization exchanges. This mechanism leverages existing infrastructure without additional message overhead, with achievable precision bounded by the timing granularity of the underlying technology.

6) Hybrid and Adaptive: Some protocols combine two or more of the above mechanisms within a single design, selecting among them based on network conditions, node role, or measured synchronization error. The dominant mechanism determines the protocol's taxonomy classification; the secondary mechanism is noted as a modifier.

C. Correction Strategy

A third dimension concerns what clock parameters a protocol estimates and corrects, and how corrections are applied. Protocols can be classified along three axes.

1) Correction Scope: Protocols differ in which clock error terms they address. Offset-only correction adjusts the instantaneous time difference between clocks but leaves the underlying frequency difference unaddressed, causing renewed divergence between synchronization events. Offset-and-skew correction additionally compensates for the relative frequency difference, reducing the rate of post-correction divergence. Full correction further addresses stochastic drift through adaptive or learning-based mechanisms.

2) Correction Layer (Hardware Versus Software): Protocols may apply corrections directly to the hardware clock, tuning the oscillator frequency via phase-locked loops or fractional-delay filters. This enables sub-microsecond precision but requires hardware access and incurs implementation cost. Alternatively, protocols maintain a logical software clock overlay [27], applying corrections to a virtual time representation without modifying the underlying oscillator. Hybrid approaches combine both: software corrections for routine adjustments and hardware corrections when accumulated error exceeds a threshold.

3) Estimation Method Class: Protocols differ in how they derive correction values from observed timestamps. Model-based methods assume a parametric clock model and derive corrections analytically; examples include least-squares regression for skew estimation and maximum-likelihood estimation for joint offset and delay. State-tracking methods, such as Kalman filtering, model clock parameters as evolving state variables and update estimates recursively as new timestamps arrive. Data-driven methods make no explicit parametric assumption, instead learning correction mappings from historical observations. The estimation method class directly determines a protocol's behavior under nonstationary channel conditions and its sensitivity to outliers and missing timestamps.

4) Outlier and Fault Handling: Protocols may be outlier-agnostic, applying estimators that assume all measurements are valid, or outlier-aware, incorporating mechanisms to detect and down-weight corrupted timestamps, missing packets, or faulty nodes. Outlier-aware protocols include those using bounded-influence estimators, robust statistical filters, and low-rank matrix completion methods for recovering missing timestamp entries. This classification axis is particularly relevant in wireless environments where packet loss and interference are endemic.

D. Delay and Uncertainty Modeling

A fourth dimension concerns the assumptions a protocol makes about the statistical nature of message propagation delays and the uncertainty they introduce into clock parameter estimates. This classification axis is distinct from the correction strategy: two protocols may use the same estimation method yet differ fundamentally in their delay model assumptions, leading to different residual error floors under the same channel conditions. Five classes can be identified, in order of increasing sophistication.

1) Model-Free: The protocol makes no explicit assumption about delay distribution. Offset is estimated directly from raw timestamp differences without any statistical model. These protocols are simple to implement but provide no mechanism to distinguish measurement noise from true clock deviation.

2) Symmetry-Assuming: The protocol assumes that one-way propagation delay is equal in both directions, estimating it as half the measured round-trip time (RTT). This is the most common assumption in classical protocols and is valid under low-load, wired-equivalent conditions. It breaks down under queuing asymmetry, half-duplex wireless channels, or directional interference.

3) Asymmetry-Aware: The protocol explicitly models or estimates directional delay differences. Asymmetry may be treated as an additional deterministic parameter to be jointly estimated alongside offset and skew, or compensated through receiver–receiver designs that eliminate sender-side delay entirely. This class is appropriate for multihop wireless channels where uplink and downlink delays differ systematically.

4) Distribution-Fitted: The protocol assumes a specific parametric distribution for one-way delays, typically Gaussian for symmetric short-range links, exponential for queuing-dominated or infrastructure-based links, or Gamma for hardware-timestamped systems with bounded skew, and derives estimators optimized for that distribution. As established in Section II-A, the choice of delay model determines which estimation methods are applicable and what residual error floor is achievable.

5) Data-Driven and ML-Based: The protocol learns delay characteristics or drift patterns from observed data rather than assuming a fixed parametric model. This class is appropriate for nonstationary environments where delay distributions shift over time due to mobility, temperature, or changing network load. It requires sufficient training data and incurs higher computational overhead compared with parametric approaches.

E. Resource and Deployment Class

A fifth dimension classifies protocols by the hardware capabilities, infrastructure dependencies, and operational resource constraints they assume. This dimension determines where a protocol can be deployed and what it costs to operate, independently of its architectural or mechanism choices. Four classification axes are relevant.

1) Oscillator Grade: Protocols differ in the oscillator quality they require or assume. Consumer-grade oscillators, typical of low-cost IoT nodes, exhibit skew in the range of 10–100 PPM and require frequent resynchronization to maintain target accuracy. Industrial-grade oscillators (1–10 PPM) reduce drift between synchronization events. Temperature-compensated crystal oscillators (TCXO, <1 PPM) and oven-controlled oscillators (<0.1 PPM) enable longer holdover intervals but at higher cost and energy consumption. A protocol designed for consumer-grade oscillators will require a higher synchronization rate than one designed for TCXO-grade hardware to achieve the same accuracy target.

2) Infrastructure Dependency: Protocols may be infrastructure-free, relying solely on peer exchanges among network nodes with no external reference. Infrastructure-assisted protocols leverage existing wireless signals, AP beacons, cellular timing broadcasts, or GNSS feeds, to reduce the synchronization message overhead or improve absolute accuracy. Infrastructure-dependent protocols require a specific external reference and cannot operate without it. This axis directly determines deployment flexibility and vulnerability to infrastructure failure.

3) Energy Mode: Protocols can be classified as continuous, event-driven, or adaptive. Continuous protocols maintain a fixed synchronization rate regardless of observed drift, providing predictable accuracy at the cost of constant energy expenditure. Event-driven protocols trigger synchronization only when accumulated drift exceeds a threshold or a node wakes from sleep, conserving energy at the cost of potentially larger transient errors. Adaptive protocols adjust their synchronization rate dynamically based on measured drift rate, channel conditions, or application demand.

4) Correction Hardware Availability: Protocols may require access to hardware timestamping at the MAC or physical layer, phase-locked loop tuning interfaces, or dedicated timing hardware such as a PPS input. Protocols that rely on software-only timestamps are deployable on any general-purpose platform but face a precision ceiling imposed by operating system scheduling jitter. This axis intersects with the correction strategy dimension in Section III-C but is classified here as a deployment constraint rather than a design choice.

F. Design Tradeoffs and Limitations

The five taxonomy dimensions introduced above are not independent: choices along one dimension constrain or interact with choices along others. This section identifies the principal tradeoff axes that shape wireless synchronization protocol design, incorporating robustness and synchronization interval as cross-cutting concerns.

1) Accuracy Versus Complexity: Achieving submicrosecond accuracy requires frequent message exchanges, hardware timestamping, and precise delay modeling, all of which increase computational and communication overhead. Simpler protocols with fewer messages and model-free delay assumptions offer lower overhead but face a precision ceiling imposed by unmodeled delay variability and quantization error. This tradeoff is fundamental: accuracy improvements beyond a certain threshold require disproportionate increases in protocol complexity.

2) Centralized Versus Distributed: Master–slave architectures are straightforward to deploy and can achieve high precision through well-understood hierarchical coordination, but they introduce single points of failure and scalability limits as network size grows. Distributed consensus approaches remove these bottlenecks and improve resilience to node failures, yet suffer from slower convergence and higher message complexity per synchronization cycle. Hybrid architectures attempt to balance these extremes but inherit complexity from both paradigms.

3) Precision Versus Scalability: Protocols that achieve sub-microsecond accuracy typically rely on frequent message exchanges, hardware timestamping, and centralized coordination, all of which degrade as network size increases. Flooding-based protocols propagate timing information efficiently across multihop networks but accumulate error at each hop. Consensus-based protocols scale to large networks but converge slowly and require high message complexity. The precision–scalability boundary is fundamental: no known wireless protocol achieves both simultaneously without hierarchical decomposition or infrastructure assistance.

4) Robustness Versus Overhead: Robust protocols, those that remain accurate under packet loss, interference, mobility, or node failure, achieve this through redundancy: more messages, more diverse paths, or more conservative estimators. Each of these increases communication overhead and energy consumption. Outlier-aware estimators add computational cost. Consensus-based resilience requires iterative exchanges. The robustness–overhead tradeoff is particularly acute in energy-constrained IoT deployments where both dimensions are simultaneously critical.

5) Energy Tradeoffs: Frequent resynchronization through flooding or iterative averaging reduces accumulated drift but drains energy in battery-powered networks. Beacon-based and event-driven approaches minimize overhead by exploiting existing transmissions or triggering updates only when drift thresholds are exceeded. Adaptive schemes adjust synchronization frequency to observed drift rate or application demand, but introduce parameter tuning complexity. The energy–accuracy tradeoff is modulated by oscillator grade: a higher grade oscillator drifts more slowly, allowing longer synchronization intervals for the same accuracy target.

6) Synchronization Interval as a Design Lever: The rate at which a protocol exchanges synchronization messages is the primary lever governing the accuracy–energy–overhead tradeoff. A shorter synchronization interval reduces accumulated drift between corrections but increases message frequency, energy consumption, and channel load. A longer interval conserves resources but allows larger transient errors, particularly on nodes with high-skew consumer-grade oscillators. The optimal synchronization interval is therefore a function of oscillator grade, target accuracy, channel reliability, and energy budget simultaneously, not a fixed protocol parameter.

7) Delay Model Mismatch: Protocols optimized for one delay distribution class perform poorly when the actual channel exhibits a different distribution. A symmetry-assuming protocol deployed on an asymmetric link will exhibit a systematic offset bias that no amount of increased synchronization frequency can correct. A distribution-fit protocol trained on indoor Gaussian delays may degrade significantly in a mobile outdoor environment. Model mismatch is a structural limitation that cannot be resolved by parameter tuning alone.

8) Failure Modes and Resilience: Master–slave architectures are vulnerable to master node failure and to injection attacks that compromise the reference clock. Peer-to-peer and consensus architectures are more resilient to individual node failures but can experience slow convergence or oscillation under network partition. Infrastructure-dependent protocols fail completely when the external reference is unavailable, a critical limitation for GNSS-dependent systems in indoor or jamming-prone environments. Outlier susceptibility is a failure mode common to all protocol classes: a small number of corrupted timestamps can significantly bias offset and skew estimates in protocols that do not incorporate robust estimation.

9) Hardware Versus Software Correction Limits: Hardware correction achieves tighter precision but requires platform support and introduces implementation complexity. Software correction is universally deployable but faces a precision floor set by operating system scheduling jitter, typically in the range of tens to hundreds of microseconds on general-purpose platforms. This limit is independent of estimation method quality and represents a fundamental deployment constraint.

Together, these tradeoffs illustrate that protocol selection is inherently application-driven. The taxonomy dimensions introduced in Sections III-A–III-E provide the analytical vocabulary for making these selection decisions explicit and systematic. Their instantiation in specific protocols is examined in Section IV.

IV. CANONICAL AND REPRESENTATIVE SYNCHRONIZATION PROTOCOLS

This section instantiates the taxonomy introduced in Section III by describing the canonical and representative synchronization protocols organized along its dimensions. Section IV-A covers the foundational wired and infrastructure protocols that provide the reference framework against which wireless protocols are compared. Section IV-B groups wireless protocols by their taxonomy position rather than by the wireless technology they operate over; technology-specific instantiations are examined in Section V. Finally, Section IV-C draws comparative insights across protocol classes through the analytical lens of the taxonomy.

A. Canonical Synchronization Protocols

The following protocols constitute the canonical reference tier of synchronization systems. They are not wireless protocols per se, but they define the accuracy benchmarks, architectural patterns, and message exchange mechanisms

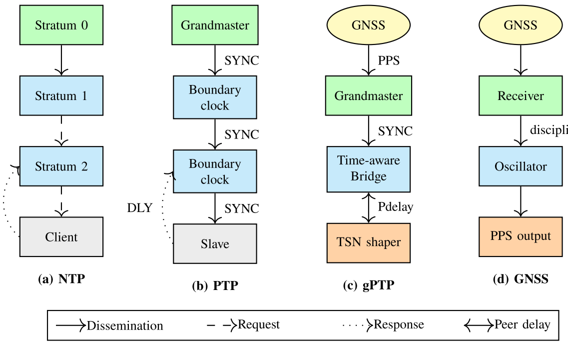

Fig. 5. ***Canonical synchronization architectures. (a) NTP strata. (b) PTP with boundary clocks. (c) gPTP with GNSS-disciplined GM and peer delay measurement. (d) GNSS disciplining a local oscillator.***

that wireless protocols extend, adapt, or interface with. Fig. 5 illustrates their contrasting architectures.

NTP [9] organizes time sources into a hierarchy of strata. Stratum 0 devices, atomic clocks, GPS receivers, or other primary references feed stratum 1 servers, which in turn serve stratum 2 clients, and so on. Each server estimates its offset and round-trip delay to its upstream peer using a two-way message exchange, applying filtering and selection algorithms to reject outliers and select the best available source. NTP achieves millisecond-level accuracy over the public internet and is the universal baseline for networked time distribution. In the proposed taxonomy, NTP is classified as a master–slave synchronization protocol that uses a two-way exchange mechanism, performs both offset and skew correction, assumes a symmetric delay model, and typically relies on existing network infrastructure for deployment.

PTP [28] was designed for local area networks requiring sub-microsecond accuracy. It retains the master–slave architecture of NTP but adds hardware timestamping at the MAC layer to eliminate software jitter and introduces boundary clocks that retime synchronization packets at each network hop to prevent queuing delay accumulation. The best master clock algorithm dynamically elects the GM based on clock quality attributes. PTP achieves sub-microsecond accuracy in switched Ethernet networks and is the foundation of different profiles used today in TSN and 5G systems. In the taxonomy, PTP occupies the master–slave architecture, two-way exchange mechanism, hardware correction layer, and symmetry-assuming delay model in its basic form, with asymmetry-aware extensions available through optional calibration.

gPTP [29] is a profile of PTP tailored for TSN. It restricts the protocol to a single clock domain, mandates hardware timestamping, and integrates directly with TSN scheduling mechanisms such as IEEE 802.1Qbv time-aware shapers. GNSS-disciplined GMs are commonly used to inject absolute UTC traceability into TSN domains. gPTP is the synchronization backbone of industrial wireless TSN deployments and the reference accuracy benchmark for 5G TSC integration. In the taxonomy, gPTP occupies the master–slave architecture, two-way exchange mechanism, hardware correction layer, and symmetry-assuming delay model with mandatory peer-delay measurement.

External Time Sources (GNSS and Atomic References): GNSS systems like GPS, Galileo, GLONASS, and BeiDou provide nanosecond-level absolute time traceability by deriving timing from atomic clocks aboard satellites [30]. A GNSS receiver disciplines a local oscillator to UTC, producing PPS output that feeds GM clocks in PTP and gPTP hierarchies. Atomic clocks (cesium, rubidium, and hydrogen maser) provide holdover capability when GNSS signals are unavailable. GNSS is the dominant external time source for infrastructure deployments but requires specialized hardware, consumes significant power, and is unavailable indoors or under jamming. In the taxonomy, GNSS occupies the infrastructure-dependent deployment class, external time source category, and provides the stratum 0 reference for all hierarchical synchronization systems.

B. Wireless Synchronization Protocol Families

Wireless synchronization protocols are grouped below by their position in the taxonomy rather than by the wireless technology they operate over. This grouping reveals structural similarities and shared limitations across protocols that would otherwise appear unrelated. Each family is described once here; their behavior across specific wireless technologies is examined in Section V.

1) Flooding-Based and Master-Driven Protocols: Flooding-based protocols elect or designate a master node that periodically disseminates timestamped synchronization messages. Intermediate nodes rebroadcast these messages, propagating timing information across multihop networks without requiring a spanning tree or pairwise exchanges.

FTSP [31] elects a root, acting as a master, dynamically and floods timestamped messages from that root. Each node applies linear regression over a sliding window of received timestamps to jointly estimate offset and skew, compensating for drift between synchronization rounds. MAC-layer timestamping reduces nondeterministic transmission delay. FTSP achieves 1–10-μs accuracy in multihop WSNs. In the proposed taxonomy, FTSP is classified as a master–slave synchronization protocol with a flooding-based synchronization mechanism, offset-and-skew correction, MAC-layer hardware timestamping, a symmetry-assuming delay model, and support for consumer-grade oscillator classes.

Glossy [32] exploits constructive radio interference to flood packets simultaneously across all nodes with near-zero timing spread. By carefully controlling retransmission timing, Glossy achieves network-wide synchronization as a side effect of data dissemination, with sub-microsecond precision and high energy efficiency. In the proposed taxonomy, Glossy is classified as a master–slave synchronization protocol with a flooding-based synchronization mechanism, hardware-layer correction, and a model-free delay class.

TPSN [33] builds a spanning tree rooted at a designated master node and then performs pairwise two-way exchanges along tree edges to estimate the offset at each hop. TPSN achieves higher accuracy than NTP-style averaging but requires tree construction overhead. In the proposed taxonomy, TPSN is classified as a master–slave synchronization protocol with a two-way exchange mechanism, offset-only correction, and a symmetry-assuming delay model.

2) Reference Broadcast-Based Protocols: Reference broadcast protocols eliminate sender-side timestamp uncertainty by using only receiver-side timestamps. A reference node transmits a broadcast packet; two or more receivers record their arrival times and exchange them to estimate relative offset.

RBS [34] is the canonical receiver–receiver protocol. Because the sender's transmission time is never used, RBS is immune to nondeterministic transmission delay. Therefore, the non-deterministic transmission delay, which is often a dominant error source in software-timestamped systems, is largely eliminated. RBS achieves high relative precision among receivers but provides no absolute UTC reference. In the proposed taxonomy, RBS is classified as a peer-to-peer synchronization protocol with a receiver–receiver synchronization mechanism, offset-only correction, a model-free delay class, and a software-layer correction approach

3) Consensus-Based and Distributed Protocols: Consensus protocols require no designated master. All nodes exchange local time estimates with neighbors and iteratively update toward a network-wide agreement through averaging or optimization. Several architectural variants have been developed.

The **Average TimeSync.** protocol iteratively averages local clock parameters. It is a fully distributed clock synchronization protocol for WSNs based on a cascade of two consensus algorithms, where the first synchronizes clock drift and the second synchronizes clock offsets, resulting in a typical precision of 100 sμs [35]. In the proposed taxonomy, Average TimeSync. is classified as a fully distributed synchronization protocol with a consensus-based synchronization mechanism, a cooperative/hybrid correction strategy, offset-and skew correction, and a low-power deployment profile.

GTSP [36] propagates a global time estimate through local gradient-following adjustments. Each node maintains a logical clock and updates it based on the maximum time seen among its neighbors, converging to a common time without a fixed root. GTSP is robust to node failures and topology changes. In the proposed taxonomy, GTSP is classified as a fully distributed synchronization protocol with a peer-to-peer synchronization mechanism, offset-and-skew correction, and a symmetry-assuming delay model.

Average consensus protocols generalize this approach by having nodes iteratively compute weighted averages of neighbor timestamps [35]. Convergence is guaranteed under mild connectivity conditions, but message complexity grows with network size and convergence time increases with diameter.

Early consensus protocols often neglected delay or treated it as bounded but unspecified, whereas new variants like the maximum time synchronization protocol and the weighted-maximum time synchronization protocol explicitly handle random communication delays [37]. Other variations introduce multihop virtual links to increase graph connectivity [38], [39] and perform intracluster and intercluster consensus [40] to accelerate convergence in large network.

4) TSN-Integrated Wireless Protocols: This family bridges IEEE 802.1AS-based wired TSN with wireless segments, treating the wireless link as a transparent TSN bridge or as a time-aware system within the TSN domain.

3GPP Release 16 defines the architecture for integrating 5G as a logical TSN bridge [41], with DS-TT and NW-TT compensating for 5G residence time and asymmetric uplink/downlink delays. Val et al. [42] demonstrated IEEE 802.1AS clock synchronization in an integrated wired-wireless TSN architecture, achieving sub-microsecond accuracy at the TSN domain boundary. Satka et al. [43] confirm that 73% of TSN-5G integration studies address time synchronization, with reported accuracies from 100 ns to 1 μs. In the taxonomy: master–slave architecture, two-way exchange mechanism, hardware correction layer, symmetry-assuming delay model with residence-time compensation, and infrastructure-dependent deployment class.

5) UWB-Based Ranging Synchronization: UWB protocols exploit sub-nanosecond pulse timing to achieve ranging accuracy below 10 cm, which translates directly to sub-nanosecond clock offset estimation between ranging partners.

IEEE 802.15.4z defines the physical layer for UWB ranging, supporting two operating modes [44]. In TWR and double-sided TWR, an initiator and responder exchange packets with hardware timestamps, simultaneously estimating distance and clock offset. In TDoA deployments, multiple anchors record arrival times of a tag transmission using their individual local clocks; the tag position is derived from arrival time differences, with relative anchor offsets compensated during position computation rather than through persistent global clock alignment. Hardware timestamping at the physical layer is mandatory in both modes, enabling consistent sub-microsecond synchronization accuracy. UWB-based synchronization is particularly relevant for indoor industrial deployments where GNSS is unavailable and sub-microsecond accuracy is required. In the proposed taxonomy, IEEE 802.15.4z/UWB-based synchronization can be classified as either peer-to-peer or master–slave, depending on the deployment mode; it uses either two-way exchange or receiver receiver timing mechanisms, relies on hardware-layer timestamping, supports asymmetry-aware delay handling, and is suitable for infrastructure-free indoor deployment.

6) Cellular Infrastructure-Based Protocols: Cellular networks provide timing signals as an inherent part of their air interface, which can be leveraged for synchronization of connected devices. In 5G, base stations (gNBs) are disciplined to GNSS or PTP GMs, and the air interface carries synchronization reference signals that enable highly accurate timing distribution to connected devices. This native timing infrastructure forms the basis upon which higher level synchronization mechanisms, such as 5G TSC for external network interoperability and LTE timing advance for uplink alignment, are built.

5G TSC extends 5G to support TSN interoperability, distributing IEEE 802.1AS timing over the 5G air interface [41]. The 5G system acts as a time-aware GM or boundary clock, compensating for air interface delay through RTT-based measurements. 3GPP TS 22.104 specifies a maximum synchronization error of 900 ns for factory automation use cases. In the proposed taxonomy, 5G TSC is classified as a master–slave synchronization approach with an infrastructure-assisted synchronization mechanism, hardware-layer correction, an asymmetry-aware delay model, and an infrastructure-dependent deployment class.

LTE timing advance provides coarser synchronization through uplink timing alignment, achieving microsecond-level accuracy suitable for coordinated multipoint (CoMP) transmission but insufficient for TSN applications without augmentation.

7) Hybrid and Cross-Layer Protocols: Hybrid protocols combine elements from multiple taxonomy positions, typically pairing a fast coarse-alignment mechanism with a slower fine-correction mechanism.

TSCH/6TiSCH [45] combines TSCH with IEEE 802.15.4e MAC-layer synchronization. Nodes maintain slot-level synchronization through periodic enhanced beacons and two-way keep-alive exchanges, achieving deterministic channel access with sub-millisecond slot alignment. 6TiSCH extends this to IPv6-based IIoT networks. In the proposed taxonomy, TSCH/6TiSCH is classified as a master–slave synchronization approach with distributed maintenance, a hybrid synchronization mechanism, hardware-layer correction, a symmetry assuming delay model, and an event-driven energy mode.

8) LPWAN and Opportunistic Protocols: LPWAN protocols prioritize energy efficiency and range over synchronization precision, making different tradeoffs than all other families.

LoRaWAN achieves coarse synchronization through downlink beacon frames that define Class B receive windows, providing gateway-to-device timing alignment at millisecond granularity [46]. Synchronization is a by-product of network access rather than a primary protocol objective. **NB-IoT** relies on LTE frame timing for slot alignment, achieving similar millisecond-level accuracy with lower energy than full LTE. In the proposed taxonomy, LoRaWAN and NB-IoT are classified as master–slave synchronization approaches with infrastructure-assisted synchronization mechanisms, software-layer correction, a

Table 2 Comparative Overview of Wireless Synchronization Protocol Families Across the Five Taxonomy Dimensions

Family	D1: Architecture	D2: Mechanism	D3: Correction strategy	D4: Delay model	D5: Resource & deployment
Flooding / master-driven (FTSP, Glossy, TPSN)	Master–slave	Flooding / multi-hop	Offset + skew; HW timestamp	Symmetry-assuming	Consumer oscillator; infra-free; moderate energy
Reference broadcast (RBS)	Peer-to-peer	Receiver–receiver	Offset only; SW correction	Model-free	Consumer oscillator; infra-free; low energy
Consensus / distributed (GTSP, averaging)	Fully distributed	Peer exchange	Offset + skew; SW correction	Symmetry-assuming	Consumer oscillator; infra-free; high energy
TSN-integrated wireless (802.1AS over 5G)	Master–slave	Two-way exchange + residence time	Offset + skew; HW timestamp	Asymmetry-aware	Industrial oscillator; infra-dependent; moderate energy
UWB ranging (IEEE 802.15.4z)	Peer-to-peer / master–slave	Two-way ranging	Offset + skew; HW timestamp	Asymmetry-aware	Specialized HW; infra-free; moderate energy
Cellular (5G TSC, LTE)	Master–slave	Infrastructure-assisted	Offset + skew; HW timestamp	Asymmetry-aware	Industrial oscillator; infra-dependent; high energy
Hybrid / cross-layer (TSCH / 6TiSCH)	Master–slave + distributed	Hybrid beacon + two-way	Offset only; HW timestamp	Symmetry-assuming	Consumer oscillator; infra-assisted; low energy
LPWAN (LoRaWAN, NB-IoT)	Master–slave	Infrastructure-assisted	Offset only; SW correction	Model-free	Consumer oscillator; infra-assisted; very low energy

model-free delay class, and a continuous low-rate energy mode.

C. Comparative Insights Across Protocol Classes

Table 2 maps each protocol family to its position across all five taxonomy dimensions, making explicit the design choices that differentiate families from one another. Using this mapping as an analytical lens, several cross-cutting observations emerge.

1) Architecture Predicts Scalability Ceiling (D1): Flooding-based and master-driven protocols achieve the highest accuracy among infrastructure-free wireless families but face a hard scalability ceiling: error accumulates with hop count in flooding protocols, and spanning-tree construction overhead grows with network size in master-driven schemes. Fully distributed consensus protocols scale more gracefully but trade convergence speed for scalability. Infrastructure-assisted and cellular protocols scale without fundamental degradation but inherit the coverage and availability constraints of their infrastructure.

2) Correction Layer Is the Primary Accuracy Determinant (D3): Across all protocol families, the single strongest predictor of achievable accuracy is whether hardware timestamping is available at the MAC or physical layer. The gap between software-timestamped protocols (typically 10 μs–1 ms) and hardware-timestamped protocols (typically 100 ns–1 μs) is larger than the gap introduced by any architectural choice alone. This makes correction hardware availability the most practically constraining taxonomy dimension for deployment decisions.

3) Delay Model Class Determines Residual Error Floor (D4): Protocols in the symmetry-assuming class exhibit a systematic bias floor on asymmetric links that cannot be reduced by increasing synchronization frequency or improving oscillator grade. This structural limit separates symmetry-assuming protocols from asymmetry-aware and distribution-fit protocols regardless of all other design choices. As Table 2 shows, only TSN-integrated wireless, UWB, and cellular families adopt asymmetry-aware delay models, the three families that also achieve the highest accuracy.

4) No Family Dominates all Five Dimensions: Every protocol family in Table 2 achieves a strong position on some taxonomy dimensions while accepting weak positions on others. UWB occupies the strongest accuracy and delay model position among infrastructure-free families but accepts limited scalability and specialized hardware requirements. Cellular protocols occupy strong positions on accuracy, delay model, and scalability but are fully infrastructure-dependent and energy-intensive. LPWAN protocols occupy the strongest energy and scalability positions but accept model-free delay assumptions and software-only correction. This confirms the core finding of the taxonomy: no universally optimal protocol family exists, selection must be driven by matching the application requirement profile against the taxonomy dimension positions of candidate protocols.

V. SYNCHRONIZATION ACROSS WIRELESS TECHNOLOGIES

The taxonomy developed in Section III and instantiated across protocol families in Section IV provides the analytical framework for this section. Here, we examine how the five taxonomy dimensions, system architecture, synchronization mechanism, correction strategy, delay and uncertainty modeling, and resource and deployment class, are shaped and constrained by the physical and MAC layer properties of each wireless technology.

Wireless technologies expose fundamentally different levels of timing support. Some provide precise hardware timestamps at the PHY or MAC layer with sub-nanosecond resolution (e.g., UWB); Others rely on coarser beacon frames or frame counters (e.g., BLE, ZigBee, and LPWAN). These differences in timestamp resolution, reference signals periodicity, and infrastructure assumptions directly govern which protocol families from Section IV-B can operate on each technology and what accuracy is achievable in practice.

Sections V-A–V-E examine each technology family in turn, following a progression from foundational low-power short-range technologies through wide-area and infrastructure-based systems. Each subsection identifies the synchronization mechanisms the technology enables, maps its constraints to the taxonomy, and states its fundamental limit. Section V-F then examines cross-technology synchronization, and Section V-G provides a comparative synthesis.

A. IEEE 802.15.4, ZigBee, BLE, and Low-Power Short-Range Technologies

These foundational low-power technologies share a common design philosophy: energy efficiency and scalable mesh operation take priority over synchronization precision. Their MAC layers provide timing support through periodic beacons, connection events, or scheduled slots, mechanisms that are sufficient for duty-cycle coordination but constrain achievable clock alignment to the millisecond range under typical deployments.

BLE (Bluetooth 4.x/5.x): The BLE MAC is connection- and event-driven [47]. Devices negotiate connection intervals (7.5 ms to several seconds), wake just-in-time to exchange frames, then return to sleep. BLE Mesh extends this through managed flooding with periodic friendship polling windows [48]. There is no network-wide absolute time reference. Achievable alignment is millisecond-scale for tight connection intervals, degrading to tens of milliseconds in Mesh deployments.

ZigBee/IEEE 802.15.4: The default CSMA/CA mode provides no global time [49]. The optional beacon-enabled mode introduces a PAN coordinator that periodically transmits superframe beacons (15.36 ms to hundreds of milliseconds), to which nodes align their wake windows [50]. Alignment is coordinator-relative with no absolute time across multiple coordinators. Typical alignment is tens of milliseconds.

TSCH/6TiSCH (IEEE 802.15.4e): TSCH adds TSCH to IEEE 802.15.4e, introducing a shared global slotframe with typical slot durations of 10–15 ms [51]. Nodes join through enhanced beacons and maintain slot alignment through incremental drift corrections. The IETF 6TiSCH stack adds IPv6 and centralized or distributed scheduling over this substrate [52]. Sub-millisecond slot alignment is achievable across multihop meshes of hundreds of nodes. The flooding-based and hybrid protocol families described in Section IV-B, including FTSP and Glossy, operate predominantly over IEEE 802.15.4 physical layers and inherit these MAC timing constraints.

In the taxonomy, these technologies share a master–slave or peer-to-peer architecture, one-way beacon or scheduled beacon mechanism, offset-only or offset-and-skew correction scope, symmetry-assuming delay model, and consumer-grade oscillator with event-driven or adaptive energy mode. TSCH is the most capable member of this family, occupying the hybrid mechanism position and achieving the tightest alignment, but still constrained by its `slotframe` granularity.

The fundamental limit of this technology family is that millisecond-scale MAC timing granularity establishes a precision ceiling that no protocol running over these physical layers can overcome without hardware timestamping augmentation.

B. LPWAN: LoRa and NB-IoT

LPWAN prioritize geographic reach and multiyear device lifetime over synchronization precision. LoRaWAN devices transmit sporadically over kilometer-scale links while sleeping the majority of the time [53]. Uplinks are driven by local application schedules with no coordination among end devices; even the structured Class A timing defines only relative receive windows at fixed offsets after each uplink, not a global time reference. NB-IoT relies on LTE frame timing for slot alignment, achieving similar second-level accuracy with lower energy than full LTE [54]. In both cases, end devices operate with loose free-running clocks; precise UTC-aligned timestamping is enforced at gateways rather than at endpoints, commonly using GNSS-disciplined receivers as described in Section IV-A.

In the taxonomy, LPWAN technologies occupy the master–slave architecture, infrastructure-assisted mechanism, software-only correction layer, model-free delay class, and consumer-grade oscillator with very low continuous energy mode, the weakest position on the correction hardware availability axis of Section III-E.

The fundamental limit of LPWAN synchronization is that second-level MAC timescales and the deliberate absence of globally synchronized slotting make these technologies structurally incompatible with deterministic microsecond scheduling, regardless of the synchronization protocol applied above them.

C. Wi-Fi (IEEE 802.11)

Wi-Fi's primary MAC-layer timing mechanism is the TSF, a 64-bit microsecond counter maintained by each AP and distributed through periodic beacon frames at 100-ms intervals [55]. Stations align their local counters to received beacons, providing intra-BSS slot coordination for power saving and contention backoff. Native TSF precision is 10–100 μs, sufficient for MAC coordination but not for TSN-class scheduling. The dominant constraints are nondeterministic CSMA/CA airtime, which introduces variable beacon delay, and the absence of asymmetry compensation, which prevents TSF from serving as a global time reference across BSSs.

Wi-Fi 6 (802.11ax) and Wi-Fi 7 (802.11be) improve this substrate through OFDMA scheduling and multilink operation, reducing timing variability and creating a more stable platform for higher layer synchronization [56]. Deploying gPTP over Wi-Fi with APs acting as boundary clocks per IEEE 802.1AS, as described in Section IV-A, achieves 1–5 μs under light load, degrading to 10–50 μs under contention. Hybrid architectures that localize wireless uncertainty at the edge while a wired TSN backbone maintains sub-microsecond determinism represent the most practical current integration pattern.

In the taxonomy, Wi-Fi occupies the master–slave architecture, one-way beacon mechanism for native TSF or two-way exchange mechanism for gPTP overlays, software correction layer for TSF or hardware correction layer for gPTP, and symmetry-assuming delay model, with the CSMA/CA nondeterminism sitting entirely within the delay model mismatch tradeoff identified in Section III-F.

Wi-Fi can serve as a capable edge extension of a TSN domain but cannot function as a deterministic synchronization backbone due to the structural nondeterminism of its MAC layer.

D. Ultrawideband

UWB transmits short pulses over bandwidths of 500 MHz or more, enabling hardware timestamping at picosecond resolution at the physical layer [44]. The IEEE 802.15.4z amendment standardizes PHY modes optimized for secure high-accuracy ranging, including hardened timestamping and scrambled timestamp sequences. This physical resolution translates directly to centimeter-scale ranging accuracy and, under favorable line-of-sight conditions, sub-microsecond clock offset estimation between ranging partners. In practice, the majority of UWB deployments exploit this capability for event time-stamping and ranging rather than for persistent global clock distribution, TWR and TDoA operating modes, and their use for synchronization, are described in Section IV-B.

The primary constraints on UWB as a synchronization substrate are multipath and non-line of sight (NLoS) propagation, which introduce systematic ranging biases that cannot be eliminated by increasing exchange frequency, and the absence of a standardized synchronization protocol analogous to IEEE 1588 that would enable interoperability across vendors [57]. Hardware timestamping augmentation is available but requires specialized radio hardware, placing UWB in the infrastructure-free but specialized hardware deployment class of Section III-E.

In the taxonomy, UWB occupies the peer-to-peer or master–slave architecture, two-way exchange mechanism, hardware correction layer, and asymmetry-aware delay model, the strongest delay model position among infrastructure-free wireless technologies, constrained by the absence of a global synchronization standard.

UWB's natural role is as a provider of ultraprecise spatial and temporal event markers that are fused into a TSN-managed network rather than as a standalone global synchronization substrate.

E. Cellular (4G/5G/6G) and Time-Sensitive Communications

Cellular systems require accurate time and phase alignment as a native infrastructure function: intercell handover, CoMP transmission, carrier aggregation, and frame timing all depend on tight synchronization across base stations [58]. In 4G/5G deployments, gNBs derive a reference from GNSS or an IEEE 1588 GM via ITU-T G.8275.1 PTP telecom profile over wireline backhaul, then propagate frame timing to UEs through system information blocks (SIBs) and radio resource control (RRC) signaling [41]. This infrastructure-managed timing chain achieves phase alignment across radio sites at the submicrosecond level under controlled backhaul conditions.

3GPP Releases 16–18 extend this timing role from radio coordination to industrial determinism through TSC, which couples 5G with IEEE TSN time models so that a 5G slice acts as a TSN boundary or transparent clock rather than sitting beside the TSN domain [59]. The gNB compensates for air interface residence time through NW-TT and DS-TT translators, extending the TSN GM clock across the wireless hop to UE-attached controllers and actuators [60]. Fig. 6 illustrates this timing chain for a converged 5G-TSN network with a downlink GM. Lab trials and early industrial pilots report 1–5-μs offset and jitter relative to the TSN backbone under GNSS-disciplined gNBs and controlled radio conditions [61]. The cellular TSC protocol family described in Section IV-B instantiates these mechanisms in full detail.

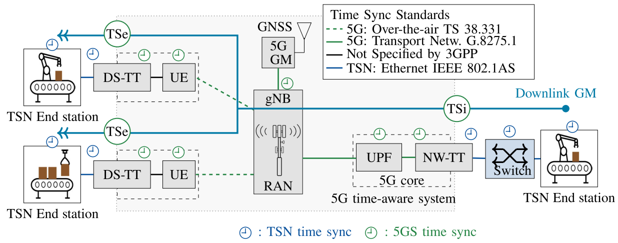

Fig. 6. ***Time synchronization for a converged 5G–TSN network with a downlink GM, illustrating the timing chain from GNSS-disciplined gNB through the air interface to UE-attached TSN devices.***

In the taxonomy, cellular technologies occupy the master–slave architecture, infrastructure-assisted mechanism, hardware correction layer, asymmetry-aware delay model with residence time compensation, and infrastructure-dependent deployment class with industrial-grade oscillator, the strongest overall taxonomy position among wireless technologies, at the cost of full infrastructure dependency.

A well-engineered 5G TSC deployment can extend wired TSN determinism to mobile endpoints at 1–5 μs, but achieving and sustaining those figures requires cross-layer timing design across GNSS discipline, scheduler configuration, and transport calibration that generic mobile broadband provisioning does not provide.

F. Cross-Technology Synchronization

Individual wireless technologies establish synchronization within their own domains, but industrial and wide-area deployments increasingly require coherent timing across heterogeneous segments. GNSS-disciplined GMs, described in Section IV-A, provide the common absolute time reference that makes cross-domain alignment possible without requiring direct coupling between wireless technologies.

The most mature realized architecture is the integration of 5G as a logical TSN bridge, standardized in 3GPP Releases 16–18 [41]. NW-TT and DS-TT map TSN streams and timing requirements onto 5G QoS flows, allowing a 5G slice to extend a wired gPTP domain across the air interface. Seijo et al. [62] demonstrated IEEE 802.15.4z synchronization in an integrated wired-wireless TSN architecture achieving sub-microsecond accuracy at the domain boundary. Satka et al. [43] confirmed that 73% of TSN-5G integration studies address time synchronization as the primary integration challenge, with reported accuracies from 100 ns to 1 μs depending on backhaul configuration and radio conditions.

Beyond 5G-TSN, heterogeneous deployments commonly adopt a layered pattern: a wired TSN core maintains sub-microsecond determinism; Wi-Fi or UWB segments extend this to wireless endpoints with gPTP boundary clocks or precise event markers; and LPWAN segments contribute asynchronous telemetry time-stamped at GNSS-disciplined gateways. Other approaches where Wi-Fi assists ZigBee devices for time synchronization [63] or bidirectional clock synchronization between BLE and IEEE 802.15.4 devices are also investigated [64], demonstrating synchronization error in microseconds. Each layer operates within its precision tier without requiring the others to change their synchronization substrate.

The current fundamental limit of cross-technology synchronization is the absence of a unified clock hierarchy standard that spans cellular, Wi-Fi, UWB, and LPWAN segments simultaneously; each technology requires its own translation mechanism at domain boundaries, and cumulative boundary errors constrain end-to-end accuracy. Open research directions addressing this limit are examined in Section VIII.

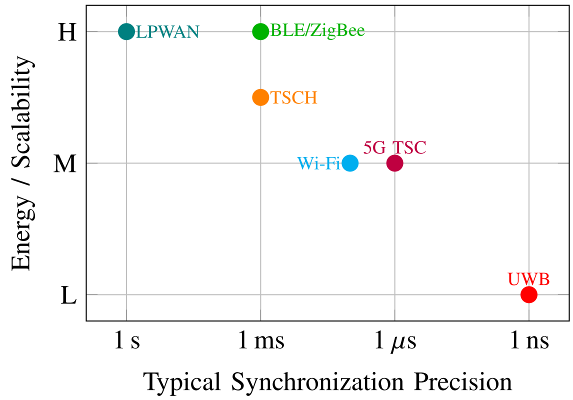

Fig. 7. ***Wireless synchronization technologies positioned by achievable precision and energy/scalability.***

G. Comparative Insights Across Technologies

Table 3 compares wireless technologies across the physical and MAC layer constraints that govern achievable synchronization performance. Using these constraints as an analytical lens, several cross-cutting observations emerge.

Timestamp Resolution Is the Primary Precision Determinant: The gap between software-timestamped technologies (IEEE 802.15.4, BLE, LPWAN, and Wi-Fi native TSF) operating at microsecond-to-millisecond granularity and hardware-timestamped technologies (TSCH, UWB, and cellular) operating at nanosecond-to-picosecond granularity is larger than any other constraint difference in the table. This mirrors the D3 correction layer finding of Section IV-C at the technology level, and hardware timestamping availability is the single strongest predictor of achievable precision tier. Fig. 7 positions all seven technologies along this precision axis against their energy and scalability characteristics, illustrating how the timestamp resolution constraint translates directly into deployment tradeoffs.

Reference Signal Periodicity Governs the Drift Correction Ceiling: Technologies with infrequent or demand-driven reference signals (LPWAN at seconds, UWB TWR on demand) cannot bound drift accumulation between corrections regardless of oscillator grade. Technologies with frequent scheduled reference signals (TSCH slotframe and cellular frame timing) maintain tighter drift bounds as a structural property of their MAC layer rather than as a protocol design choice.

Infrastructure Assumption Constrains Deployment Flexibility: Technologies requiring fixed infrastructure (cellular and Wi-Fi) achieve the highest precision tiers but cannot operate in infrastructure-free environments. Technologies capable of infrastructure-free operation (IEEE 802.15.4, BLE, and UWB) sacrifice precision or require protocol-level compensation to approach the accuracy of infrastructure-dependent alternatives.

Table 3 Wireless Synchronization Technologies Compared Across Physical and MAC Layer Constraints Governing Achievable Synchronization Performance

Technology	Timestamp resolution	Reference signal	Infrastructure assumption	Precision tier	TSN suitability
IEEE 802.15.4 ZigBee	SW; frame counter (~30 μs)	Periodic beacon (15 ms–100s ms)	PAN coordinator (infra-free)	10s ms	Not suitable
BLE	SW; connection event counter (~1 ms)	Connection interval (7.5 ms–seconds)	Master device (infra-free)	1–10s ms	Not suitable
TSCH / 6TiSCH	HW; MAC layer (~1 μs)	Enhanced beacon + keep-alive (slotframe)	Root node (infra-assisted)	Sub-ms	Partial , gateway only
LPWAN (LoRa, NB-IoT)	SW; gateway only (seconds at device)	Downlink window (seconds)	Network server + gateway (infra-assisted)	Seconds at device; μs at gateway	Not suitable
Wi-Fi (IEEE 802.11)	SW native TSF (~1 μs); HW with gPTP	Beacon (100 ms) + CSMA/CA variable	Access point (infra-dependent)	10–100 μs native; 1–5 μs gPTP	Edge only , not backbone
UWB (IEEE 802.15.4z)	HW; PHY layer (~10s ps)	TWR / TDoA exchange (on demand)	Anchors optional (infra-free capable)	Sub-μs (lab); limited by NLOS	Event markers only , not global clock
Cellular (4G/5G TSC)	HW; PHY layer (<100 ns)	Frame timing + SIB / RRC signaling	gNB + GNSS backhaul (infra-dependent)	1–5 μs (TSC slice)	Yes , via NW-TT / DS-TT

No Technology Is Universally TSN-Suitable: Only cellular via NW-TT/DS-TT achieves full TSN domain participation. TSCH and Wi-Fi gPTP overlays reach the edge of TSN suitability as gateway or boundary clock technologies. All other technologies require a TSN-capable gateway to interface with deterministic domains. Technology selection must therefore be driven by the application requirement profile examined in Section VI, matched against the precision tier and TSN suitability columns of Table 3.

VI. APPLICATIONS AND USE CASES

Wireless clock synchronization underpins coordinated sensing, actuation, and communication across a wide range of distributed real-time systems. Precision requirements vary by several orders of magnitude across domains, from sub-microsecond for industrial automation and immersive XR, through sub-millisecond for vehicular safety and grid protection, to second level for delay-tolerant wide-area sensing. Each domain imposes a distinct requirement profile across the performance metrics defined in Section II-B, which in turn selects among the protocol families of Section IV-B and the wireless technologies of Section V.

Sections VI-A–VI-E survey six representative application domains in order of decreasing synchronization precision requirement. For each domain, we identify the timing demands, the protocol families and technologies that serve them, and the open challenges addressed in Section VIII.

A. Industrial Automation and IIoT

Modern factories coordinate networks of machines, sensors, and controllers whose actions must align within tight temporal budgets. In discrete manufacturing, multi-axis computer numerical control (CNC) and robotic cells require simultaneous motion starts and phase-aligned trajectories with sub-100-μs jitter; high-speed pick-and-place lines and inline vision inspection systems impose similar constraints [65]. Process industries (chemicals, food and beverage, pulp, and paper) operate with slower control loops where sub-millisecond timing typically suffices, though packaging and labeling subsystems inherit the stricter requirements of robotics.

The synchronization substrate for most smart factories is a wired TSN backbone distributing IEEE 802.1AS time at sub-microsecond accuracy, with wireless segments layered on top where mobility or retrofit requirements outweigh cable runs [66]. Wi-Fi 6/7 gPTP overlays serve coordinated but not hard real-time functions such as mobile inspection carts and retrofit I/O, as described in Section V-C. 5G TSC links, described in Section V-E, support autonomous mobile robots and mobile arms requiring microsecond-scale coordination across zones. UWB ranging, described in Section V-D, complements these technologies by providing sub-microsecond event markers for precise indoor localization of autonomous mobile robots and collaborative robot safety zones, where centimeter-scale position accuracy translates directly to sub-nanosecond timestamp resolution at anchors. BLE and IEEE 802.15.4/TSCH segments, described in Section V-A, contribute millisecond-level alignment for environmental sensing and auxiliary inputs, with gateways timestamping data against the plant time base for unified diagnostics.

Industrial digital twins, virtual representations of physical assets updated in real time, impose continuous sub-millisecond timestamp consistency across sensor streams to maintain model fidelity; coarser synchronization introduces state divergence that degrades predictive maintenance and closed-loop simulation accuracy [67]. AI inference at the factory edge imposes stricter requirements still: vision-based quality inspection and robotic guidance systems that feed inference results directly

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

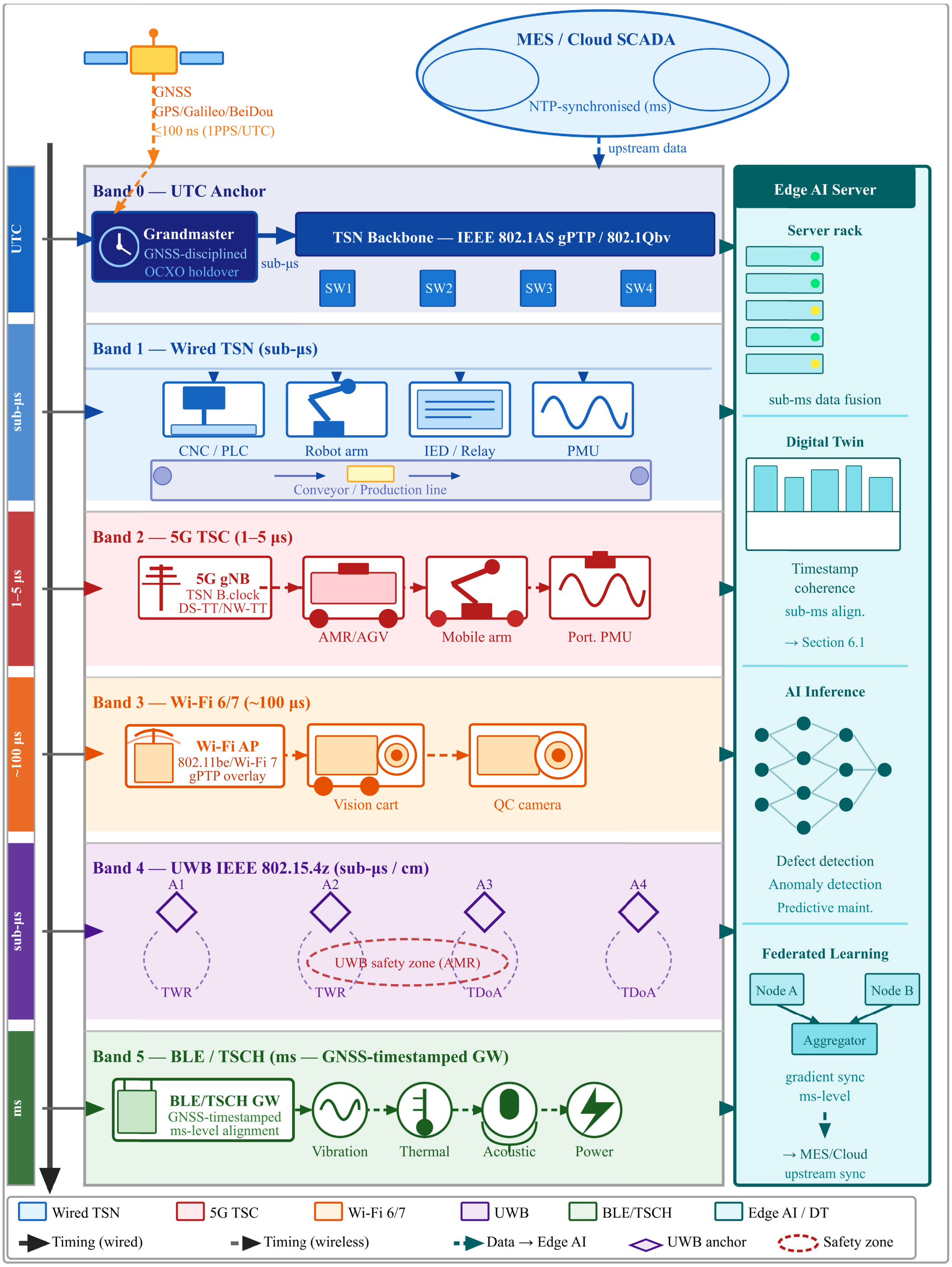

Fig. 8. *Layered synchronization architecture for industrial automation and IIoT. Six coexisting timing tiers span from sub-microsecond wired TSN (Band 1) for hard real-time motion control and PMU synchrophasor measurement, through 5G TSC (Band 2) for mobile platforms and portable PMUs, Wi-Fi 6/7 gPTP (Band 3) for vision and inspection, UWB (Band 4) for sub-centimeter indoor localization and safety zoning, to millisecond BLE/TSCH (Band 5) for low-power condition monitoring sensors. All data streams converge on an edge AI server maintaining sub-millisecond timestamp coherence for digital twin consistency, distributed inference, and federated learning. The entire hierarchy is anchored by a GNSS-disciplined GM clock (Band 0) distributing UTC via IEEE 802.1AS gPTP.*

into control loops require sub-microsecond sensor stream alignment to prevent pipeline stalls and ensure that actuator commands are issued within the control cycle budget. Distributed inference across partitioned edge models similarly requires synchronized activation passing at sub-microsecond precision across edge nodes. Federated learning across factory edge nodes operates at the relaxed end of this spectrum, requiring millisecond-level synchronization of model update rounds to prevent gradient staleness.

Advanced deployments combine a wired TSN core for hard real-time motion control, 5G TSC or Wi-Fi TSN for mobile machinery and vision, and BLE or ZigBee for low-power sensing, with GNSS-disciplined GMs anchoring a single UTC time base across all segments [62] (see Fig. 8). Protocol families serving this domain span the full range of Section IV-B, from flooding-based WSN protocols at the sensing tier to TSN-integrated wireless at the control tier. A practical design principle for factory edge AI is to apply TSCH or GNSS-disciplined 5G synchronization at sub-microsecond granularity to safety-critical and high-value sensor streams, while accepting millisecond-scale BLE or opportunistic alignment for lower priority telemetry, matching synchronization cost to inference pipeline sensitivity. Time synchronization of IIoT devices targeting energy efficiency is explored in [68]. Open challenges including multivendor timing alignment, RF resilience in harsh factory environments, and unified clock hierarchies across heterogeneous segments are examined in Section VIII.

B. Distributed XR and Metaverse

XR applications, encompassing VR, AR, and mixed reality, require both low latency and tight interdevice time alignment across devices, users, and edge compute nodes. Shared VR sessions require device clocks aligned within 1–5 ms to avoid perceptible scene inconsistencies. AR overlays tracking physical actions require similar millisecond alignment to prevent visible slip. Haptic feedback systems push requirements to sub-100-μs jitter; deviations of even a few hundred microseconds are detectable by the human nervous system [69]. Distributed edge inference engines performing real-time rendering across partitioned compute nodes inherit these sub-100-μs alignment requirements at the activation passing boundary, making AI-assisted XR one of the most demanding synchronization use cases in the confirmed precision ordering of this section.

Multimodal XR systems integrate visual scenes rendered at 60–120 Hz, spatialized audio, and haptic actuation. For convincing immersion, these streams must remain temporally aligned: visual and audio updates should differ by no more than a few milliseconds, while haptic feedback must track visual cues within $\sim$100 μs. Fig. 9 illustrates a collaborative XR scenario where visual, audio, and haptic events are synchronized via a shared global clock. Wi-Fi 6/7 gPTP overlays, described in Section V-C, can limit jitter to $\sim$100 μs under controlled conditions, sufficient for audio-visual coherence in multiuser VR but marginal for haptic-grade performance. 5G TSC links, described in Section V-E, align rendering, sensing, and haptic events to within microseconds across rooms or campuses and enable edge-assisted multiview rendering with time-consistent sensor fusion [70]. UWB, described in Section V-D, provides sub-microsecond event markers for haptic and proximity-triggered interactions where Wi-Fi jitter is insufficient.

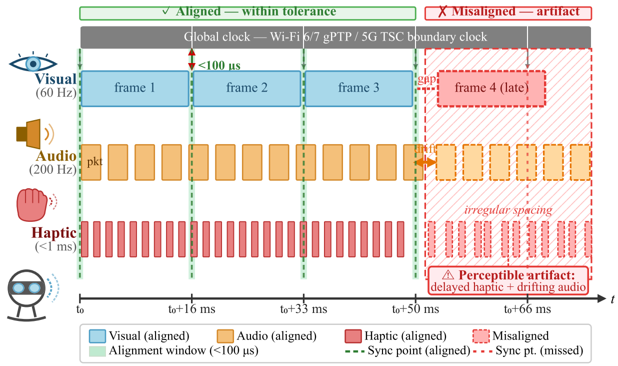

Fig. 9. ***Multimodal XR synchronization: visual, audio, and haptic streams are aligned to a shared global clock. Even sub-millisecond misalignments can cause perceptible artifacts.***

Haptic use cases in current deployments commonly fall back to UWB or wired TSN for guaranteed sub-100-μs jitter; fully wireless microsecond-level synchronization for distributed XR and Metaverse environments remains an open challenge as Wi-Fi 7 and 5G TSC mature.

C. Vehicular Networks and V2X

V2X coordination spanning V2V, vehicle-to-infrastructure, and vehicle-to-pedestrian interactions requires a shared time base under high mobility, rapidly changing topologies, and variable radio conditions. Truck platooning with meter-scale gaps requires end-to-end control loops within 10 ms and sensor timestamp alignment within sub-millisecond bounds. Intersection negotiation and cooperative collision avoidance operate within similar 10-ms round-trip budgets. Cooperative perception, fusing LiDAR, radar, and camera detections across vehicles for beyond-line-of-sight awareness, requires sub-millisecond clock agreement to prevent misalignment of fast-moving object detections [71]. AI-based inference engines performing real-time sensor fusion across vehicle fleets inherit these sub-millisecond timestamp alignment requirements at the fusion input.

V2X synchronization follows a three-tier hierarchy illustrated in Fig. 10. At the global tier, GNSS disciplines each vehicle's local clock to UTC, providing fleet-wide temporal coherence and enabling post hoc event correlation even without direct intervehicle communication [72]. At the local tier, 5G C-V2X PC5 sidelink with TSC links provides fast bounded-latency intervehicle exchanges aligned to GNSS or gNB-distributed time. At the intravehicle tier, automotive TSN maintains sub-microsecond synchronization among LiDAR, electronic control units (ECUs), and actuators as described in Section VI-A.

GNSS vulnerabilities, urban canyons, tunnels, jamming, and spoofing require holdover strategies using high-stability oscillators and opportunistic assistance from cellular timing or infrastructure beacons to bound drift during outages [73]. The early maturity of deterministic

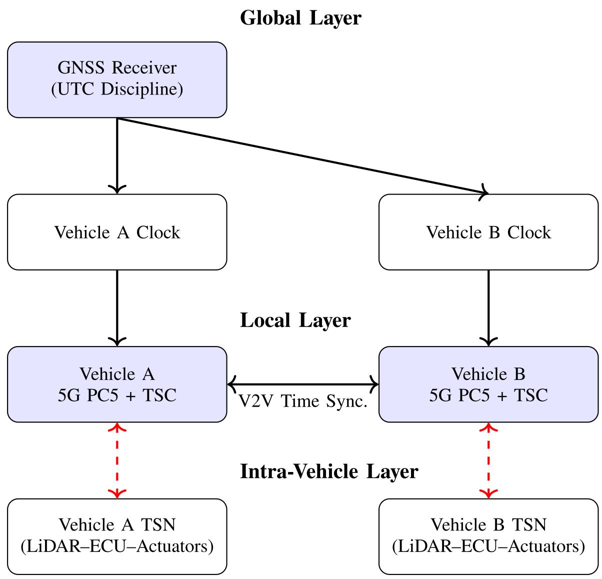

Fig. 10. ***Hierarchical time model for V2X: GNSS provides global UTC alignment; 5G PC5/TSC enables rapid intervehicle coordination; in-vehicle TSN ensures sub-μs determinism among sensors and actuators.***

TSC slicing and multivendor timing interoperability across GNSS, TSN, and 5G subsystems remains the primary open challenges.

D. Infrastructure Monitoring: Smart Grids and Sensor Networks

Infrastructure monitoring spans a wide precision tier, from sub-microsecond synchrophasor measurement in power grids to millisecond environmental sensing in industrial and urban deployments, but shares a common synchronization philosophy: precise timestamps at data sources enable spatially distributed measurements to be correlated in time for anomaly detection, fault isolation, and predictive control.

Smart grids impose the tightest timing budgets among infrastructure monitoring applications. PMUs generate synchrophasors from UTC-stamped voltage and current samples requiring sub-microsecond alignment; 10-μs skew produces degree-level phase error that masks oscillations and distorts wide-area stability analysis [74]. Remedial action scheme (RAS) and special protection system (SPS) must detect, isolate, and reconfigure faults within 10 ms across geographically separated substations. SCADA systems tolerate millisecond-to-second latency but benefit from consistent timestamps for postevent forensics and compliance. Fig. 11 illustrates these tiered requirements.

The synchronization substrate for smart grids combines GNSS-disciplined GMs providing UTC-aligned 1 PPS at the substation level with IEEE 1588 PTP and gPTP distributing sub-microsecond time to PMUs, merging units, and intelligent electronic device over fiber or TSN-capable Ethernet, as described in Section IV-A. Wireless augments this where cabling is impractical: 5G TSC links, described in Section V-E, support portable PMUs and time-aware distributed energy resource coordination at sub-millisecond accuracy under GNSS-disciplined gNBs [59]. LPWAN technologies, described in Section V-B, serve low-rate telemetry with timestamps applied at GNSS-disciplined gateways. The practical tiered design maps sub-microsecond synchronization to protection and synchrophasor systems, millisecond-class wireless to mobile and remote assets, and loosely synchronized LPWAN to noncritical telemetry.

WSNs and environmental monitoring occupy the relaxed end of the infrastructure monitoring spectrum. Industrial condition monitoring, vibration, thermal, and acoustic sensing for predictive maintenance, requires millisecond-level coherence across sensor streams to avoid spurious fault correlations. Urban sensing deployments such as traffic cameras and acoustic monitors feeding curbside edge clusters for accident detection require millisecond GNSS or 5G TSC synchronization to improve cross-modal correlation and reduce false positives. Environmental monitoring networks, air quality, flood detection, and precision agriculture tolerate second-level alignment, with timestamps enforced at GNSS-disciplined gateways rather than at individual sensor nodes. IEEE 802.15.4/TSCH and BLE technologies described in Section V-A serve these deployments, with LPWAN from Section V-B covering wide-area deployments.

The fundamental challenge across both tiers is GNSS vulnerability, spoofing, jamming, and urban canyon multipath can induce timestamp drift that propagates through the entire synchronization hierarchy, and resilient holdover strategies using high-stability oscillators are essential; these are examined in Section VIII.

E. Wide-Area IoT and Precision Agriculture

Wide-area IoT deployments prioritize geographic reach, multiyear device lifetime, and massive node density over synchronization precision. Representative applications include precision agriculture, soil moisture, microclimate, and irrigation monitoring across kilometer-scale fields, distributed energy metering, environmental sensing networks for flood and air quality monitoring, and wide-area asset tracking across logistics and supply chains. These applications share a common tolerance for second-level

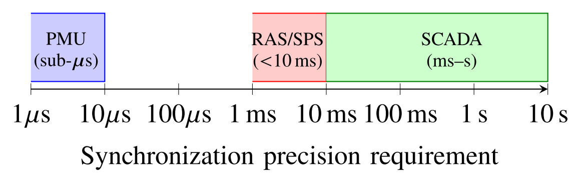

Fig. 11. ***Smart grid synchronization precision tiers: PMUs require sub-μs alignment for synchrophasor accuracy; RAS/SPS demand $<$10 ms for fault isolation; SCADA tolerates ms–s for monitoring and forensics.***

timestamp alignment at end devices, provided that events can be correlated across sites for supervisory analysis and anomaly detection.

End devices in these deployments operate as free-running nodes transmitting sporadically over LPWAN links, as described in Section V-B. Precise UTC-aligned timestamping is enforced at GNSS-disciplined gateways rather than at individual sensors, as described in Section IV-A. This architecture enables forensic-grade event correlation across geographically distributed gateways, for example, correlating soil moisture anomalies across a field, or reconstructing the sequence of fault events across a distribution network, without requiring tight synchronization at the sensor tier [54]. In more advanced deployments, multiple LPWAN gateways are each disciplined to UTC via GNSS, allowing cross-site correlation at the gateway timestamp granularity of tens of nanoseconds even when end-device clocks drift freely at second-level accuracy [75].

The fundamental characteristic of this application domain is that synchronization is a gateway-level concern rather than a network-wide concern; end-to-end timing coherence is reconstructed in postprocessing from gateway timestamps rather than maintained continuously across all nodes. This makes wide-area IoT structurally incompatible with deterministic real-time control but well-suited to large-scale supervisory monitoring where latency tolerance is measured in seconds to minutes. Open challenges including GNSS availability in remote deployments and energy-constrained holdover strategies are examined in Section VIII.

VII. TOOLS, TESTBEDS, AND DATASETS

Rigorous evaluation of wireless synchronization protocols requires controlled, instrumented environments capable of capturing timing deviations on the microsecond or even nanosecond scale. Over the past decade, a diverse ecosystem of software simulators, hardware testbeds, and public datasets has emerged to support this need. These resources enable researchers to analyze synchronization accuracy, convergence, scalability, and robustness under varying interference, topology, and mobility conditions across technologies such as Wi-Fi 6/7, 5G TSC, UWB, BLE, and TSCH. Table 4 summarizes representative tools, testbeds, and datasets discussed in this section, organized by synchronization paradigm, to support tool selection for a given research context.

A. Simulation Frameworks

Simulation frameworks remain the primary tools for early stage evaluation of wireless synchronization algorithms, allowing researchers to study clock offset, drift, jitter, and scheduling effects without requiring costly physical deployments. By abstracting hardware while maintaining temporal fidelity, these simulators support repeatable, scalable experimentation across technologies.

1) OMNeT++/INET/Simu5G: OMNeT++ is a modular, event-driven simulator widely used for deterministic networking research [76]. With the INET framework, it supports IEEE 802.1AS (gPTP) and IEEE 802.1Qbv (time-aware shaping), enabling fine-grained analysis of TSN timing behavior and gate-based scheduling. Actively maintained through OMNeT++ 6.3 and INET 4.6, the toolchain was presented in a dedicated TSN tutorial at the IEEE Factory Communication Systems Conference in 2025, underscoring its status as the community reference for wired and wireless TSN simulation. The Simu5G extension further models 5G NR, making it suitable for hybrid TSN–5G synchronization studies and time-sensitive flow management in industrial or vehicular networks [77], [94]. The INET-based open-source time synchronization simulator presented in [95] and [96] uses the 6G DetCom node with uplink and downlink delays from the PD-Wireless-5G-1 dataset. This enables the analysis of standards-compliant time synchronization protocols for TSN and 5G-TSN networks.

2) NS-3: It is a discrete-event simulator implemented in C++ with Python bindings, released as version 3.44 in early 2025 [78]. It offers high configurability for modeling Wi-Fi MAC behavior, OFDMA scheduling, and 5G V2X PC5 sidelink communication [97]. Researchers extend it with custom clock models to emulate offset correction, GNSS-like disciplining, or PTP message exchanges, balancing scalability with timing precision.

3) 6TiSCH Simulator: It is a Python-based discrete-event simulator developed as part of the IETF 6TiSCH standardization activity [79]. It implements the full 6TiSCH protocol stack, TSCH scheduling, IPv6 over low-power wireless personal area networks (6LoWPANs), and the routing protocol for low-power and lossy networks (RPL), with a configurable per-node clock drift model (PPM), guard-time management, and supports the same $k7$ trace-based radio link model used in OpenBenchmark [84], enabling direct simulation-to-testbed comparison on FIT IoT-LAB and w-iLab.t. The simulator scales to networks of hundreds of nodes and has been used extensively within the IETF working group to benchmark 6TiSCH against realistic deployment scenarios for slot-frame design, guard-time optimization, and synchronization overhead evaluation.

In practice, many studies adopt a two-tier evaluation strategy: drift and offset models are first validated in MATLAB/Simulink or Python-based frameworks (e.g., SimPy with Kalman or least-squares correction filters [26], then embedded into OMNeT++ or NS-3 to assess end-to-end effects on network scheduling, congestion, and synchronization convergence. This approach provides both theoretical fidelity and network-level realism.

In summary: the 6TiSCH simulator is the reference for TSCH synchronization and guard-time studies; OMNeT++/INET is preferred for TSN gate scheduling and 5G-TSC integration; and NS-3 is preferred for Wi-Fi 6/7 MAC and vehicular side-link scenarios.

Table 4 Representative Tools, Testbeds, and Datasets for Wireless Synchronization Research

Name	Type	Sync. Paradigm	Key Sync. Feature	Status	Ref.
Simulation Frameworks					
OMNeT++/INET	Sim	TSN/PTP, gPTP, 5G-TSC	IEEE 802.1AS, Qbv gate scheduling	Active (v6.3)	[76]
Simu5G	Sim	5G NR, TSC, URLLC	5G slicing, TSN–5G bridge	Active (2024)	[77]
ns-3	Sim	Wi-Fi, V2X, 5G NR	Custom clock models, PC5 sidelink	Active (v3.44)	[78]
6TiSCH Simulator	Sim	TSCH / 6TiSCH	Per-node ppm drift, guard-time mgmt	Active (IETF)	[79]
Academic and Open Testbeds					
FIT IoT-LAB	Testbed	IoT/WSN, TSCH, BLE	2700+ nodes; multi-site low-power	Active (SLICES-RI)	[80]
ORBIT (WINLAB)	Testbed	Wi-Fi, LTE, SDR	Radio-grid; configurable topology	Active	[81]
COSMOS (Manhattan)	Testbed	5G, mmWave, edge	NSF PAWR; gPTP over 5G fronthaul	Active	[81]
POWDER (Salt Lake City)	Testbed	5G NR, TSC, SDR	NSF PAWR; 5G-TSC outdoor slicing	Active	[82]
Colosseum	Emulator	SDR, 5G, O-RAN	256 USRPs; HiL; precision timing servers	Active (NSF)	[83]
OpenBenchmark	Platform	TSCH / 6TiSCH	Reproducible KPI; IoT-LAB + w-iLab.t	Active	[84]
Industrial, Grid, and V2X Testbeds					
Siemens / NI / Fraunhofer	Lab	TSN + Wi-Fi 6/7, 5G-TSC	μs wired–wireless offset	Active	[85]
5G-ACIA Labs	Lab	5G TSC, NW-TT/DS-TT	TSN Translator interoperability	Active	[86]
NREL / TU Dortmund	Lab	PTP, PMU, IEC 61850	GNSS holdover, spoofing resilience	Active	[87]
5G-MOBIX	Field	5G V2X, GNSS	Cross-border V2X timing trials	Data public	[88]
Smart Mobility Corridor	Field	V2X, GNSS, 5G	Sub-ms V2X timing; tunnel holdover	Active	[89]
Open Datasets					
5G-MOBIX V2X dataset	Dataset	5G V2X, GNSS	GNSS offset + PC5 jitter traces	Available	[88]
ORNL/PNNL synchrophasor	Dataset	PMU / IEC 61850	GPS dropout + clock-error annotation	Available	[90], [91]
NASPI / PNNL PMU logs	Dataset	Synchrophasor, GNSS	Timestamped V/I + sync.-quality meta	Available	[74]
IEEE DataPort WSN/TSCH	Dataset	TSCH / 6TiSCH	Slot latency, drift, OpenMote B	Available	[92]
LoRaWAN Class-B traces	Dataset	LoRaWAN / LPWAN	GPS/UTC leap-second misalignment	Available	[93]

B. Experimental Testbeds and Field Pilots

While simulations offer controlled insights into drift and scheduling effects, real-world testbeds remain essential for exposing synchronization systems to radio noise, GNSS variability, and multivendor interoperability challenges [58], [98].

1) Academic and Open Research Platforms: Large-scale shared facilities provide reproducible experimentation with programmable radios and distributed sensing. The **FIT IoT-LAB** in France offers over 2700 heterogeneous low-power nodes across six sites for wireless and IoT synchronization research, and is now integrated into the SLICES-RI pan-European research infrastructure [80]. The **ORBIT** radio-grid testbed at Rutgers University (WINLAB), operational since 2005, supports over 1000 registered users for configurable interference and topology experiments, and has been federated with the city-scale **COSMOS** testbed in Manhattan, an NSF PAWR deployment focused on ultrahigh-bandwidth, low-latency wireless tightly coupled with edge-cloud computing [81]. The **POWDER** platform in Salt Lake City further extends this ecosystem to 5G-TSC slicing and gPTP time-transfer studies in outdoor environments [82].

Colosseum, operated by Northeastern University through an NSF grant and originally built by DARPA for the Spectrum Collaboration Challenge, is the world's largest hardware-in-the-loop wireless network emulator [83]. It combines 256 USRP X310 software-defined radios with an FPGA-backed massive channel emulator capable of reproducing up to 256 × 256 independently configurable wireless channels in real time. Colosseum includes precise timing servers that synchronize all SDR nodes, making it well suited for studying interference-aware synchronization, AI/ML-driven clock correction, and O-RAN time-sensitive flows at scale. In 2024, Colosseum integrated the **SLICES-RI Beyond-5G Blueprint**, a pan-European ESFRI research infrastructure covering 16 countries, enabling reproducible 5G and post-5G experiments across federated European and U.S. platforms [99]. The Blueprint mandates 3GPP-compliant PTP references across the DU-RU split, making SLICES-RI directly relevant to 5G TSC synchronization research.

OpenBenchmark [100], developed by the University of Montenegro and Inria, is a cloud-based benchmarking service that automates the full 6TiSCH experiment lifecycle across FIT IoT-LAB and w-iLab.t: it provisions testbed nodes, flashes firmware, instruments it in real time, and collects key performance indicators (reliability, latency, radio duty cycle, and network formation time). By sharing scenario definitions and raw KPI traces alongside results, OpenBenchmark enforces reproducibility for TSCH synchronization and scheduling studies; it is the testbed counterpart to the 6TiSCH simulator and the IEEE DataPort WSN/TSCH dataset.

2) Industrial TSN and 5G Evaluation Laboratories: Vendors and applied research institutes—including Siemens, NI, Fraunhofer IIS, and the 5G-ACIA consortium—operate hybrid TSN testbeds that merge wired deterministic Ethernet with synchronized wireless overlays [85], [86], [101]. These facilities instrument microsecond-level offsets across mixed wired–wireless domains, analyzing how gPTP and 802.1Qbv scheduling behave under Wi-Fi 6/7 or pre-standard 5G TSC operation. Evaluation scenarios typically target motion control, mobile tooling, or inspection robots roaming between wired docks and wireless access zones, measuring how handover and resynchronization affect microsecond budgets.

3) 5G TSC Field Pilots: Telecom and industrial partners—including Ericsson, Nokia, and Huawei—are conducting field pilots that couple GNSS-disciplined 5G base stations with TSN-enabled factories [59], [102]. These trials validate sub-μs synchronization over realistic radio channels, evaluating bounded-jitter performance under fading and mobility, and testing TSN translator (NW-TT/DS-TT) interoperability. Results confirm that TSC links can sustain μs-scale alignment under controlled loads, while maintaining accuracy under dense interference or high-mobility handovers remains an open engineering challenge.

4) Power and Utility Grids: Utility R&D facilities and university laboratories—such as NREL and TU Dortmund—deploy substations with GPS-disciplined clocks, PMUs, and IEC 61 850 merging units to study synchronization robustness under GNSS outages, jamming, or spoofing [87]. Experiments focus on holdover stability, phase drift under timing loss, and resilience of TSN/PTP-based protection relays, providing the empirical foundation for fault-tolerant grid synchronization architectures.

5) Vehicular and V2X Test Tracks: Automotive test sites (e.g., UTAC, CERM, and U.S. Smart Mobility Corridor) combine GNSS-disciplined vehicles, TSN-enabled in-vehicle networks, and 5G C-V2X PC5 radios to evaluate time coordination during motion [89], [103]. Field measurements assess offset propagation across hops, holdover stability in tunnels or urban canyons, and synchronization drift during handover—ensuring that cooperative braking and perception tasks maintain sub-millisecond coherence even under dynamic multipath conditions.

6) Cross-Domain Insights: Across these testbeds, a consistent theme emerges: achieving deterministic time transfer across heterogeneous links requires hybrid synchronization hierarchies that combine GNSS anchoring, wired TSN backbones, and wireless overlays. This architectural pattern, validated in industrial, vehicular, and utility trials alike, directly applies the design principles of Sections IV-A–V-E and the tradeoff framework of Section III-F.

For tool selection: Colosseum suits large-scale interference-aware and AI/ML synchronization studies; FIT IoT-LAB and ORBIT/COSMOS suit protocol level and scalability studies, with OpenBenchmark providing the reproducibility layer for TSCH; industrial labs suit mixed wired–wireless TSN integration; 5G field pilots suit NW-TT/DS-TT interoperability; utility testbeds suit GNSS holdover and spoofing resilience; and V2X tracks suit cross-domain timing under mobility.

C. Open Datasets and Measurement Campaigns

Simulations and controlled testbeds cannot fully capture the environmental and hardware-induced variability that affects wireless synchronization in practice. Open datasets and long-term measurement campaigns bridge this gap, providing empirical traces of real clock behavior under conditions such as multipath fading, GNSS interference, oscillator aging, and mobility. These resources support reproducible evaluation, cross-protocol comparison, and ML-based prediction of synchronization anomalies.

1) Representative Datasets: Several publicly available datasets document timing behavior across heterogeneous technologies. The **5G-MOBIX V2X** dataset [88], produced by the EU Horizon 2020 project of the same name, provides cross-border vehicular traces combining GNSS time offsets with PC5 sidelink latency and jitter measurements collected during real cross-border driving trials in Europe. For grid synchronization, the **ORNL Grid Event Signature library** [90], [91] is an open-access repository of two years of field synchrophasor measurements from multiple U.S. utilities, curated with labeled timing anomalies including GPS signal loss, internal clock errors, and quantization artifacts. Complementing this, NASPI and PNNL publish PMU logs with timestamped voltage and current data and synchronization-quality metadata, enabling study of phase-angle drift, GNSS outage effects, and timing fault resilience. For TSCH-based industrial networks, the **IEEE DataPort WSN/TSCH dataset** [92] provides slot-level reliability, latency, and channel-hopping traces collected from real OpenMote B devices running the 6TiSCH protocol, including analysis scripts for timing performance indices. The dataset is the natural empirical complement to the 6TiSCH simulator and OpenBenchmark, closing the simulation-to-hardware validation loop for TSCH synchronization studies.

For LPWAN synchronization, the **LoRaWAN Class-B beacon timing dataset** [93] provides over-the-air LoRaWAN traces captured across four European cities using a custom GPS-disciplined open-source sniffer. The dataset documents Class-B beacon timing including a systematic 18-sUTC/GPS leap-second misalignment present in a significant fraction of real-world deployments, a synchronization failure mode that had not previously been characterized at scale. Raw captures, a Wireshark dissector, and postprocessing scripts are publicly available, making this the primary open dataset for evaluating LPWAN Class-B synchronization accuracy and compliance.

2) Utility and Research Use Cases: Empirical traces underpin diverse research objectives across the application domains of Section VI. Vehicular datasets from 5G-MOBIX and related projects record GNSS and PC5 timing deviations during real driving, supporting evaluation of platooning stability and GNSS-recovery behavior. Grid PMU logs enable characterization of holdover degradation and timing-fault propagation in IEC 61850 protection systems. The IEEE DataPort TSCH traces and LoRaWAN Class-B captures together span the IoT/LPWAN timing-space, enabling comparison of μs-class TSCH precision against the second-scale challenges of wide-area LPWAN synchronization. For all application classes, combining empirical datasets with simulation replay in OMNeT++ or the 6TiSCH simulator supports cross-layer benchmarking under identical temporal disturbances.

3) Benchmarking and Reproducibility: Trace-based analysis enables validation of clock models, protocol comparison, and training of anomaly detectors or holdover estimators. By releasing raw logs, configuration files, and analysis scripts alongside measurement results, recent projects promote the reproducible synchronization research advocated in open-science frameworks. As the community moves toward standardized benchmarking, datasets that pair timing traces with annotated ground truth, covering synchronization events, GNSS dropouts, and resynchronization transients, will be essential for enabling fair comparison of time-aware wireless protocols.

VIII. CHALLENGES AND FUTURE DIRECTIONS

Wireless clock synchronization has evolved into a mature yet rapidly diversifying research field. Across domains such as industrial automation, vehicular networks, XR, and smart grids, synchronization now underpins not only communication but also sensing, control, and actuation. Despite the progress outlined in Sections V–VII, several open challenges remain before wireless systems can achieve deterministic, self-healing, and globally consistent timing comparable to wired TSN infrastructures.

This section synthesizes these challenges and outlines emerging research directions across five dimensions: 1) fundamental technical challenges of precision, robustness, and scalability; 2) distributed and consensus-based synchronization for infrastructure-free operation; 3) intelligent, self-healing architectures that leverage ML; 4) secure and resilient synchronization against wireless-specific threats; and 5) cross-domain convergence encompassing TSN–5G/6G interoperability, joint communication-sensing-timing, and global time fabrics.

A. Fundamental Technical Challenges

Achieving microsecond- or nanosecond-level alignment in large, heterogeneous, and mobile wireless networks remains an unsolved systems problem. Variability in oscillator stability, multihop propagation, handover behavior, and channel interference collectively limit current synchronization methods. As applications expand toward dense multicell factories, vehicular platoons, and distributed XR environments, scalability, mobility, robustness, and ultralow-jitter operation emerge as critical frontiers.

1) Scalability and Heterogeneous Environments: Future industrial and IoT systems will comprise tens of thousands of interconnected devices spanning multiple physical and logical domains—wired TSN backbones, Wi-Fi 6/7 access networks, 5G TSC links, BLE or TSCH meshes, and UWB-based ranging infrastructures. Maintaining submicrosecond coherence at this scale challenges classical hierarchical clock-tree models designed for static, homogeneous topologies. In large, mixed deployments, synchronization errors arise not only from propagation and queuing delays but also from the fundamental differences in timestamp granularity, oscillator stability, and update frequency among technologies.

Heterogeneity is a primary stressor on time discipline. Each network technology—TSN, Wi-Fi, 5G, BLE, TSCH, or UWB—embodies distinct assumptions about delay variation, message periodicity, and jitter tolerance. When these domains interconnect, the lack of consistent time semantics produces subtle but consequential misalignments. A TSN bridge, for instance, schedules egress gates assuming deterministic arrivals, while Wi-Fi suffers stochastic contention and retransmissions that shift packet timing. A 5G TSC cell synchronized to a GNSS-disciplined gNB may interface with a TSCH cluster whose slot-frame synchronization drifts as nodes join or leave. Such discrepancies can desynchronize multisensor fusion, distort control loops, or compromise safety-critical actuation in cooperative robotic or vehicular systems. The challenge is not only a single protocol's failure but also the absence of cross-domain coordination that reconciles different notions of time accuracy, granularity, and reliability.

Scalable synchronization in these conditions demands architectures that are adaptive rather than rigidly hierarchical. A promising paradigm is federated multisource time discipline, where each node maintains and blends multiple timing references—GNSS, local PTP or TSN domains, peer-to-peer exchanges, and oscillator-based prediction—rather than depending on a single GM [58]. Synchronization becomes a multi-input estimation process in which sources are weighted by runtime confidence metrics such as GNSS signal quality, local network stability, or peer-to-peer consistency. During GNSS outages, for example, nodes can rely on local PTP timing and predictive drift models to preserve alignment; when a PTP domain becomes unstable under congestion, trust can shift toward peers or disciplined oscillators. This federated approach enables graceful degradation and self-recovery, maintaining relative coherence even when individual timing paths fail.

Resilience under churn is another critical dimension. Industrial and IoT networks are inherently dynamic, wireless nodes may sleep, reboot, or hand over between APs; TSN routes and 5G schedules may reconfigure as traffic patterns evolve. Next-generation synchronization frameworks must support fast local convergence, fault containment, and smooth reintegration. Unstable or drifting segments must be detected and quarantined through continuous offset monitoring and jitter analysis, then gradually realigned to the global time base to avoid phase discontinuities. Cross-layer coordination between timing protocols and network-management logic, rerouting synchronization paths, validating time quality, and triggering holdover strategies is essential to achieve without operator intervention. Model-based and digital-twin approaches are being actively explored to predict each device's clock behavior, maintaining time with far fewer message exchanges and thereby cutting overhead in large IIoT deployments while coping with diverse oscillator qualities and delay patterns [4], [67]. In addition, many deployments cannot host expensive disciplined oscillators or run GPS continuously, given energy and cost constraints. Here, hybrid architectures that duty-cycle high-quality references (e.g., GPS ON/OFF) combined with strong prediction/filters, and better exploitation of existing protocol clocks (Wi-Fi TSF and BLE timers) for fine-grained skew corrections [104], [105] could be the answer.

A related but distinct open challenge concerns the coordination of traffic patterns across the TSN–5G boundary. In integrated TSN–5G deployments, periodic traffic patterns—characterized by parameters such as the burst arrival time (BAT), which defines the exact start and duration of periodic traffic flows—must be transferred from the TSN domain to the 5G system, typically from the centralized network configuration to the 5G application function. This transfer currently relies on per-stream filtering and policing (PSFP) support on both sides, and the precision of this mechanism remains questionable. Without accurate traffic pattern transfer, 5G scheduling cannot precisely allocate resources, causing traffic to accumulate in buffers and undermining the determinism that TSN guarantees. Conversely, while the RAN can detect misalignment between incoming traffic patterns and scheduled resources and estimate the resulting offset, no standardized mechanism currently exists to propagate this offset back to the TSN domain so that end devices or applications can adjust their sending times accordingly [106].

2) High-Precision Synchronization Under Mobility: Maintaining microsecond-level synchronization under mobility remains one of the hardest open problems in time-sensitive wireless systems. When nodes move (vehicles forming platoons, drones flying in swarms, or AGV roaming factories) propagation delays fluctuate, Doppler shifts perturb oscillators, and handovers disrupt timing continuity across APs and radio domains. These effects cause offset spikes, jitter bursts, and sometimes complete loss of lock, undermining the deterministic guarantees expected by TSN-class systems.

Table 5 Typical Timing Challenges in Mobile Scenarios

Scenario	Typical Timing Problem
Wi-Fi handover between APs	Access points in distinct gPTP domains cause temporary misalignment and jitter during roaming events.
5G NR mobility across gNBs	State transfer and rescheduling create multi-ms timing gaps even in TSC links.
GNSS visibility loss	Oscillator drift accumulates in holdover, degrading phase coherence within seconds.
Drone or AGV movement	Rapid distance changes alter propagation delay and Doppler shift, disturbing delay compensation models.

The root of the problem lies in the mismatch between the assumptions of existing time-distribution protocols and the realities of mobility. Mechanisms such as gPTP depend on stable round-trip delay measurements, conditions rarely met when radios experience fading, shadowing, or cell transitions. In Wi-Fi networks, roaming across APs disciplined by slightly different time masters can lead to transient offset discontinuities. In 5G, mobility events require state transfer and new scheduling windows across gNBs that may belong to distinct synchronization domains. Table 5 summarizes typical synchronization stress points in representative mobile scenarios.

Several research directions address these challenges. *Predictive synchronization handover* aims to prestage timing adjustments before mobility events occur. Using trajectory or path planning, APs and base stations can align to a common reference in advance, minimizing phase discontinuities during roaming. *Fast redisciplining protocols* accelerate recovery by dynamically increasing synchronization message frequency immediately after a handover or by transferring state variables, such as offset estimates and filter coefficients, between nodes. *Mobility-aware hybrid holdover* blends multiple timing references to sustain accuracy during transient losses: mobile nodes can combine short-term holdover with peer-to-peer correction from nearby nodes that still maintain GNSS or PTP discipline, or apply predictive models trained on historical drift patterns. Such hybrid methods have been shown to limit offset growth to below 100 μs even through several-second GNSS gaps. Recent developments in joint waveform-receiver design that explicitly models time-varying multipath and antenna group delay variation, and robust estimators allow for sub-nanosecond accuracy under mobility [107].

Evaluating synchronization under mobility calls for *mobility-sensitive metrics* that quantify temporal resilience: the duration of "time holes" during which offset exceeds tolerance thresholds, the maximum offset spike following a handover, and the recovery latency required to return within acceptable jitter bounds. For applications such as cooperative perception or distributed haptics, even brief

Table 6 Challenges in Extending TSN Jitter Guarantees Over Wireless Links

Wireless Mechanism	Impact on Synchronization
OFDMA and MU-MIMO in Wi-Fi 6/7	Dynamic sub-channel allocation can create microburst jitter when clients change resource units between frames.
5G TSC mobility and handovers	Buffer transfer and rescheduling across gNBs cause transient offset spikes of several μs.
RF link variability and HARQ retransmissions	SNR fluctuations trigger retries that violate TSN-style deterministic gating and periodic scheduling.
MAC contention (CSMA/CA)	Random backoff delays cause unpredictable inter-packet timing, undermining sub-μs gate precision.

1–2-ms desynchronization can cause perceptible misalignment or unsafe actuation.

3) Ultralow-Jitter and Deterministic Wireless Synchronization: As wireless domains converge with TSN and other time-aware frameworks, the need for **ultralow-jitter synchronization**, below 100 ns and ideally approaching tens of nanoseconds, is becoming a defining challenge. Industrial automation, cooperative robotics, haptic XR, and distributed sensing increasingly depend on consistent sub-microsecond timing to preserve determinism across hybrid wired–wireless infrastructures. While wired TSN already achieves sub-μs synchronization through IEEE 802.1AS and PTP profiles, extending such guarantees over Wi-Fi 6/7 and 5G TSC remains one of the most difficult frontiers in TSN.

The difficulty arises from the fundamental variability of wireless media. Channel fading, mobility, contention, and retransmissions introduce nondeterministic delay variations that cannot be fully eliminated by existing MAC scheduling or retransmission control. Even small deviations in packet timing, on the order of hundreds of nanoseconds, can misalign actuator triggers, corrupt multisensor fusion pipelines, or degrade immersive XR experiences. Table 6 summarizes how each major wireless mechanism contributes to synchronization jitter.

Achieving deterministic timing over inherently stochastic channels demands **cross-layer synchronization control**. Recent studies propose architectures where PHY- and MAC-layer metrics, SNR, queue occupancy, or retransmission count, are exposed to TSN shapers, allowing dynamic gate adjustment and *proactive slot realignment* before temporal violations occur [108], [109]. ML-based temporal models can learn correlations between channel states, mobility patterns, and delay variations, enabling preemptive adjustments to TSN schedules or 5G link configurations. Simulation studies indicate that predictive gating can substantially reduce peak jitter compared with static scheduling in variable-load environments [110], [111].

Table 7 summarizes illustrative open research questions that follow from these ultra-low-jitter synchronization challenges. Realizing ultralow-jitter synchronization over wireless will require deep cross-layer integration: real-time PHY state monitoring to dynamically reshape gate schedules; MAC-level determinism through scheduled access and bounded HARQ latency; AI-guided local timing agents that autonomously adapt to jitter patterns; and hybrid analog–digital correction loops combining hardware timestamping with digital compensation filters to achieve below-100 ns stability even under interference. Together, these developments point toward **wireless deterministic networks** capable of delivering TSN-class precision without sacrificing flexibility or mobility.

4) Robust Clock Parameter Estimation: Standard maximum-likelihood estimator (MLE) for clock offset and skew assumes that network delays follow Gaussian or exponential distributions, an assumption grounded in queuing theory but routinely violated in wireless deployments. Channel fading, MAC contention, HARQ retransmissions, and cross-technology interference produce delay distributions that are heavy-tailed, multimodal, or time-varying, causing MLE-based synchronization to incur substantial estimation bias [17], [18]. A single anomalous delay sample, arising from a collision or a sudden propagation transient, can destabilize a conventional Kalman or proportional integral servo for several update cycles, temporarily driving offset beyond acceptable bounds.

Robust estimation addresses this by designing clock parameter estimators whose performance degrades gracefully under nonnominal delay distributions. *M-estimators and bounded-influence methods* replace the squared residuals of least-squares fitting with loss functions (e.g., Huber and Tukey bisquare) that down-weight outlying timestamp observations, effectively limiting the influence of any single corrupted measurement on the offset or skew estimate [15]. *Composite particle filters* and bootstrap-sampling variants extend this idea to nonparametric settings, achieving near-optimal performance across asymmetric Gaussian, exponential, Gamma, and Weibull delay models without requiring prior knowledge of the distribution shape [17]. For PTP in multipath environments, expectation–maximization approaches can identify asymmetric master–slave communication paths and discard them before applying a mini–max optimum phase-offset estimator to the remaining symmetric paths.

Several open problems remain. First, the joint estimation of offset, skew, and asymmetry under time-varying delay distributions has no closed-form solution, and computationally efficient approximations suitable for embedded clock servos are still an active research area. Second, robust estimators designed for two-node exchanges do not extend straightforwardly to multihop topologies where correlated delay patterns propagate across the synchronization tree. Third, the interaction between robust estimation and adaptive correction algorithms (e.g., PI controllers and Kalman filters) has not been fully characterized: a robust front-end may suppress legitimate large corrections that are needed during handovers, creating a

Table 7 Illustrative Open Questions for Ultralow-Jitter Wireless Synchronization

Open Question	**Why It Matters**
How to maintain <1 μs jitter over Wi-Fi 7 or 5G TSC?	Required for robotic coordination and motion control spanning wired and wireless TSN segments.
Can ML-based schedulers anticipate RF-induced microbursts?	Enables preemptive slot shifting and resource reservation to prevent time violations.
What trade-offs exist between HARQ reliability and jitter bounds?	Essential for haptic feedback and closed-loop control where late packets are worse than losses.
How to measure and mitigate cross-technology jitter drift (TSN→5G→Wi-Fi)?	Needed for end-to-end time continuity across multi-radio hybrid infrastructures.

tension between outlier rejection and fast redisciplining. Resolving these tensions is central to achieving reliable sub-microsecond timing in the heterogeneous delay environments described in Section II-B.

B. Distributed and Consensus-Based Synchronization

Classical synchronization protocols such as PTP and TSCH assume the existence of a time reference to which all others align. While this hierarchical model is effective in stable, well-connected deployments, it creates a single point of failure and scales poorly in infrastructure-free or partitioned networks such as drone swarms, vehicular platoons, disaster-response sensor meshes, and disconnected industrial subnetworks. *Distributed and consensus-based synchronization* addresses this by deriving a common time reference from collective agreement among peers, without any designated authority.

1) Pulse-Coupled Oscillators: Inspired by the spontaneous synchronization of biological oscillators, most famously the synchronized flashing of fireflies described by Mirollo and Strogatz [112], PCO schemes model each network node as a phase oscillator that fires a brief pulse when its phase reaches a threshold. Upon receiving a pulse, each node adjusts its own phase by a coupling function, and under appropriate conditions the network globally converges to a common firing phase without any exchange of explicit timestamps. Practical implementations on MicaZ and OpenMote platforms have demonstrated microsecond-class synchronization accuracy in networks of tens of nodes, with energy consumption substantially lower than packet-based protocols due to the absence of timestamp messages [113]. A key challenge is robustness to propagation delay: because a PCO scheme conflates an incoming pulse's reception time with the sender's phase, even small and variable delays distort convergence. Recent work addresses this through delay-compensated coupling functions and resilient mean subsequence reduced algorithms that tolerate a bounded number of faulty or adversarial nodes [114]. PCO networks face significant challenges when moving from idealized theory to real-world settings: classical assumptions of identical oscillators and negligible delays break down under heterogeneity, propagation delays, and complex topologies, making global synchrony and clustered firing only partially understood under realistic conditions [115]. Resilient synchronization methods inspired by mean subsequence reduced techniques can filter malicious behavior, yet lack tight theoretical guarantees, efficient communication bounds, and robustness to asynchronous or lossy channels [116].

2) Average Consensus and Diffusion Algorithms: Average-consensus algorithms operate on explicit clock readings. Each node iteratively updates its local time estimate as a weighted average of its own value and those received from neighbors, causing all estimates to converge to the network-wide mean. Diffusion-based extensions allow each node to maintain an independent estimate updated only from one-hop neighbors, achieving convergence without global coordination. The primary open problem is drift: average-consensus on clock readings corrects phase offset, but not skew, so relative drift between oscillators causes the common estimate to degrade over time. Second-order consensus algorithms that jointly estimate both offset and rate provide a solution at the cost of increased state and communication [37]. Average consensus and diffusion-based time synchronization face fundamental challenges in balancing convergence speed, robustness, accuracy, and resource efficiency in realistic networks: iterative schemes incur high communication and energy costs, especially in large multihop systems [117], while acceleration techniques (e.g., multihop links or optimized weights) introduce design complexity and sensitivity to topology changes [39], [118]. Time-varying, directed, and noisy links with random delays can degrade accuracy or even cause divergence, and although delay-compensation, skew estimation, and event-triggered updates improve stability [39], [119], they rely on restrictive assumptions and lack unified robustness guarantees. Even when convergent, many methods suffer from steady-state bias under dynamic conditions, and dynamic average consensus introduces additional tradeoffs between tracking accuracy, stability, memory, and communication overhead [120], [121]. Security, privacy, and learning-enhanced approaches further complicate design: randomized consensus can preserve privacy without sacrificing accuracy, and learning-based adaptations improve performance in harsh environments, but both lack strong theoretical guarantees and clear impacts on synchronization quality [122].

3) Byzantine-Resilient Consensus: In adversarial environments, multioperator industrial networks, contested vehicular corridors, or security-sensitive critical infrastructure, consensus algorithms must tolerate nodes that transmit incorrect or strategically crafted timing information. Byzantine fault-tolerant consensus protocols guarantee that a correct node reaches agreement with other correct nodes even when up to f out of n nodes behave arbitrarily, provided $n > 3f$. Adapting these guarantees to the wireless synchronization setting is nontrivial: classical Byzantine agreement assumes synchronous message delivery, whereas wireless channels introduce variable and unpredictable delays that blur the boundary between a slow-but-correct node and a faulty one. Recent approaches combine signed timestamps with statistical outlier rejection and multipath cross-validation to provide practical Byzantine resilience in TSCH and 5G mesh networks [114], [123], though formal guarantees under realistic wireless delay models remain an open research challenge.

4) Open Problems: The distributed synchronization paradigm raises several questions that are actively being studied. How should consensus weights be adapted in real time to reflect link quality, oscillator stability, or GNSS availability? Can distributed algorithms achieve sub-microsecond convergence in large-scale industrial meshes where slot-frame constraints limit message frequency? What is the minimum connectivity requirement to guarantee convergence in dynamic topologies where nodes enter and leave the network? And how can distributed schemes be made interoperable with existing PTP or TSCH GM hierarchies so that they serve as a resilient fallback rather than a replacement? Addressing these questions will require combining insights from distributed computing, control theory, and wireless communication system design. Key open problems cut across PCOs, consensus/diffusion methods, and Byzantine-resilient synchronization. A central direction could be the development of unified frameworks that integrate PCO-style pulse interactions with average or dynamic consensus mechanisms and Byzantine fault-tolerant layers, enabling end-to-end secure and accurate clock synchronization rather than treating these components in isolation [120], [124], [125], [126]. Another major challenge lies in achieving complexity-efficient Byzantine fault-tolerant and self-stabilizing pulse synchronization protocols with near-optimal stabilization time and communication overhead, while also resolving fundamental theoretical questions about computational hardness, in particular whether pulse synchronization is intrinsically as difficult as exact consensus or more closely aligned with approximate agreement [125].

C. Intelligent and Self-Healing Synchronization

AI and ML are emerging as key enablers of adaptive and predictive synchronization in dynamic wired–wireless environments. Traditional protocols such as PTP, gPTP, or TSCH rely on periodic message exchanges and static correction models that assume relatively stable operating conditions. In modern deployments, where temperature, voltage, interference, and mobility fluctuate, these methods struggle to maintain accuracy without excessive overhead. AI and ML approaches provide a data-driven alternative, allowing systems to learn temporal patterns, predict timing deviations, and autonomously adapt synchronization policies in real time.

Predictive Drift and Oscillator Compensation: ML-based predictive models, such as recurrent neural networks, long short-term memory (LSTM) networks, or

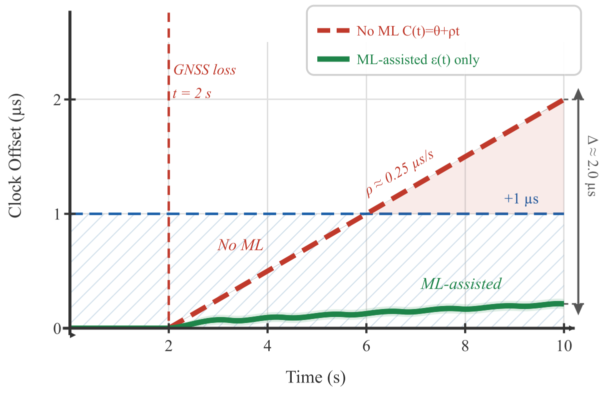

Fig. 12. ***Clock offset evolution during holdover following GNSS signal loss at $t = 2$ s. Without ML compensation, the offset grows linearly at rate $\rho \approx 0.25$ µs/s (affine clock model $C(t) = \theta + \rho t$), breaching the ± 1-µs tolerance bound at $t \approx 6$ s and reaching $\Delta \approx 2.0$ µs by $t = 10$ s. With ML-assisted holdover, the predicted drift correction suppresses ρ, confining the residual to the stochastic term $\varepsilon(t)$ well within the tolerance bound throughout. Hatched band: ± 1-µs tolerance region.***

Kalman-enhanced regressors, forecast oscillator drift based on historical offset data and environmental telemetry, as depicted in Fig. 12. These models enable proactive clock adjustment before significant offset accumulates. An ML-assisted holdover model can maintain ± 1-μs precision for several seconds without requiring new synchronization messages, whereas classical approaches may need frequent updates every 100 ms, substantially reducing network traffic and energy consumption for battery-constrained nodes. Current schemes mainly treat drift as a slowly varying scalar; coupling between drift, random phase noise, and network-induced delays remains largely unmodeled [127]. Future work must develop lightweight online predictors that separate deterministic drift from stochastic jitter, adapt across heterogeneous oscillators, and come with stability/safety guarantees when embedded in clock loops [128].

Learning-Based Jitter Control and Scheduling: Network-induced variability, from contention, interference, or congestion, is a major source of jitter in hybrid TSN and 5G systems. ML models can continuously monitor queue lengths, SINR variation, or retransmission statistics, and predict transient disruptions before they affect deterministic scheduling. Controllers can then preemptively reconfigure TSN GCLs (IEEE 802.1Qbv) or reallocate 5G resource blocks to critical slices, minimizing timing violations. Such predictive orchestration, illustrated in Fig. 13, has demonstrated significant jitter reduction in experimental and simulation studies [110], [111], enabling sub-10-μs consistency even in heavily loaded wireless environments.

Federated and Edge Learning for Scalability: Large-scale industrial and IoT systems cannot rely on centralized synchronization models due to privacy, latency, and heterogeneity constraints. Federated learning frameworks

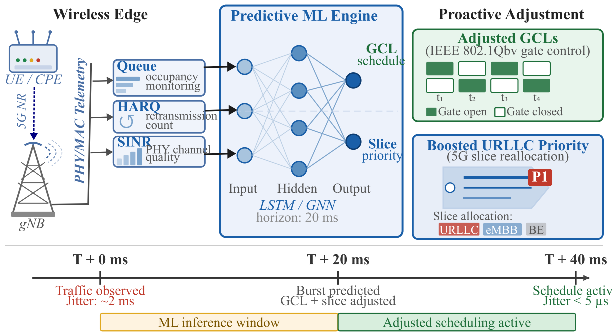

Fig. 13. ***ML-assisted jitter control in a hybrid 5G–TSN network. PHY/MAC telemetry, queue occupancy, HARQ retransmission count, and SINR are fed into an LSTM/GNN prediction engine that forecasts congestion bursts 20 ms ahead. The inferred schedule adjustments are applied proactively: TSN GCLs (IEEE 802.1Qbv) are realigned, and the URLLC slice is granted elevated priority, reducing jitter from $\sim$2 ms to below 5 μs within one scheduling cycle.***

address this by enabling distributed adaptation without exposing raw data: each node trains a local drift or jitter prediction model and periodically shares aggregated parameters with peers or a coordinating server, enabling system-wide refinement while maintaining autonomy. Open questions include how to encode time-sync. loss functions (drift, jitter, and deadline misses) within federated learning objectives, and robust aggregation under adversarial (Byzantine) or low-quality clients.

RL for Adaptive Control: RL has been explored to dynamically tune synchronization policies, such as adjusting update intervals, selecting reliable reference nodes, or managing fallback modes under GNSS loss [4], [111]. RL agents optimize long-term stability by balancing precision, energy, and communication cost. Graph neural networks (GNNs) extend this concept to multinode synchronization, enabling distributed optimization and consensus in mesh or multihop networks.

Future research will need to address the explainability, safety, and standardization of ML-based timing control. Hybrid models combining physics-based drift estimation with AI prediction appear especially promising, offering interpretable yet adaptive solutions [4], [67]. Ultimately, ML-driven synchronization moves the field beyond static correction loops toward **self-optimizing, predictive, and resilient timing fabrics** capable of sustaining deterministic microsecond-level precision in large, heterogeneous, and dynamic systems. Open problems include: certifiable jitter bounds for learned policies, sample-efficient training under rare but critical overloads, and combining model-based timing analysis with learning-based fine-tuning.

D. Secure and Resilient Synchronization

As synchronization becomes integral to industrial automation, vehicular coordination, and smart infrastructures, it also emerges as a prime attack vector. Timing information underpins control, safety, and monitoring processes, making even small perturbations potentially catastrophic. Attackers can exploit the inherent openness of time-distribution mechanisms to inject subtle, persistent errors that compromise entire systems. Thus, securing synchronization is not merely about confidentiality but about maintaining operational integrity and trust in time itself. This section is deliberately separated from the intelligent synchronization treatment above (see Section VIII-C) because the threat model, adversarial assumptions, and design constraints differ fundamentally from those of performance optimization under benign conditions.

1) Wireless-Specific Attack Surface: While wired synchronization infrastructures face well-understood threats, delay injection at intermediate switches, PTP master impersonation, the wireless medium introduces an additional and qualitatively different attack surfaces. An adversary with a radio transmitter can mount *over-the-air timestamp injection* attacks without any physical access to the network: by transmitting forged IEEE 802.11 or BLE timing frames, an attacker can shift the reference clock of an entire TSCH cluster or Wi-Fi-TSN cell within the reach of a commodity SDR. The broadcast nature of wireless channels means that a single malicious transmission can affect every node within range simultaneously, unlike wired delay injection which requires per-link access. Furthermore, *delay asymmetry exploitation* is substantially easier over wireless links, where uplink and downlink paths traverse different interference environments: an adversary who can selectively jam one direction of a two-way timestamp exchange induces a systematic offset bias that conventional protocols cannot distinguish from genuine propagation delay. Finally, the computational and energy constraints of IoT and TSCH nodes limit the overhead that can be spent on cryptographic verification, creating a tension between security strength and real-time timing performance that does not exist in the same form for wired PTP GMs.

2) Protocol-Level Vulnerabilities: Modern synchronization infrastructures, particularly GNSS, IEEE 1588 PTP, and gPTP, are vulnerable to several classes of attacks.

1) *GNSS Spoofing and Jamming:* By transmitting counterfeit satellite signals, adversaries can mislead receivers into aligning with false UTC references or deny synchronization entirely. The resulting time shifts may trigger incorrect phase-angle calculations in PMUs, mistimed actuation in robots, or cascading coordination failures.
2) *PTP/gPTP Delay Manipulation:* Attackers controlling intermediate switches or routers can delay, reorder, or duplicate sync. messages, inducing clock offsets that accumulate silently. Such "delay attacks" evade basic integrity checks and are particularly insidious in multihop wired–wireless topologies.
3) *Replay and Drift Injection:* Captured synchronization messages can be replayed at precise intervals, or timing parameters can be tampered with to induce

gradual drift, violating control loop stability without raising immediate alarms.

These attacks undermine determinism, safety, and trust, particularly in systems where even microsecond-level deviations can destabilize operations. Yet traditional security tools, digital signatures, MACs, or TLS encapsulation, introduce unacceptable jitter and delay. The central research challenge is to design synchronization protection that preserves both *precision* and *performance*.

3) Emerging Defenses: Emerging "Secure PTP" extensions (e.g., IEEE 1588 Annex K, IEC 62439-3) advocate authenticated synchronization messages using elliptic-curve signatures or HMACs executed within sub-millisecond bounds through hardware-assisted cryptography, TEEs, or cryptographic co-processors. Data-driven anomaly detection complements cryptographic approaches by identifying unusual offset corrections inconsistent with oscillator drift models, repeated message patterns suggestive of replay attacks, or divergent delay asymmetries beyond physical plausibility. Resilience also depends on multisource cross-checking: nodes continuously compare GNSS, PTP, peer-to-peer sync., and local oscillator estimates to compute trust scores, with outlier rejection or weighted fusion ensuring continuity even when a single source is spoofed or degraded. Future architectures will likely integrate **hardware roots of trust** for timing: secure oscillators, tamper-resistant timestamp units, and cryptographically signed time attestations ensure that local clocks cannot be manipulated without detection.

Promising directions include **layered trust architectures** combining authenticated PTP with anomaly detection and multisource voting; **adaptive trust scoring** dynamically adjusting synchronization weights based on observed stability and environmental conditions; and **collaborative detection** sharing timing health metrics across TSN, 5G, and GNSS domains to identify correlated attacks or faults. Achieving secure and reliable synchronization with minimal latency remains an interdisciplinary challenge spanning cryptography, control theory, and distributed systems.

E. Cross-Domain Convergence and Integration

A defining challenge for the next decade is the convergence of previously isolated time domains, wired TSN, cellular TSC, Wi-Fi TSN, TSCH, and satellite timing, into a coherent, end-to-end synchronization fabric. Where current deployments maintain discrete timing islands, emerging 5G/6G standards and deterministic Ethernet profiles increasingly seek unified timing semantics across technologies.

1) TSN, 5G/6G, and Cross-Domain Interoperability: As deterministic communication expands beyond wired Ethernet, integrating TSN with next-generation wireless systems such as 5G and 6G has become one of the most pressing synchronization challenges. While IEEE 802.1AS provides sub-microsecond precision in wired domains, maintaining this level of alignment across heterogeneous segments, Ethernet boundary clocks, gNBs, and Wi-Fi APs remains a fundamental open problem.

Traditional layered network architectures were never designed to manage ultratight timing constraints. The PHY may detect oscillator drift or SNR degradation long before upper layers notice a timing fault; the MAC layer may retransmit packets without awareness of global cycle deadlines; and TSN shapers assume fixed link delays, unaware of transient latency spikes from fading or interference. These blind spots collectively threaten the coherence of cross-domain synchronization.

Achieving microsecond-scale determinism across hybrid wired–wireless systems demands a **cross-layer synchronization model** where timing control and state information are shared across the stack in real time. In 5G TSC deployments, this integration is beginning to take shape: radio-level metrics such as SINR evolution, HARQ latency, and channel coherence time are being exposed to TSN translation entities, the NW-TT and DS-TT, aligned with 3GPP Release 16 and 17 specifications [59], [129]. In parallel, IEEE and 3GPP working groups are developing interworking models that map gPTP states onto 5G timing references to ensure unified temporal semantics. Cross-layer coordination further extends to PHY-aware time shaping: TSN-enabled switches and Wi-Fi 7 APs can expose instantaneous link-delay statistics and feed them directly into adaptive GCLs [108], [109].

2) Joint Communication, Sensing, and Timing: The emergence of ISAC in 5G and 6G networks creates a new class of synchronization requirements that are qualitatively different from those of communication-only systems. Synchronization accuracy sufficient for communication, achievable with gPTP at the 100-ns–1-μs level and distributed via TSC and TSN mechanisms, is insufficient for sensing applications such as localization, where ranging precision of centimeters requires picosecond- to nanosecond-class clock alignment [130], [131].

We use the term **JCST** to denote the design paradigm in which synchronization, communication, and sensing are co-optimized over the same physical resources and hardware, rather than being handled by separate subsystems. The concept generalizes the *Timing-as-a-Service* idea embedded in 5G's time-exposure mechanism (3GPP TS 23.501 [41]) by requiring synchronization precision to scale with sensing requirements rather than communication requirements alone. JCST is distinct from ISAC in that it explicitly includes timing as a first-class co-optimization objective alongside data throughput and sensing resolution; ISAC literature has largely treated synchronization as a background requirement rather than a design variable.

Several works have begun to explore the operational implications of this co-design. Integrated localization and synchronization, where ranging procedures simultaneously estimate internode distances and clock offsets, is a well-established thread in the literature [132], [133]. More recent contributions propose synchronization from sensing pilot signals [134], demonstrate concepts of integrated communication, localization, sensing, and synchronization [135], and analyze the impact of synchronization accuracy on radar ranging ambiguities when radar sensing is embedded into communication networks [136]. Kalman-filter-based joint estimation of clock asynchronism and sensing parameters in OFDM ISAC waveforms [137] offers a concrete algorithmic framework for JCST at the receiver level.

The JCST paradigm raises several open research problems. First, what synchronization precision is required as a function of sensing application (localization, gesture recognition, and synthetic aperture radar (SAR) imaging), and how does this propagate back into waveform design and pilot overhead? Second, can picosecond-level synchronization be achieved wirelessly using ISAC signals without dedicated timing hardware? Third, how should JCST resources, pilot symbols, bandwidth, and correction message rate be jointly allocated to balance communication, sensing, and timing performance within a unified framework? Fourth, how does synchronization uncertainty propagate into sensing accuracy, and can joint Cramér–Rao bounds be derived that capture this interdependency? Addressing these questions is essential for realizing the full potential of ISAC in next-generation networks, and positions JCST as a key unresolved challenge at the intersection of wireless systems, estimation theory, and TSN.

3) Global Time Distribution Across Wired, Wireless, and Space Domains: The proliferation of GNSS, emerging LEO constellations, and optical backbones is catalyzing *planetary-scale time fabrics* in which UTC anchors in space blend with terrestrial PTP/gPTP and TSN domains to deliver seamless timing from substations to spacecraft. Realizing this vision demands that sub-μs alignment survive translation across heterogeneous media, fiber TSN backbones, 5G TSC and Wi-Fi 6/7 access, and satellite links, despite divergent jitter profiles, propagation dynamics, and failure modes.

A unified architecture benefits from three pillars: *cross-domain orchestration* (SDN-inspired) that maintains per-domain timing models and mediates between GMs, gNBs, and APs; *adaptive correction* that fuses telemetry from satellite propagation, Wi-Fi contention, and 5G link jitter budgets into real-time adjustments of TSN GCLs and radio schedules; and *standardized timing APIs and metadata* across IEEE 802.1, IETF DetNet, and 3GPP that expose clock confidence, drift rates, and re-discipline events for proactive rather than reactive coordination. Together, these requirements lead to several representative open research questions for global time fabrics, summarized in Table 8.

Table 8 Representative Research Questions for Global Time Fabrics

Open Question	Why It Matters
Real-time correction for LEO Doppler and beam handovers	Enables satellite-assisted UTC with sub-μs stability.
Cross-domain orchestration sustaining μs alignment across TSN/Wi-Fi/5G	Required by roaming industrial, XR, and vehicular workloads.
Resilient trust: detect, isolate, and recover from spoofed or unstable time	Prevents cascaded timing faults in critical infrastructure.
Lightweight standardised inter-domain timing APIs and metadata	Makes proactive, predictive alignment feasible at scale.

Looking forward, **Timing-as-a-Service** and JCST suggest networks where time is a first-class, distributed service co-optimized with data and sensing. Edge time-anchors in switches, gNBs, and LEO payloads with disciplined oscillators and radio ranging can deliver nanosecond-class references across vast areas; TSN and 5G translators consume PHY/MAC state to adapt schedules; and federated, authenticated time flows enable graceful degradation and fast resynchronization. Building this planetary-scale time fabric is a grand systems challenge, demanding tight co-design of protocols, control, and security, but it is the bedrock for the next decade of industrial autonomy, wide-area V2X, and immersive XR at global scale.

F. Visionary Outlook

The convergence of TSN, 5G/6G, and AI marks a paradigm shift in how time is perceived, distributed, and utilized across interconnected systems. Rather than serving as an auxiliary protocol parameter, time is becoming an active, programmable resource, a shared fabric uniting communication, sensing, and control across all network layers.

Future synchronization frameworks will merge ultra-stable optical and quantum clocks with AI-driven orchestration, federated multisource time fusion, and resilient trust architectures. Networks will not simply exchange timestamps; they will reason about, negotiate, and adapt their temporal states collaboratively. Emerging 6G concepts such as TaaS and JCST envision globally distributed spatio-temporal fabrics capable of delivering nanosecond-level precision and contextual reliability, forming the temporal backbone of cyber-physical intelligence.

Across the technologies surveyed in this work, several key insights crystallize:

Ultratight synchronization, within sub-microsecond or few-microsecond bounds, is indispensable for deterministic TSN scheduling, precise phase alignment in smart grids, and haptic-level consistency in XR and teleoperation [138], [139].

Hybrid timing architectures, integrating GNSS discipline, TSN/PTP domains, Wi-Fi 6/7 extensions, and

5G TSC links, enable flexibility and coverage but amplify challenges around jitter propagation, mobility-induced drift, and cross-domain calibration [8], [140].

Security and trust in time have become as critical as precision. Attacks such as GNSS spoofing, delay injection, and replay manipulation threaten not only synchronization accuracy but also system safety [123], [141].

Benchmarking discipline remains a weak link. Evaluations must extend beyond idealized simulations to include empirical traces from PMUs, V2X pilots, Wi-Fi/TSN testbeds, and realistic interference patterns.

Looking forward, synchronization systems are expected to evolve into **intelligent, adaptive, and resilient infrastructures**. Core developments will include 6G-era TSC capabilities with nanosecond-scale access scheduling and seamless integration with sensing [142]; edge intelligence and ML-enhanced control enabling predictive drift modeling and autonomous rediscipline mechanisms [110], [111]; and satellite-integrated time overlays combining GNSS and emerging LEO constellations to maintain high-assurance synchronization even in GPS-denied environments [143].

To accelerate this evolution, the community must prioritize open collaboration and reproducibility through shared tools and data. Comprehensive, multilayer benchmarks should evaluate performance under real-world stressors, wireless mobility, GNSS outages, asymmetric delay, interference bursts, and oscillator aging. Open-source synchronization stacks spanning TSN, Wi-Fi 6/7, 5G TSC, BLE, GNSS, and LEO overlays will enable experimentation with adaptive filters, ML-based schedulers, and self-healing correction logic. And unified simulation–replay testbeds combining OMNeT++, NS-3, and the 6TiSCH simulator with empirical trace replay will support end-to-end validation under both synthetic and measured noise profiles [79], [140].

Ultimately, the trajectory of research points toward globally consistent, fault-tolerant, and context-aware synchronization fabrics, systems that continuously adapt to network, environmental, and security dynamics. Time will no longer be a static reference but a dynamic, self-optimizing service, anchoring the reliability and intelligence of next-generation cyber-physical and autonomous systems.

IX. CONCLUSION

The central finding of this survey is not that wireless clock synchronization is hard, but that the field has fractured along technology boundaries in a way that obscures the underlying unity of the problem. Whether a protocol targets a TSCH mesh, a 5G TSC, a UWB ranging system, or a GNSS-disciplined wide-area sensor network, it is solving the same estimation problem under different noise regimes and resource constraints. The five-dimension taxonomy introduced in this work makes that unity explicit: it reveals that the dominant source of synchronization failure in heterogeneous deployments is not the precision of any individual protocol but the absence of a coherent framework for composing protocols across technology and administrative boundaries. Fragmented, technology-specific solutions will not close the gap between wired TSN determinism and wireless reality; the field needs architectural rethinking, not better implementations of the same design pattern.

The architectural direction we advocate is the replacement of rigid master–slave hierarchies with *federated, adaptive time fabrics*, where each node blends multiple timing references—GNSS, local PTP or gPTP domains, peer-to-peer exchanges, and oscillator-based prediction—weighted by runtime confidence metrics rather than static topology. Synchronization becomes a multisource estimation process with graceful degradation, shifting trust across references as conditions change. The building blocks already exist in IEEE 802.1AS, 3GPP Release 16/17 TSC, and the consensus algorithms surveyed in Section VIII-B; what is missing is the cross-layer orchestration that binds them. Standardization bodies and the research community should treat federated time management as a first-class network function—with its own APIs, health metrics, and fault-containment semantics—rather than an emergent property of running several protocols in parallel.

The deeper structural shift is the one that JCST represents. Classical synchronization treats time distribution as a service layer beneath communication and sensing; in 6G systems this ordering is no longer tenable, as sensing accuracy, timing precision, and channel state form a closed dependency loop that JCST collapses into a single co-design problem. Timing-as-a-Service provides the abstraction that makes this deployable at scale, through distributed time anchors, radio-based ranging integrated with clock estimation, and nanosecond-scale timing delivered as a network service, but requires commitment to co-design from the outset, not retrofitting timing onto communication architectures built without it.

Two further gaps demand urgent attention. Security remains largely an afterthought in synchronization protocol design: GNSS spoofing, PTP delay injection, and replay attacks are practical, demonstrated threats, yet are addressed by optional extensions rather than foundational constraints. Authenticated synchronization with hardware-rooted trust and multisource cross-validation should be a baseline requirement. Benchmarking presents an equally critical gap: the evaluation culture relies heavily on idealized simulation, rarely exposing protocols to the combination of radio frequency interference, GNSS outage, asymmetric delay, and oscillator aging that characterizes real deployments. Shared, multilayer benchmarks with open traces, standardized stress scenarios, and verifiable baselines are not obstacles to future work—they are the preconditions for it. ■

REFERENCES

[1] B. Sundararaman, U. Buy, and A. D. Kshemkalyani, "Clock synchronization for wireless sensor networks: A survey," *Ad hoc Netw.*, vol. 3, no. 3, pp. 281–323, 2005.

[2] Q. Dong and W. Dargie, "A survey on mobility and mobility-aware MAC protocols in wireless sensor networks," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 1, pp. 88–100, 1st Quart., 2013.

[3] H. Ji, S. Park, J. Yeo, Y. Kim, J. Lee, and B. Shim, "Ultra-reliable and low-latency communications in 5G downlink: Physical layer aspects," *IEEE Wireless Commun.*, vol. 25, no. 3, pp. 124–130, Jun. 2018.

[4] X. Wang, P. Jia, X. Shen, and H. V. Poor, "Intelligent and low overhead network synchronization for large-scale industrial IoT systems in the 6G era," *IEEE Netw.*, vol. 37, no. 3, pp. 76–84, May 2023.

[5] F. Dang, X.-K. Sun, K.-B. Liu, Y.-F. Xu, and Y.-H. Liu, "A survey on clock synchronization in the industrial internet," *J. Comput. Sci. Technol.*, vol. 38, no. 1, pp. 146–165, 2023.

[6] K. Balakrishnan et al., "Clock synchronization in industrial Internet of Things and potential works in precision time protocol: Review, challenges and future directions," *Int. J. Cognit. Comput. Eng.*, vol. 4, pp. 205–219, 2023.

[7] A. Morato, E. Ferrari, C. Zunino, M. Cheminod, S. Vitturi, and F. Tramarin, "IEEE 1588 (PTP) over mmWave 5G NR: Preliminary assessment of the synchronization accuracy for TSN applications," in *Proc. IEEE 29th Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2024, pp. 1–4.

[8] J. Sasiain, D. Franco, A. Atutxa, J. Astorga, and E. Jacob, "Toward the integration and convergence between 5G and TSN technologies and architectures for industrial communications: A survey," *IEEE Commun. Surveys Tuts.*, vol. 27, no. 1, pp. 259–321, 4th Quart., 2024.

[9] D. Mills, J. Martin, J. Burbank, and W. Kasch, *Network Time Protocol Version 4: Protocol and Algorithms Specification*, document RFC 5905, RFC Editor, 2010.

[10] J. C. Eidson, *Measurement, Control, and Communication Using IEEE 1588*. Cham, Switzerland: Springer, 2006.

[11] S. Rodrigues and J. Lv, "Synchronization in time-sensitive networking: An introduction to IEEE std 802.1AS," *IEEE Commun. Standards Mag.*, vol. 6, no. 4, pp. 14–20, Dec. 2022.

[12] A. Najafi and M. Wei, "Graham: Synchronizing clocks by leveraging local clock properties," in *Proc. USENIX Symp. Networked Syst. Des. Implement. (NSDI)*, 2022, pp. 453–466.

[13] T. Schmid, Z. Charbiwala, J. Friedman, Y. H. Cho, and M. B. Srivastava, "Exploiting manufacturing variations for compensating environment-induced clock drift in time synchronization," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 36, no. 1, pp. 97–108, Jun. 2008.

[14] N. Kerö, A. Puhm, T. Kernen, and A. Mroczkowski, "Performance and reliability aspects of clock synchronization techniques for industrial automation," *Proc. IEEE*, vol. 107, no. 6, pp. 1011–1026, Jun. 2019.

[15] H. Wang, R. Lu, Z. Peng, and M. Li, "Clock synchronization with partial timestamp information for wireless sensor networks," *Signal Process.*, vol. 209, Aug. 2023, Art. no. 109036.

[16] A. Guruswamy, R. S. Blum, S. Kishore, and M. Bordogna, "Minimax optimum estimators for phase synchronization in IEEE 1588," *IEEE Trans. Commun.*, vol. 63, no. 9, pp. 3350–3362, Sep. 2015.

[17] A. K. Karthik and R. S. Blum, "Optimum full information, unlimited complexity, invariant, and minimax clock skew and offset estimators for IEEE 1588," *IEEE Trans. Commun.*, vol. 67, no. 5, pp. 3624–3637, May 2019.

[18] A. K. Karthik and R. S. Blum, "Robust clock skew and offset estimation for IEEE 1588 in the presence of unexpected deterministic path delay asymmetries," *IEEE Trans. Commun.*, vol. 68, no. 8, pp. 5102–5119, Aug. 2020.

[19] H. Wang, X. Guo, X. Liu, W. Ma, and M. Li, "Timestamp-free synchronization under delays with arbitrary distribution in wireless sensor networks," *IEEE Commun. Lett.*, vol. 27, no. 3, pp. 986–990, Mar. 2023.

[20] M. Wyss, M. Frei, J. Kwon, and A. Perrig, "Low-cost and robust global time synchronization," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2025, pp. 1891–1909.

[21] R. Ghasemi, P. Fenske, T. Koegel, M. Hehn, I. Ullmann, and M. Vossiek, "Ultrahigh-performance radio frequency system-on-chip implementation of a Kalman filter-based high-precision time and frequency synchronization for networked integrated sensing and communication systems," *IEEE Open J. Instrum. Meas.*, vol. 4, pp. 1–15, 2025.

[22] J. Song et al., "Design and implementation of a new wireless time synchronization method over IEEE 802.11," *IEEE Trans. Ind. Informat.*, vol. 21, no. 6, pp. 5011–5021, Jun. 2025.

[23] Y. Lei, J. Li, Z. Liu, R. Joshi, and Y. Xia, "Nanosecond precision time synchronization for optical data center networks," 2024, *arXiv:2410.17012*.

[24] H. Baniabdelghany, "Reliable, energy-efficient and temporally predictable time-triggered hybrid TSN systems combing wireless and wirebound networks," Ph.D. dissertation, Univ. Siegen, Siegen, Germany, 2023, doi: 10.25819/ubsi/10270.

[25] Z. Wang, T. Yong, X. Song, and D. Sun, "Maximum likelihood estimation of relative clock skew for distributed time synchronization in industrial wireless sensor networks," *IEEE Sensors J.*, vol. 23, no. 24, pp. 31359–31367, Nov. 2023.

[26] Y. Liu et al., "Time synchronization techniques in the modern smart grid: A comprehensive survey," *Energies*, vol. 18, no. 5, p. 1163, 2025.

[27] R. N. Gore, E. Lisova, J. Åkerberg, and M. Björkman, "CoSiNeT: A lightweight clock synchronization algorithm for industrial IoT," in *Proc. 4th IEEE Int. Conf. Ind. Cyber-Phys. Syst. (ICPS)*, May 2021, pp. 92–97.

[28] M. K. Hasan, M. M. Ahmed, A. H. A. Hashim, A. Razzaque, S. Islam, and B. Pandey, "A novel artificial intelligence based timing synchronization scheme for smart grid applications," *Wireless Pers. Commun.*, vol. 114, no. 2, pp. 1067–1084, 2020.

[29] *IEEE Standard for Local and Metropolitan Area Networks—Timing and Synchronization for Time-Sensitive Applications*, Standard 802.1AS-2020, 2020.

[30] E. Kaplan and C. Hegarty, *Understanding GPS/GNSS: Principles and Applications*, 3rd ed., Norwood, MA, USA: Artech House, 2017.

[31] C. Lenzen, P. Sommer, and R. Wattenhofer, "Optimal clock synchronization in networks," in *Proc. 7th ACM Conf. Embedded Networked Sensor Syst.*, Nov. 2009, pp. 225–238.

[32] F. Ferrari, M. Zimmerling, L. Thiele, and O. Saukh, "Efficient network flooding and time synchronization with glossy," in *Proc. 10th ACM/IEEE Int. Conf. Inf. Process. Sensor Netw.*, Apr. 2011, pp. 73–84.

[33] S. Ganeriwal, R. Kumar, and M. B. Srivastava, "Timing-sync protocol for sensor networks," in *Proc. 1st Int. Conf. Embedded Netw. Sensor Syst.*, 2003, pp. 138–149.

[34] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," *OSDI*, vol. 36, no. 5, pp. 147–163, 2002.

[35] L. Schenato and F. Fiorentin, "Average TimeSynch: A consensus-based protocol for clock synchronization in wireless sensor networks," *Automatica*, vol. 47, no. 9, pp. 1878–1886, Sep. 2011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0005109811003116

[36] P. Sommer and R. Wattenhofer, "Gradient clock synchronization in wireless sensor networks," in *Proc. Int. Conf. Inf. Process. Sensor Netw.*, Apr. 2009, pp. 37–48.

[37] J. He, P. Cheng, L. Shi, J. Chen, and Y. Sun, "Time synchronization in WSNs: A maximum-value-based consensus approach," *IEEE Trans. Autom. Control*, vol. 59, no. 3, pp. 660–675, Mar. 2014.

[38] F. Shi, X. Tuo, L. Ran, Z. Ren, and S. X. Yang, "Fast convergence time synchronization in wireless sensor networks based on average consensus," *IEEE Trans. Ind. Informat.*, vol. 16, no. 2, pp. 1120–1129, Feb. 2020.

[39] H. Wang, Y. Zou, X. Liu, and M. Li, "A fast convergence scheme for distributed consensus time synchronization using multihop virtual links in industrial wireless sensor networks," *IEEE Sensors J.*, vol. 24, no. 12, pp. 20009–20017, Jun. 2024.

[40] J. Wu, L. Zhang, Y. Bai, and Y. Sun, "Cluster-based consensus time synchronization for wireless sensor networks," *IEEE Sensors J.*, vol. 15, no. 3, pp. 1404–1413, Mar. 2015.

[41] M. Pope, *System Architecture for the 5G System (5GS)*, Standard TS 23.501, version 18.6.0, 3GPP, 2023. [Online]. Available: https://www.3gpp.org/DynaReport/23501.htm

[42] I. Val, O. Seijo, R. Torrego, and A. Astarloa, "IEEE 802.1AS clock synchronization performance evaluation of an integrated wired–wireless TSN architecture," *IEEE Trans. Ind. Informat.*, vol. 18, no. 5, pp. 2986–2999, May 2022.

[43] Z. Satka, M. Ashjaei, H. Fotouhi, M. Daneshtalab, M. Sjödin, and S. Mubeen, "A comprehensive systematic review of integration of time sensitive networking and 5G communication," *J. Syst. Archit.*, vol. 138, May 2023, Art. no. 102852. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1383762123000310

[44] *IEEE Standard for Low-Rate Wireless Networks. Amendment 1: Enhanced Ultra Wideband (UWB) Physical Layers (PHYs) and Associated Ranging Techniques*, Standard 802.15. 4z, 2020.

[45] D. Dujovne, T. Watteyne, X. Vilajosana, and P. Thubert, "6TiSCH: Deterministic IP-enabled industrial Internet (of Things)," *IEEE Commun. Mag.*, vol. 52, no. 12, pp. 36–41, Dec. 2014.

[46] LoRa Alliance. (2021). *LoRaWAN Regional Parameters*. [Online]. Available: https://resources.lora-alliance.org/technical-specifications/rp002-1-0-4-regional-parameters

[47] Bluetooth SIG. (2021). *Bluetooth Core Specification v5.3*. [Online]. Available: https://www.bluetooth.com/specifications/bluetooth-core-specification/

[48] Bluetooth SIG. (2017). *Bluetooth Mesh Profile Specification v1.0*. [Online]. Available:https://www.bluetooth.com/specifications/mesh-specifications/

[49] L. Alkama and L. Bouallouche-Medjkoune, "IEEE 802.15.4 historical revolution versions: A survey," *Computing*, vol. 103, no. 1, pp. 99–131, Jan. 2021.

[50] A. R. Urke, Ø. Kure, and K. Øvsthus, "A survey of 802.15. 4 TSCH schedulers for a standardized industrial Internet of Things," *Sensors*, vol. 22, no. 1, p. 15, 2021.

[51] *IEEE Standard for Local and Metropolitan Area Networks—Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC Sublayer*, Standard 802.15.4e-2012, Amendment to IEEE 802.15.4-2011 to Support TSCH and Other MAC Enhancements, Apr. 2012. [Online]. Available: https://standards.ieee.org/ieee/802.15.4e/5051/

[52] M. R. Palattella, P. Thubert, X. Vilajosana, T. Watteyne, Q. Wang, and T. Engel, "6tisch wireless industrial networks: Determinism meets ipv6," in *Internet Things: Challenges Opportunities*. Cham, Switzerland: Springer, 2014, pp. 111–141.

[53] N. Sornin, O. Luis, T. Eirich, T. Kramp, and O. Hersent. (2015). *LoRaWAN Specification*. LoRa Alliance. [Online]. Available: https://resources.lora-alliance.org/technical-

specifications/lorawan-specification-v1-0

[54] U. Raza, P. Kulkarni, and M. Sooriyabandara, “Low power wide area networks: An overview,” *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 855–873, 2nd Quart., 2017.

[55] *IEEE Standard for Information Technology-Telecommunication and Information Exchange Between Systems-Local and Metropolitan Area Networks-Specific Requirements—Part11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment1: Radio Resource Measurement of Wireless LANs*, Standard IEEE 802.11bh-2024, 2020. [Online]. Available: https://www.ieee802.org/misc-docs/GlobeCom2009/IEEE802d11Kraemer.pdf

[56] E. Khorov, A. Kiryanov, A. Lyakhov, and G. Bianchi, “A tutorial on IEEE 802.11ax high efficiency WLANs,” *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 197–216, 1st Quart., 2019.

[57] M. F. R. Al-Okby, S. Junginger, T. Roddelkopf, and K. Thurow, “UWB-based real-time indoor positioning systems: A comprehensive review,” *Appl. Sci.*, vol. 14, no. 23, p. 11005, 2024.

[58] M. Ahmed et al., “Resilient time synchronization in 6G networks: A hot standby solution,” *IEEE Commun. Standards Mag.*, vol. 9, no. 1, pp. 52–58, Mar. 2025.

[59] A. Sultan and M. Pope. *Service Requirements for Ultra-Reliable Low-Latency Communications (URLLC)*, Standard TS 22.104, version 18.2.0, 3GPP, 2022. [Online]. Available: https://www.3gpp.org/DynaReport/22104.htm

[60] M. K. Atiq, R. Muzaffar, Ó. Seijo, I. Val, and H.-P. Bernhard, “When IEEE 802.11 and 5G meet time-sensitive networking,” *IEEE Open J. Ind. Electron. Soc.*, vol. 3, pp. 14–36, 2022.

[61] 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 Commun. Mag.*, vol. 57, no. 12, pp. 45–51, Dec. 2019.

[62] Ó. Seijo, X. Iturbe, and I. Val, “SHARP: Implementation of a hybrid wired-wireless TSN network to enable flexible smart factories,” in *Proc. 17th IEEE Int. Conf. Factory Commun. Syst. (WFCS)*, Jun. 2021, pp. 95–98.

[63] D. Gao et al., “Time synchronization based on cross-technology communication for IoT networks,” *IEEE Internet Things J.*, vol. 10, no. 22, pp. 19753–19764, Jun. 2023.

[64] R. Hofmann, D. Grubmair, C. A. Boano, and K. Römer, “X-sync: Cross-technology clock synchronization among off-the-shelf wireless IoT devices,” in *Proc. IEEE 46th Conf. Local Comput. Netw. (LCN)*, Oct. 2021, pp. 115–122.

[65] L. L. Bello and W. Steiner, “A perspective on IEEE time-sensitive networking for industrial communication and automation systems,” *Proc. IEEE*, vol. 107, no. 6, pp. 1094–1120, 2019.

[66] M. Gutiérrez, W. Steiner, R. Dobrin, and S. Punnekkat, “Synchronization quality of IEEE 802.1AS in large-scale industrial automation networks,” in *Proc. IEEE Real-Time Embedded Technol. Appl. Symp. (RTAS)*, Apr. 2017, pp. 273–282.

[67] P. Jia, X. Wang, and X. Shen, “Digital-twin-enabled intelligent distributed clock synchronization in industrial IoT systems,” *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4548–4559, Mar. 2021.

[68] T. Qiu, Y. Zhang, D. Qiao, X. Zhang, M. L. Wymore, and A. K. Sangaiah, “A robust time synchronization scheme for industrial Internet of Things,” *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3570–3580, Aug. 2018.

[69] C. Jay, M. Glencross, and R. Hubbold, “Modeling the effects of delayed haptic and visual feedback in a collaborative virtual environment,” *ACM Trans. Comput.-Human Interact. (TOCHI)*, vol. 14, no. 2, p. 8, 2007.

[70] S. K. Jagatheesaperumal, K. Ahmad, A. Al-Fuqaha, and J. Qadir, “Advancing education through extended reality and Internet of Everything enabled metaverses: Applications, challenges, and open issues,” *IEEE Trans. Learn. Technol.*, vol. 17, pp. 1120–1139, 2024.

[71] I. Llatser, T. Michalke, M. Dolgov, F. Wildschutte, and H. Fuchs, “Cooperative automated driving use cases for 5G V2X communication,” in *Proc. IEEE 2nd 5G World Forum (5GWF)*, Sep. 2019, pp. 120–125.

[72] K. F. Hasan, C. Wang, Y. Feng, and Y.-C. Tian, “Time synchronization in vehicular ad-hoc networks: A survey on theory and practice,” *Veh. Commun.*, vol. 14, pp. 39–51, Oct. 2018.

[73] K. F. Hasan, Y. Feng, and Y.-C. Tian, “Precise GNSS time synchronization with experimental validation in vehicular networks,” *IEEE Trans. Netw. Service Manage.*, vol. 20, no. 3, pp. 3289–3301, Sep. 2023.

[74] N. A. S. I. (NASPI). (2020). *NASPI Technical Report: Synchrophasor Technology and Applications*. [Online]. Available: https://www.naspi.org/

[75] L. Chasserat, N. Accettura, and P. Berthou, “LoRaSync: Energy efficient synchronization for scalable LoRaWAN,” *Trans. Emerg. Telecommun. Technol.*, vol. 35, no. 2, p. 4940, Feb. 2024.

[76] A. Varga and R. Hornig, “An overview of the OMNeT++ simulation environment,” in *Proc. 1st Int. ICST Conf. Simulation Tools Techn. Commun. Netw. Syst.*, 2008, pp. 1–10.

[77] G. Nardini, D. Sabella, G. Stea, P. Thakkar, and A. Virdis, “Simu5G—An OMNeT++ library for end-to-end performance evaluation of 5G networks,” *IEEE Access*, vol. 8, pp. 181176–181191, 2020.

[78] G. F. Riley and T. R. Henderson, “The NS-3 network simulator,” in *Modeling and Tools for Network Simulation*. Berlin, Germany: Springer, 2010, pp. 15–34.

[79] E. Municio et al., “Simulating 6TiSCH networks,” *Trans. Emerg. Telecommun. Technol.*, vol. 30, no. 3, p. 3494, Mar. 2019.

[80] C. Adjih et al., “FIT IoT-LAB: A large scale open experimental IoT testbed,” in *Proc. IEEE World Forum on Internet of Things (WF-IoT)*, Milan, Italy, Dec. 2015, pp. 459–464.

[81] D. Raychaudhuri et al., “Challenge: COSMOS: A city-scale programmable testbed for experimentation with advanced wireless,” in *Proc. 26th Annu. Int. Conf. Mobile Comput. Netw.* New York, NY, USA: Association for Computing Machinery, Apr. 2020, pp. 1–13, doi: 10.1145/3372224.3380891.

[82] J. Breen et al., “Powder: Platform for open wireless data-driven experimental research,” *Comput. Netw.*, vol. 197, Oct. 2021, Art. no. 108281.

[83] L. Bonati et al., “Colosseum: Large-scale wireless experimentation through hardware-in-the-loop network emulation,” in *Proc. IEEE Int. Symp. Dyn. Spectr. Access Netw. (DySPAN)*, Dec. 2021, pp. 105–113.

[84] M. Vucinic, T. Chang, B. Škrbic, E. Kocan, M. Pejanovic-Djurišic, and T. Watteyne, “Key performance indicators of the reference 6TiSCH implementation in Internet-of-Things scenarios,” *IEEE Access*, vol. 8, pp. 79147–79157, 2020, doi: 10.1109/ACCESS.2020.2990278.

[85] S. Sudhakaran, K. Montgomery, M. Kashef, D. Cavalcanti, and R. Candell, “Wireless time sensitive networking impact on an industrial collaborative robotic workcell,” *IEEE Trans. Ind. Informat.*, vol. 18, no. 10, pp. 7351–7360, Oct. 2022.

[86] 5G-ACIA. (2024). *5G-ACIA Industrial 5G & TSN Interoperability Lab Overview*. Accessed: Jul. 18, 2025. [Online]. Available: https://www.5g-acia.org/testbeds

[87] Y.-G. Yue and P. He, “A comprehensive survey on the reliability of mobile wireless sensor networks: Taxonomy, challenges, and future directions,” *Inf. Fusion*, vol. 44, pp. 188–204, Nov. 2018.

[88] 5G-MOBIX Consortium. (2022). *5G-MOBIX V2X Field Trial Logs and Timing Dataset*. Accessed: Jul. 20, 2025. [Online]. Available: https://5g-mobix.com/resources/deliverables

[89] B. Fong, L. Situ, and A. C. Fong, “Smart technologies and vehicle-to-X (V2X) infrastructures for smart mobility cities,” in *Smart Cities: Foundations and Principles*, H. Song, R. Srinivasan, T. Sookoor, and S. Jeschke, Eds., Hoboken, NJ, USA: Wiley, 2017, pp. 181–208.

[90] A. J. Wilson et al., “The grid event signature library: An open-access repository of power system measurement signatures,” *IEEE Access*, vol. 12, pp. 76207–76218, 2024.

[91] S. Biswas et al., “An open-access repository of synchrophasor data quality examples: Curation and example applications,” *IEEE Access*, vol. 13, pp. 66445–66457, 2025.

[92] S. Scanzio et al., “Wireless sensor networks and TSCH: A compromise between reliability, power consumption, and latency,” *IEEE Access*, vol. 8, pp. 167042–167058, 2020.

[93] A. Povalac, J. Kral, H. Arthaber, O. Kolar, and M. Novak, “Exploring LoRaWAN traffic: In-depth analysis of IoT network communications,” *Sensors*, vol. 23, no. 17, p. 7333, Aug. 2023.

[94] M. Seliem, A. Zahran, and D. Pesch, “TSN-based industrial network performance analysis,” in *Proc. IEEE 8th World Forum Internet Things (WF-IoT)*, Oct. 2022, pp. 1–7.

[95] M. Ahmed et al., “Analysis of time synchronization for 6G-TSN networks with hot standby,” in *Proc. 34th Int. Conf. Comput. Commun. Netw. (ICCCN)*, Aug. 2025, pp. 1–6.

[96] M. Ahmed, L. Haug, R. Muzaffar, D. Hamidovic, A. Hadziaganovic, and H.-P. Bernhard, “First analysis of time synchronization for TSN networks with hot standby GM,” in *Proc. IEEE 21st Int. Conf. Factory Commun. Syst. (WFCS)*, Jun. 2025, pp. 1–4.

[97] H. Li, G. Shou, Y. Liu, and Y. Hu, “Precise timing over beyond 5G networks for intelligent transport in the smart city,” in *Proc. IEEE 35th Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Sep. 2024, pp. 1–6.

[98] D. K. Sah, M. Vahabi, and H. Fotouhi, “A comprehensive review on 5G IIoT test-beds,” *IEEE Trans. Consum. Electron.*, vol. 71, no. 2, pp. 4139–4163, May 2025.

[99] D. Saucez, S. Gallenmuller, R. Knopp, N. Makris, and S. Fdida, “Blueprint-based reproducible research with the SLICES research infrastructure,” in *Proc. IEEE INFOCOM Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Vancouver, BC, Canada, May 2024, pp. 1–2.

[100] M. Vucinic, B. Škrbic, E. Kocan, M. Pejanovic-Djurišic, and T. Watteyne, “OpenBenchmark: Repeatable and reproducible Internet of Things experimentation on testbeds,” in *Proc. IEEE INFOCOM Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Paris, France, Apr. 2019, pp. 289–294.

[101] Fraunhofer IIS. (2024). *Industrial 5G & TSN Integration Test Facility: Timing, Reliability, and Mobility Results*. Accessed: Jul. 18, 2025. [Online]. Available: https://www.iis.fraunhofer.de/industrial5g

[102] Ericsson. (2023). *Enabling Time-Critical Applications Over 5G With Rate Adaptation*. Accessed: Jul. 20, 2025. [Online]. Available: https://www.ericsson.com/en/reports-and-papers/white-papers/enabling-time-critical-applications-over-5g-with-rate-adaptation

[103] K. F. Hasan, Y. Feng, and Y.-C. Tian, “GNSS time synchronization in vehicular ad-hoc networks: Benefits and feasibility,” *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 12, pp. 3915–3924, Dec. 2018.

[104] M. Aslam et al., “Hardware efficient clock synchronization across Wi-Fi and Ethernet-based network using PTP,” *IEEE Trans. Ind. Informat.*, vol. 18, no. 6, pp. 3808–3819, Jun. 2022.

[105] D. Pallier, V. Le Cam, and S. Pillement, “Energy-efficient GPS synchronization for wireless nodes,” *IEEE Sensors J.*, vol. 21, no. 4, pp. 5221–5229, Feb. 2021.

[106] D. Hamidovic et al., “6G schedule and application traffic alignment for efficient radio resource utilization,” in *Proc. IEEE 100th Veh. Technol. Conf. (VTC-Fall)*, Oct. 2024, pp. 1–7.

[107] C. Fan, Z. Yao, and M. Lu, "Wireless subnanosecond time synchronization for moving platforms," *IEEE Trans. Instrum. Meas.*, vol. 74, pp. 1–10, 2025.

[108] A. Morato, E. Ferrari, S. Vitturi, and F. Tramarin, "A TSN-based technique for latency measurement in real-time wireless communication networks," in *Proc. IEEE Int. Symp. Meas. Netw. (M&N)*, Jul. 2024, pp. 1–6.

[109] H. Chahed and A. Kassler, "TSN network scheduling—Challenges and approaches," *Network*, vol. 3, no. 4, pp. 585–624, Dec. 2023.

[110] Y. Peng, A. Jolfaei, Q. Hua, W.-L. Shang, and K. Yu, "Real-time transmission optimization for edge computing in industrial cyber-physical systems," *IEEE Trans. Ind. Informat.*, vol. 18, no. 12, pp. 9292–9301, Dec. 2022.

[111] D. Santos et al., "Machine learning-based holdover for IEEE 1588 PTP packet fronthauling," in *Proc. 19th Int. Symp. Wireless Commun. Syst. (ISWCS)*, Jul. 2024, pp. 1–6.

[112] R. E. Mirollo and S. H. Strogatz, "Synchronization of pulse-coupled biological oscillators," *SIAM J. Appl. Math.*, vol. 50, no. 6, pp. 1645–1662, Dec. 1990.

[113] O. Simeone and U. Spagnolini, "Distributed time synchronization in wireless sensor networks with coupled discrete-time oscillators," *EURASIP J. Wireless Commun. Netw.*, vol. 2007, no. 1, p. 57054, Dec. 2007.

[114] Y. Kato, L. Yuan, and H. Ishii, "Resilient synchronization of pulse-coupled oscillators with time continuity," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 11197–11202, Jul. 2023.

[115] J. F. Schmidt, U. Schilcher, A. Vogell, and C. Bettstetter, "Using randomization in self-organized synchronization for wireless networks," *ACM Trans. Auto. Adapt. Syst.*, vol. 18, no. 3, pp. 1–20, Sep. 2023.

[116] J. Yan and H. Ishii, "Secure synchronization of heterogeneous pulse-coupled oscillators," in *Proc. Eur. Control Conf. (ECC)*, Jun. 2025, pp. 647–652.

[117] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1520–1533, Sep. 2004.

[118] J. Wu, K. Bai, and H. Wu, "Advancing convergence speed of distributed consensus time synchronization algorithms in unmanned aerial vehicle ad hoc networks," *Drones*, vol. 8, no. 7, p. 285, Jun. 2024.

[119] H. Wang, X. Zhu, X. Liu, Y. Zou, T. Tan, and M. Li, "Event-triggered time synchronization with varying skews for wireless sensor networks based on average consensus," *IEEE Trans. Netw. Sci. Eng.*, vol. 12, no. 4, pp. 2895–2906, Jul. 2025.

[120] S. S. Kia, B. Van Scoy, J. Cortes, R. A. Freeman, K. M. Lynch, and S. Martinez, "Tutorial on dynamic average consensus: The problem, its applications, and the algorithms," *IEEE Control Syst. Mag.*, vol. 39, no. 3, pp. 40–72, Jun. 2019.

[121] Y. Du, H. Tu, H. Yu, and S. Lukic, "Accurate consensus-based distributed averaging with variable time delay in support of distributed secondary control algorithms," *IEEE Trans. Smart Grid*, vol. 11, no. 4, pp. 2918–2928, Jul. 2020.

[122] H. Gao and Y. Wang, "Algorithm-level confidentiality for average consensus on time-varying directed graphs," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 2, pp. 918–931, Mar. 2022.

[123] A. Nasrullah and F. M. Anwar, "SoK: State of the time: On trustworthiness of digital clocks," 2025, *arXiv:2502.09837*.

[124] Z. Wang and Y. Wang, "Global synchronization of pulse-coupled oscillator networks under Byzantine attacks," *IEEE Trans. Signal Process.*, vol. 68, pp. 3158–3168, 2020.

[125] C. Lenzen and J. Rybicki, "Self-stabilising Byzantine clock synchronisation is almost as easy as consensus," *J. ACM*, vol. 66, no. 5, pp. 1–56, Oct. 2019.

[126] S. Koushkbaghi, M. Safi, A. M. Amani, M. Jalili, and X. Yu, "Byzantine-resilient second-order consensus in networked systems," *IEEE Trans. Cybern.*, vol. 54, no. 9, pp. 4915–4927, Sep. 2024.

[127] Y. Yang et al., "Clock drift and time error compensation with neural network for time synchronization in TSN," in *Proc. IEEE 7th Adv. Inf. Technol., Electron. Autom. Control Conf. (IAEAC)*, Mar. 2024, pp. 440–444.

[128] P. Liu and S. Duan, "An open-loop synchronization technique with simple structure for phase error compensation and frequency estimation," *IEEE Trans. Ind. Electron.*, vol. 67, no. 10, pp. 8936–8940, Oct. 2020.

[129] P. Muñoz, P. Rodríguez-Martín, J. Caleya-Sánchez, J. Prados-Garzón, Ó. Adamuz-Hinojosa, and A. P. Ameigeiras, "Joint scheduling of IEEE 802.1AS gPTP and industrial data traffic in TSN-6G networks," *IEEE Access*, vol. 13, pp. 99842–99862, 2025.

[130] F. Liu et al., "Integrated sensing and communications: Toward dual-functional wireless networks for 6G and beyond," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 6, pp. 1728–1767, Jun. 2022.

[131] S. Lu et al., "Integrated sensing and communications: Recent advances and ten open challenges," *IEEE Internet Things J.*, vol. 11, no. 11, pp. 19094–19120, Jun. 2024.

[132] B. Etzlinger, F. Meyer, F. Hlawatsch, A. Springer, and H. Wymeersch, "Cooperative simultaneous localization and synchronization in mobile agent networks," *IEEE Trans. Signal Process.*, vol. 65, no. 14, pp. 3587–3602, Jul. 2017.

[133] J. Wei, X. Wang, and B. Zhou, "Joint clock synchronization and localization for 6G integrated communication and sensing," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, May 2023, pp. 200–206.

[134] H. Wymeersch et al., "Cross-layer integrated sensing and communication: A joint industrial and academic perspective," *IEEE Open J. Commun. Soc.*, pp. 6966–7015, 2025.

[135] D. Volgushev and G. Fokin, "Integrated communication, localization, sensing and synchronization in 6G cognitive wireless networks," in *Proc. Syst. Signal Synchronization, Generating Process. Telecommun. (SYNCHROINFO)*, Jul. 2024, pp. 1–6.

[136] J. A. Zhang, K. Wu, X. Huang, Y. J. Guo, D. Zhang, and R. W. Heath, "Integration of radar sensing into communications with asynchronous transceivers," *IEEE Commun. Mag.*, vol. 60, no. 11, pp. 106–112, Nov. 2022.

[137] X. Chen, Z. Feng, J. A. Zhang, X. Yuan, and P. Zhang, "Kalman filter-based sensing in communication systems with clock asynchronism," *IEEE Trans. Commun.*, vol. 72, no. 1, pp. 403–417, Jan. 2024.

[138] N. Force, "Time synchronization in the electric power system," NASPI, Richland, WA, USA, Tech. Rep. NASPI-2017-TR-001, 2017.

[139] T. Taleb, Z. Nadir, H. Flinck, and J. Song, "Extremely interactive and low-latency services in 5G and beyond mobile systems," *IEEE Commun. Standards Mag.*, vol. 5, no. 2, pp. 114–119, Jun. 2021.

[140] A. Eloudrhiri Hassani, A. Sahel, A. Badri, and I. El Mourabit, "Multiple channel access techniques in industrial IoT: Evaluation case of time-slotted channel hopping," *Wireless Netw.*, vol. 30, no. 5, pp. 3119–3133, Jul. 2024.

[141] W. Alghamdi and M. Schukat, "Cyber attacks on precision time protocol networks—A case study," *Electronics*, vol. 9, no. 9, p. 1398, Aug. 2020.

[142] M. Giordani, M. Polese, M. Mezzavilla, S. Rangan, and M. Zorzi, "Toward 6G networks: Use cases and technologies," *IEEE Commun. Mag.*, vol. 58, no. 3, pp. 55–61, Mar. 2020.

[143] F. S. Prol et al., "Position, navigation, and timing (PNT) through low Earth orbit (LEO) satellites: A survey on current status, challenges, and opportunities," *IEEE Access*, vol. 10, pp. 83971–84002, 2022.

ABOUT THE AUTHORS

Mohamed Seliem (Member, IEEE) received the B.Sc. degree in electronics and communication engineering from Cairo University, Giza, Egypt, in 2011, the M.Sc. degree in electronics and electrical communications engineering from Cairo University, in 2016, the M.Eng. degree in computer engineering from the University of Louisiana at Lafayette, Lafayette, LA, USA, in 2019, and the Ph.D. degree in computer science from University College Cork, Cork, Ireland, in 2024.

He is currently a Research Fellow with University College Cork, affiliated with the NASC Research Group and the CONNECT Center. His research focuses on time-sensitive networking (TSN), deterministic communications, and converged wired–wireless architectures for industrial automation and smart factory applications. He has developed open-source simulation frameworks, predictive redundancy mechanisms, and machine-learning-based fault anticipation models for industrial IoT systems. His work spans network calculus, 5G/TSN integration, Wi-Fi 6/7, fault-tolerant networking, edge computing, and digital twins for Industry 4.0.

Utz Roedig (Member, IEEE) received the Dr.-Ing. degree in computer science from Darmstadt University of Technology, Darmstadt, Germany, in 2002.

He joined the School of Computer Science and Information Technology (CSIT), University College Cork (UCC), Cork, Ireland, in January 2019. Before moving to Cork, he was a Professor with Lancaster University, Lancaster, U.K., where he led the Academic Centre of Excellence in Cyber Security Research (ACE-CSR). Prior to his work at Lancaster University, he held research positions at UCC and Darmstadt University of Technology. His research interests are computer networks and security with a focus on the Internet of Things (IoT). His work looks at the IoT communication mechanisms and the software used to construct the IoT systems with a particular focus on cybersecurity. He has published over 160 peer-reviewed articles in this field, and his research collaborations with industry partners have resulted in several patents.

Mahin Ahmed (Member, IEEE) received the M.Sc. degree in mobile communications from Sejong University, Seoul, South Korea, in 2014, and the Dr.Techn. (Ph.D.) degree in information technology from the University of Klagenfurt, Klagenfurt, Austria, in 2021.

From 2015 to 2019, she was a Research and Teaching Staff Member with the University of Klagenfurt. Her Ph.D. thesis focused on interference dynamics in wireless networks. She is currently a Senior Scientist with Silicon Austria Labs (SAL), Linz, Austria. Her research interests include wireless networks, industrial wireless networks, wireless time-sensitive networking, and 5G/6G systems.

Dr. Ahmed is an Active Member of IEEE 3388 and 802.1 WGs.

Raheeb Muzaffar (Member, IEEE) received the joint Ph.D. degree from the University of Klagenfurt, Klagenfurt, Austria, and the Queen Mary University of London, London, U.K., in 2016.

He is currently a Senior Scientist with Silicon Austria Labs (SAL), Linz, Austria. His current work focuses on the research and development of 5G/6G communication for the industrial IoT, wireless time-sensitive networking, and communication functional safety. He has authored several research articles published in prestigious journals and conferences, two book chapters, and has been involved in several national and international research projects.

Damir Hamidovic (Member, IEEE) received the B.Sc. and M.Sc. degrees in telecommunication engineering from the University of Tuzla, Tuzla, Bosnia and Herzegovina, in 2013 and 2015, respectively, and the Ph.D. degree from the Department of Communication Engineering and RF Systems, Johannes Kepler University (JKU) Linz, Linz, Austria, in 2022.

Previously, he worked at JKU as a Researcher and at Intel, Villach, Austria, and Apple, Linz, as a System Engineer in the field of digitally assisted RF transceivers for future mobile communications. He has been a Scientist with Silicon Austria Labs GmbH, Linz, Austria, since 2022. His expertise and focus of interests include industrial wireless communications with a focus on testbed deployments, measurements, analysis of wireless technologies, situational awareness, and digital twins. He has been involved in several national Austrian and international European research projects.

Armin Hadziaganovic (Member, IEEE) received the M.Sc. degree in telecommunication engineering from the Faculty of Electrical Engineering, University of Tuzla, Tuzla, Bosnia and Herzegovina, in 2015.

The research for his M.Sc. degree was also conducted at Buskerud and Vestfold University College, Drammen, Norway. From 2015 to 2020, he worked on various projects in the healthcare IT industry. In 2020, he enrolled in Ph.D. studies at Johannes Kepler University in Linz (JKU) to perform scientific work in the area of industrial wireless communications with a focus on functional safety, safety-critical communication, situational awareness, and digital twins. He is currently a Scientist with Silicon Austria Labs GmbH (SAL), Linz, Austria.

Cormac J. Sreenan (Senior Member, IEEE) received the Ph.D. degree in computer science from the University of Cambridge, Cambridge, U.K., in 1992.

He is currently a Full Professor and the Former Head of computer science at University College Cork (UCC), Cork, Ireland. Before joining UCC in 1999, he worked at AT&T Labs–Research, Florham Park, NJ, USA, and Bell Labs, Murray Hill, NJ, USA. He currently directs the NASC Research Centre, a team of about 50 people investigating networks, systems, and security. His projects are funded by industry, EU, Enterprise Ireland, and Taighde Éireann—Research Ireland. He is also a Principal Investigator and a Co-PI of the national Irish CONNECT Research Centre on Future Networks and Communications.

Dr. Sreenan was elected as a Fellow of British Computer Society in 2005 and Irish Academy of Engineering in 2023. He has served on the Editorial Boards for IEEE Transactions on Mobile Computing, *ACM Transactions on Sensor Networks*, and other journals.

Dirk Pesch (Senior Member, IEEE) received the Dipl.-Ing. degree from RWTH Aachen University, Aachen, Germany, in 1993, and the Ph.D. degree from the University of Strathclyde, Glasgow, U.K., in 1999.

He is currently a Professor with the School of Computer Science and Information Technology, University College Cork, Cork, Ireland, and the Former Head of the Nimbus Research Centre, Munster Technological University. With over 30 years of research and development experience, he has authored more than 250 publications. He is also a Principal Investigator in the Research Ireland-funded CONNECT Centre and the Director of the Research Ireland Centre for Research Training in Advanced Networks for Sustainable Societies. His research focuses on the architecture, design, and performance of low-power, dense, and vehicular wireless networks for IoT and cyber–physical systems in smart environments. He has been involved in various EU projects and regularly contributes to international conference organization.