On the Impact of gPTP Time Synchronization Attacks on Time-Sensitive Networks

Cornelia Samp, Nurefşan Sertbaş Bülbül, Mathias Fischer
Department of Informatics, University of Hamburg, Germany
{cornelia.bruelhart, nurefsan.sertbas, mathias.fischer}@uni-hamburg.de

***Abstract*—Time synchronization is a fundamental building block for Time-Sensitive Networking (TSN), as it aligns transmission schedules across network nodes. The IEEE 802.1AS specifies the generalized Precision Time Protocol (gPTP), which provides sub-microsecond accuracy but lacks integrity protection, making it vulnerable to message delay, spoofing, and timestamp manipulation. These vulnerabilities pose challenges for TSN scheduling mechanisms, particularly time-triggered approaches like IEEE 802.1Qbv Time-Aware Shaper (TAS), but also rate-based shapers like IEEE 802.1Qav Credit-Based Shaper (CBS). This paper presents a systematic, end-to-end analysis of how gPTP time synchronization attacks degrade TSN performance. We study three realistic attacker types: a malicious Grandmaster distributing falsified time, cumulative delay injection attacks by compromised boundary clocks, and asymmetric delay attacks that selectively manipulate synchronization messages. We implement these attacks in OMNeT++ and evaluate their impact on CBS in 300 network instances (random, mesh, and scale-free topologies) with mixed-criticality traffic. Results indicate that per-message time offsets up to 5 µs, can accumulate to over 100 µs of clock divergence within one second, causing more than 80% of CBS-scheduled flows to miss deadlines even with a 10% tolerance. Our findings demonstrate the severe vulnerability of TSN scheduling to gPTP attacks across diverse network structures.**

I. Introduction

Mission-critical applications such as industrial automation, medical systems, and in-vehicle networks rely on packet networks to coordinate distributed control loops, safety functions, and automation processes with tight latency and jitter bounds. The Time-Sensitive Networking (TSN) standards extend Ethernet with deterministic forwarding and traffic isolation to provide these guarantees. Accurate time synchronization is crucial as it enables time-aware forwarding, scheduling, and coordinated actuation across multiple hops [1] [2].

TSN relies on IEEE 802.1AS generalized Precision Time Protocol (gPTP) for sub-microsecond accuracy, enabling shapers such as IEEE 802.1Qav Credit-Based Shaper (CBS) and IEEE 802.1Qbv Time-Aware Shaper (TAS) to ensure flows meet strict end-to-end deadlines and required Quality of Service (QoS) levels. As networks increase in size and diversity, maintaining precise synchronization becomes challenging. Clock drift, asymmetric delays, and heterogeneous hardware can cause timing errors that accumulate across multiple hops, undermining the guarantees on which scheduling mechanisms rely [3]. While standard Ethernet allows multiple traffic classes to share the same physical medium, TSN extends Ethernet with mechanisms that provide deterministic isolation and bounded latency guarantees. This improves network utilization but increases scheduling complexity, as multiple classes compete for resources, making precise time synchronization essential to prevent interference and preserve latency guarantees.

ISBN 978-3-903176-82-9 © 2026 IFIP

The original gPTP standard lacks built-in security features, leaving synchronization messages vulnerable to replay or tampering. Attackers can thus introduce timing inconsistencies, e.g., by spoofing gPTP messages. Although the 802.1ASdm amendment proposes countermeasures to address these vulnerabilities, e.g., a hot-standby Grandmaster (GM), it does not specifically address targeted attacks [4]. Consequently, gPTP-based networks remain susceptible to timing manipulations that can induce jitter and cause data flows to violate strict QoS constraints.

Existing work predominantly demonstrates how to desynchronize gPTP nodes but does not provide a quantitative assessment of its consequences, in particular, its impact on end-to-end latency, deadline guarantees and concrete TSN schedulers [5] [2] [6]. This research gap is particularly concerning for mission-critical environments, where such attacks may have severe implications. The main contribution of this paper is a systematic investigation of how attacks on gPTP time synchronization affect TSN performance. Specifically, our contributions are as follows:

- We present three realistic attack scenarios on gPTP time synchronization: (i) a malicious GM distributing corrupted time, (ii) a cumulative delay attack by compromised boundary clocks that distort residence time, and (iii) an asymmetric delay attack that selectively delays Follow_Up messages.
- We implement these attacks in OMNeT++ and evaluate them on different network topologies (random, mesh, and scale-free). We also quantify how longer paths and high-degree nodes amplify the propagation and severity of timing disruptions.
- We evaluate the end-to-end impact of gPTP synchronization attacks on mixed-flow networks scheduled with CBS, demonstrating how per-message offsets (5 µs) can accumulate to 116.7 µs within one second, causing over 80% of flows to miss deadlines even with a 10% latency tolerance margin.
- We perform a fine-grained sensitivity analysis of sub-microsecond switch clock offsets, measuring how even standard-compliant deviations can cause deadline viola-

tions for high-priority flows in CBS networks. Although our evaluation focuses on CBS, these results provide a conservative upper bound for TAS, where stricter time-based scheduling is expected to be even more sensitive to timing attacks.

The remainder of this paper is structured as follows: Section II introduces the time synchronization mechanism in TSN, outlines its vulnerabilities, and summarizes related work. Section III presents our attacker models and briefly describes their expected implications on TSN. Section IV summarizes our evaluation results. Section V concludes the paper and summarizes future work.

II. Background and Related Work

This section outlines the principles and mechanisms of time synchronization in TSN, identifies its inherent limitations, and discusses its security vulnerabilities.

A. Time Synchronization in TSN

Traditional protocols like Network Time Protocol (NTP) rely on software-based time stamping and best-effort delivery, providing only millisecond-level accuracy. Precision Time Protocol (PTP) offers significantly higher accuracy through hardware-assisted time stamping and a hierarchical clock structure, but still depends on a properly configured hierarchy, symmetric paths, and a PTP-aware network.

To enable time synchronization in TSN-compliant networks, the IEEE 802.1AS standard specifies gPTP, a profile of PTP [1]. The protocol establishes a hierarchical clock structure by electing a single GM using the Best Master Clock Algorithm (BMCA). The selected GM, typically the most accurate clock and often synchronized to an external reference (e.g., GPS), periodically transmits synchronization messages to propagate a common time across the network. The receiving nodes adjust their local clocks based on these gPTP messages and peer-to-peer delay measurements before forwarding the information downstream.

The gPTP synchronization process, as shown in Fig. 1, involves several types of messages. *Announce* messages are periodically broadcast by candidate GMs and include clock properties that help in selecting the most suitable GM using the BMCA. *Pdelay_Req* and *Pdelay_Resp* messages are used to measure link delays between peers. Once a GM is elected, it periodically transmits *Sync* and *Follow_Up* with precise timestamps to disseminate the current time. To achieve sub-microsecond synchronization accuracy, gPTP relies on hardware timestamping to bypass software delays by recording timestamps close to the physical layer. This significantly reduces jitter and uncertainty compared with software-based approaches. The tight coupling between gPTP and the TSN scheduling stack makes the global time base a critical dependency and, consequently, a high-impact attack surface.

B. Attacks and Vulnerabilities in gPTP

Even well-configured networks are subject to synchronization inaccuracies, e.g., due to hardware limitations, clock drift,

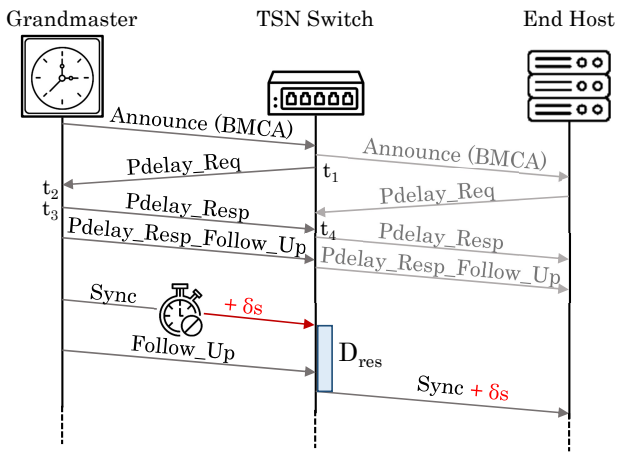

Fig. 1: gPTP message exchange with adversarial delay δ injected to a *Sync* message.

and network-induced effects such as jitter [7]. Beyond these inherent challenges, gPTP is also susceptible to targeted attacks that directly manipulate the time distribution mechanism.

Attacks can target the GM to undermine the synchronization tree. In spoofing attacks, an adversary impersonates the GM by manipulating *Announce* messages, e.g., to win the BMCA election. Once accepted as GM, the attacker can disseminate incorrect time, affecting all downstream nodes. In addition to GM spoofing, a compromised switch can tamper with legitimate *Announce* messages, degrading trust in the actual GM and promoting an attacker-controlled clock instead [8]. Recent studies explore how adversaries might mimic the identity and timing behavior of a GM, potentially leading to widespread desynchronization and degradation of time-sensitive applications [9].

Delay attacks are another prominent example of path manipulation. In these attacks, compromised TSN switches introduce artificial latency into *Sync* and/or *Follow_Up* messages, misleading downstream devices. In [10], the authors investigate how adversaries can disrupt servo convergence by replaying outdated *Sync* messages or injecting forged messages with incorrect offsets.

Furthermore, in message flooding attacks, adversaries inject large volumes of *Sync* or *Delay_Req* messages, consuming network and processing resources and blocking critical traffic [5], [11]. While these studies demonstrate protocol-level disruptions, they do not quantify how such timing manipulations affect end-to-end latency and deadline guarantees in TSN.

C. Existing Detection and Hardening Approaches

The IEEE 802.1ASdm amendment proposes enhancements to gPTP, to improve synchronization robustness and accuracy in TSN [4]. Key features include a hot-standby GM functionality to facilitate the transition if the primary GM becomes unavailable. Additionally, the amendment introduces mechanisms to refine message formats, thereby improving the accuracy of neighbors' clock-speed calculations. These improvements primarily address natural clock drift, hardware

drift, and failures. However, they do not protect against intentional synchronization attacks, leaving time-sensitive networks vulnerable to targeted disruptions.

Beyond standardization efforts, several detection-based approaches have been proposed to identify synchronization anomalies. Threshold-based methods monitor clock behavior and raise alerts when deviations exceed predefined limits. For instance, Buscemi et al. [12] and Lisova et al. [13] apply thresholds to clock frequency and offset deviations in conjunction with an Intrusion Detection System (IDS). Similarly, Mizrahi [14] focuses on round-trip delay measurements, triggering alarms when observed delays surpass acceptable thresholds.

Furthermore, path redundancy can enhance resilience by providing multiple synchronization paths, potentially limiting the impact of faulty nodes. Luo et al. [15] demonstrate that monitoring redundant links can help detect inconsistencies caused by delay attacks. This enables early detection and fault tolerance but increases resource consumption and management overhead. Also, an attacker may evade or undermine redundant synchronization paths. Standard authentication methods, such as Type-Length-Value (TLV) extensions, can prevent unauthorized spoofing, but cannot stop a compromised legitimate GM from distributing incorrect time.

A core limitation of current gPTP implementations is the lack of cryptographic protection for message authenticity and integrity, mainly due to concerns about added latency and computational overhead in TSN environments [16]. While security extensions, such as message authentication codes (MACs), IPsec, and MACsec have been proposed and evaluated, they primarily protect against spoofing and tampering. However, these methods fail to address delay-based or internal attacks in which messages remain syntactically valid and authenticated while their timing is manipulated [10].

Threshold-based anomaly detection and redundant path monitoring can help identify synchronization inconsistencies, but subtle timing manipulations often remain below detection thresholds. Consequently, hybrid mitigation strategies that combine authentication, redundancy, and structural network awareness are needed to enhance resilience without compromising latency or determinism. In summary, these mitigation techniques offer valuable improvements but do not provide a complete defense against time synchronization attacks.

III. Time Synchronization Attacks on gPTP

This section covers attacker models on gPTP and their effects on CBS scheduling, with practical defense measures.

A. Attacker Model

We define three attacker models targeting different synchronization roles within TSN as illustrated in Fig. 2. Each attacker is assumed to have persistent access to at least one network device and the ability to manipulate gPTP messages at the source or along the forwarding path. The injected timing offsets, drawn from a bounded distribution, can be positive or negative, making detection via simple thresholds or drift monitoring difficult.

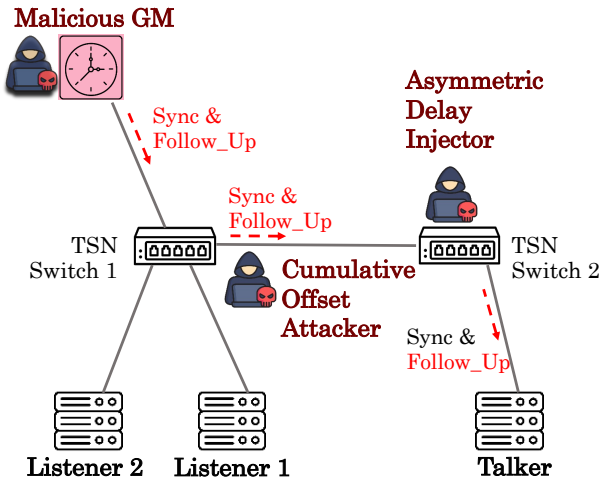

Fig. 2: Exemplary network with three attackers injecting offsets into gPTP messages to disrupt the network's time synchronization.

1) **Malicious Grandmaster:** An attacker takes control of the elected GM and generates falsified *Sync* and *Follow_Up* messages containing incorrect timestamps. Instead of providing a consistent and accurate reference clock, the GM provides varying incorrect time information in each interval, causing downstream time-aware systems to align with an unstable time base. This type of attack has the highest privileges and is the most severe as it undermines the foundations of the synchronization hierarchy and affects the entire domain.
2) **Cumulative Delay Attack:** A subset of malicious TSN switches is compromised to intercept and modify gPTP messages in transit. Both *Sync* and *Follow_Up* messages are delayed by injecting small varying timing offsets per message. This requires access to the forwarding path (e.g., via firmware compromise or control of the switch) and allows the attacker to degrade synchronization without replacing the GM.
3) **Asymmetric Delay Attack:** A subset of malicious TSN switches, selectively delays *Follow_Up* messages, causing asymmetric propagation delays. This attack is stealthier and results in location-dependent synchronization errors, with certain network parts drifting out of sync more than others. Unlike the first two attacks, the message content remains unchanged, making it more difficult to detect.

These attacks are realistic because they exploit inherent trust assumptions in TSN. For instance, boundary clocks can accumulate per-hop delays without requiring network-wide disruption, especially in industrial networks [2].

B. gPTP Attacks and TSN Scheduling

In the following, we discuss how clock drift induced by time synchronization attacks impacts key TSN mechanisms, namely

TAS, CQF, and CBS. Their commonality lies in being affected by timing deviations, even though their sensitivity differs.

a) IEEE 802.1Qbv Time-Aware Shaper (TAS): TAS-based scheduling is the most severely affected by time synchronization attacks. TAS organizes communication into repeating cycles, each divided into time slots assigned to specific traffic classes. Each slot controls the transmission eligibility of traffic classes via a Gate Control List (GCL) that references the global time base. TAS-configured devices open and close gates on transmission queues based on their local clock aligned to this time base. If an attacker causes a device's local clock to drift, gates may close prematurely or open too late, leading to dropped or delayed packets. Furthermore, incorrectly opened gates can also cause transmissions to collide with higher-priority traffic or violate isolation constraints.

b) IEEE 802.1Qch Cyclic Queueing and Forwarding (CQF): CQF is a time-aware forwarding mechanism that divides time into fixed-duration cycles. Each switch buffers incoming frames during one cycle and transmits them in the next. Unlike TAS, CQF does not rely on global time synchronization; instead, it relies on synchronization between adjacent switches to maintain aligned cycle boundaries. If synchronization between neighboring switches is disrupted, e.g., due to a time synchronization attack, cycle misalignment occurs: A receiving switch may close its ingress gate while the upstream switch continues to transmit. Consequently, CQF loses its latency guarantees, with increased packet loss and delays, particularly in multi-hop scenarios where misalignments accumulate.

c) IEEE 802.1Qav Credit-Based Shaper (CBS): The IEEE 802.1Qav Credit-Based Shaping (CBS) is a rate-based traffic-shaping mechanism whose performance is only indirectly affected by synchronization accuracy. In CBS, each traffic class maintains a credit counter that increases during idle periods and decreases while frames are transmitted. Transmission is permitted only when sufficient credit is available; otherwise, frames remain queued until credit is replenished. Unlike time-triggered TSN mechanisms such as TAS, CBS does not rely on strict global time synchronization to operate. Nevertheless, each time-aware node runs a clock servo that continuously adjusts its local clock frequency based on Sync/Follow_Up timestamps. Under attack, the servo converges to a frequency that compensates for the injected offset rather than the true offset, resulting in a persistent frequency error at each node. Since each boundary clock runs its own servo independently, this error is not corrected but carried forward and amplified across hops. Consequently, traffic bursts arrive earlier or later than expected at downstream switches, encountering credit balances that no longer match the anticipated arrival pattern. When the resulting credit mismatch exceeds the shaping threshold, frames are delayed by a full credit replenishment cycle or dropped, increasing queuing delay and degrading timeliness.

Among the TSN scheduling mechanisms, CBS is the least sensitive to synchronization inaccuracies. As a result, observing performance degradation in CBS under gPTP-based attacks, such as missed deadlines or increased queueing delays, provides a conservative lower bound on the impact of timing manipulation. This indicates that other time-sensitive mechanisms, like TAS, are likely to be even more sensitive to clock offsets. Therefore, we use CBS as a robust reference point to evaluate the indirect effects of timing manipulations on TSN performance.

C. Practical Challenges and Applicability of Existing Defenses

In this section, we review mitigation approaches under our attacker model, emphasizing key challenges and deployment trade-offs.

a) Malicious Grandmaster: A compromised legitimate GM that intentionally distributes incorrect time can severely impact the network. Standard authentication mechanisms, such as Type-Length-Value (TLV) extensions, prevent spoofing by unauthorized nodes but cannot prevent a malicious GM from issuing incorrect, yet validly authenticated updates. TLV authentication also increases message size, which may impact bandwidth usage and processing efficiency in resource-constrained networks [16]. Moreover, an internal adversary with access to shared group keys can still manipulate mutable fields, such as the *CorrectionField*. This limits the effectiveness of authentication unless combined with robust key management and additional security mechanisms [17].

Another mitigation approach is to disable dynamic GM election entirely and assign a trusted GM statically. Although this prevents unauthorized GM takeovers, it destroys redundancy and fault tolerance: In case of GM failure, synchronization is disrupted until manual reconfiguration takes place. Cryptographic protection of gPTP messages using MACsec, IPsec, or HMACs can help detect tampering or spoofing by attackers on the network path. Still, they do not protect against a compromised GM that generates validly authenticated messages [17]. Detecting malicious GMs, therefore, relies on plausibility checks, statistical monitoring, or cross-validation with neighboring clocks or independent time sources. These measures introduce complexity and may impact latency, a significant concern in a time-sensitive environment.

b) Cumulative and Asymmetric Delay Attacks: For delay-based attacks, potential countermeasures include measuring message delays along multiple independent paths (redundant path measurements) and threshold-based monitoring of delays and offsets to identify timing anomalies or compromised nodes. While these measures can detect coarse inconsistencies, our evaluation reveals that subtle manipulations remain sufficient to significantly degrade synchronization. Cryptographic protection is generally ineffective against delay-focused attacks, in which an adversary manipulates message delivery timing without altering contents [6]. As a result, messages remain syntactically valid and authenticated, while synchronization quality degrades. Path redundancy with cyclic asymmetry analysis, e.g., measuring gPTP message delays over multiple node- and link-disjoint paths and comparing their evolution across synchronization cycles, presents a promising

defense against delay-focused attacks. However, this approach requires sufficient topological diversity, which may not be available in all TSN deployments [18]. Complementary to path-based defenses, in-switch traffic characterization could offer a lightweight alternative for detecting timing irregularities [19].

Overall, existing mitigation techniques either assume uncompromised time sources, struggle with subtle delay manipulations, or rely on rich path redundancy. These approaches therefore provide valuable improvements but remain insufficient for defending against targeted and stealthy time synchronization attacks.

IV. Evaluation

This section investigates the impact of adversarial delay injection on time synchronization messages in CBS-scheduled TSN. We describe the evaluation setup and metrics, and examine how attack severity varies with topology and the proportion of compromised devices. Finally, we summarize key findings for CBS robustness in time-sensitive networks.

A. Experimental Setup

We use OMNeT++ 6.0.1 simulations with INET Framework 4.5, extending IEEE 802.1AS (gPTP) and IEEE 802.1Qav (CBS) modules with custom configurations and attack scenarios. All devices, including the GM, switches, and end hosts, are equipped with the *RandomDriftOscillator* to simulate realistic oscillation. Each oscillator is configured with a drift rate that is uniformly distributed between $\pm$ 100 µs and a drift rate change interval of 100 ms, capturing natural clock variability due to environmental fluctuations and oscillator aging. The gPTP synchronization interval is set to 1 ms, and an initial delay offset of 100 µs is introduced at the start of the simulation to reflect the initial calibration conditions and prevent all starting nodes from starting in a perfectly synchronized state [4]. The GM operates as a gPTP GM clock, end hosts act as ordinary clocks, and all TSN switches serve as gPTP boundary clocks, which synchronize to an upstream port and redistribute time downstream.

Propagation of synchronization errors can be closely related to topological properties; therefore, we examine three types of topologies to analyze the impact of synchronization attacks:

(i) *Mesh topologies*: Grid-based switch networks where end hosts are assigned to edge switches according to a Gaussian distribution.
(ii) *Random topologies*: Switches are connected using the Erdős–Rényi model. Each edge switch has at least one end host, with the remaining hosts distributed via the Gaussian allocation.
(iii) *Scale-free topologies*: Networks generated using the Barabási–Albert model to produce a power-law degree distribution with highly connected nodes. Each edge switch receives at least one host, while the remaining hosts are distributed randomly.

For each topology type, we generated 100 unique medium-scale networks, resulting in 300 simulated networks in total. Each network consists of 16 switches and 32 end hosts, representing typical real-world topologies [20]. Across all topologies, the network diameter ranged from 4 to 11 hops, and the average path lengths ranged from about 3.0 to 5.2 hops. We focus on medium-scale networks, as they provide a representative baseline, while considering that the effects observed would likely be even more pronounced in larger-scale deployments. All devices start each simulation in a synchronized state. For the gPTP synchronization experiments (Table I), Ethernet links operate at 1 Mbit/s, while for the CBS scheduling experiments (Fig. 3 and 4), links operate at 12 Mbit/s to accomodate mixed-criticality traffic. The traffic configuration results in an aggregate offered load of approximately 74 Mbit/s across all flows (48 Mbit/s best-effort, 19 Mbit/s video, 7 Mbit/s gPTP), over links with 100 Mbit/s capacity. To distinguish overload effects from synchronization-induced timing errors, we first evaluate a baseline scenario without clock offsets under the same load. Since the baseline exhibits no deadline violations, any additional deviations observed under injected clock offsets can be attributed to synchronization errors rather than bandwidth underprovisioning.

To maintain realism, during the attack phase *Sync* messages are perturbed with additional delay in the range of $[0, 499]$ ns, and *Follow_Up* messages with random offsets in the range of ± 5000 ns [4]. These values are selected to adhere to the timing tolerance outlined by IEEE 802.1AS and IEC/IEEE 61850-9-3, which describe a synchronization accuracy of ±1 µs and a maximum allowable jitter of 500 ns [21].

We generated realistic TSN traffic featuring multiple priority classes, varying packet lengths, and precise production intervals. Additionally, we implement the attacker models discussed in Section III-A using OMNeT++. By varying attacker placement and behavior, we systematically analyze the effects of different timing disruptions.

B. Evaluation Metrics

We evaluate the worst-case deviation between a device's local clock and the GM clock over the simulation period of $T = 1000$ ms. During this time, the GM transmits 1000 synchronization messages at 1 ms intervals, which are forwarded along the gPTP synchronization hierarchy by intermediate nodes. For each device $j \in \mathcal{D}$, the instantaneous offset at synchronization event t_i is defined as:

$$\delta_j(t_i) = \left|\text{Clock}_j(t_i) - \text{Clock}_{\text{GM}}(t_i)\right|. \tag{1}$$

Our first evaluation metric is the **maximum offset to GM**, which is defined as the largest clock difference observed between any device in the network and the GM during the simulation period. It captures the worst-case deviation from the reference clock among all switches and end hosts.

$$\delta_{\text{max}} = \max_{j \in \mathcal{D},\, t_i \in [0,T]} \delta_j(t_i) \tag{2}$$

where $\delta_j(t_i)$ is the absolute clock offset of device j at time t_i and $\mathcal{D}$ is the set of all devices (e.g., switches and end hosts).

To quantify how far devices drift apart relative to each other at the same time, we use **max inter-node offset**. It captures the

internal desynchronization within the network, independent of deviations from the reference clock, GM. We also measure the **mean clock drift** to quantify how fast a device's clock deviates from the GM clock over time, indicating the long-term stability of synchronization under attack. It is computed using linear regression over the synchronization interval, capturing the slope of the clock deviation over time.

$$\mu_j = \frac{\sum_i (t_i - \bar{t})(\delta_j(t_i) - \bar{\delta}_j)}{\sum_i (t_i - \bar{t})^2} \tag{3}$$

where t_i is the time of the i-th synchronization event, $\bar{t}$ is the mean of all t_i, and μ_j represents the average drift rate in seconds per second. Also, the **standard deviation of drift** reflects variations across devices, showing how unevenly the attack affects synchronization.

Furthermore, we use **jitter**, defined as the inter-quartile range of the clock offsets, to capture short-term fluctuations in synchronization. While drift reflects long-term stability, jitter highlights timing variability between updates. In TSN, jitter is a critical metric as it directly impacts the predictability and stability of time-sensitive traffic. Low jitter ensures that traffic adheres to deterministic delivery, maintaining consistent latency and synchronization, whereas high jitter introduces unpredictability, potentially causing delays and an overall degradation of the QoS through missed deadlines.

Finally, to assess CBS robustness, we use the **packet delay deviation from the mean**, measuring how each packet's delay differs from the flow's average and capturing delivery consistency across traffic loads and topologies.

C. Results

1) Impact of Time Synchronization Attacks: We evaluate the effects of different delay injection strategies on IEEE 802.1AS by analyzing clock deviation across various network topologies and attacker models. Summary results are presented in Table I, where rows list attack types, topologies, and compromised switches, and columns report key clock deviation metrics, including offsets, drift, and jitter.

a) Malicious Grandmaster: Overall, the malicious GM attack is highly effective, with the strongest impact observed in topologies that feature longer or more variable communication paths. Random topologies suffer the most, showing a maximum offset to the GM of 244.3 ms and the highest jitter (116.7 µs). This is primarily due to the longer and more diverse communication paths, which introduce greater inconsistency in synchronization timing and amplify the effects of delay manipulation. These conditions result in a maximum inter-node offset of 2.93 ms and a high drift variability in local clock drift. Consequently, some nodes exhibit a positive time offset relative to the GM (running ahead), while others exhibit a negative offset (lagging behind).

Scale-free topologies benefit from well-connected hub nodes that serve as relay points for gPTP messages. This structure reduces average path length and ensures faster, more consistent message propagation, resulting in a lower maximum offset and reduced inter-node skew. The slightly negative mean drift of -0.097 s, as calculated using Equation 3, suggests that some nodes overcorrect their clocks, likely due to faster receipt and adjustment from nearby hubs.

Mesh topologies exhibit the most stable behavior under this attack. Their equal node degree and uniform node distances support consistent message delays across the network, as reflected in the modest mean drift and low drift variation of 2.7 ms.

b) Cumulative Delay Attack: A fixed subset of adversary switches (10% to 30%) inject offsets into gPTP message timestamps, mimicking a malicious GM.

Random topologies exhibit the strongest degradation, with a maximum offset to the GM of 42.7 ms at 30% attackers and a max inter-node offset of 0.62 ms. These effects arise from the inherently longer paths and the random placement of compromised switches along the synchronization paths. The structural inconsistency results in uneven time corrections across the network, as clocks are adjusted to run slower, producing a persistent negative mean drift across nodes.

In scale-free topologies, delay propagation through hub nodes increases timing variability and an increased maximum offset to the GM. When the malicious switches are leaf switches, the observed timing deviations are more moderate compared to attacks involving hub switches, as reflected in scenarios with smaller percentages of adversaries.

Mesh networks maintain low offsets overall, but the direction of drift depends on attacker placement: redundant paths may either absorb or amplify local manipulations, making highly connected topologies sensitive to attacker position.

c) Asymmetric Delay Attack: Only the *Follow_Up* messages are manipulated, introducing asymmetry in delay across 10%, 20%, and 30% of compromised switches.

For random topologies, overall drift and offset remain modest, but the standard deviation of local clocks increases at 30% attacker ratio to 2.18 ms. This indicates that asymmetric delay effects are highly path-dependent. Some nodes lie on synchronization paths traversing multiple attacker-controlled points, whereas others are not affected at all, resulting in uneven synchronization across the network. Although some nodes remain close to the GM, a total collapse of network-wide synchronization is prevented, yet the unpredictability remains problematic.

Scale-free topologies exhibit slightly more stable behavior, with mean drift near zero and a low standard deviation, thanks to high-degree hub nodes that accelerate message dissemination and mitigate asymmetric distortions. Additionally, randomly chosen attacker nodes are typically leaf switches, whose limited role in synchronization paths reduces their overall impact compared to attacks targeting hub nodes.

Mesh networks, however, exhibit a consistent positive drift across all attacker ratios, which increases with the number of compromised nodes. High path redundancy can mitigate uncorrelated and random delay variations through averaging. However, this does not apply for asymmetric delay, since

TABLE I: Clock Synchronization Deviation under Different Attack Scenarios

Attack Type	Topology	Devices	Max Offset to GM [ms]	Mean Offset to GM [ms]	Max Inter-Node Offset [ms]	Mean Drift [s]	Std. Dev. Local Clock [ms]	Jitter at Switches [µs]
Malicious GM	Random	–	**244.317**	7.821	2.926	1.943	10.137	**116.702**
	Scale-Free		37.574	3.903	0.528	-0.097	3.477	58.536
	Mesh		27.715	3.010	0.105	0.843	2.709	48.315
Compromised Switch(es)	Random	10%	34.789	-0.005	0.566	-0.061	0.927	5.378
		20%	35.800	-0.006	0.607	-0.205	0.993	7.854
		30%	**42.673**	-0.012	0.622	-0.063	1.024	8.272
	Scale-Free	10%	1.536	-0.001	0.047	0.014	0.172	1.656
		20%	1.901	-0.002	0.054	0.073	0.236	3.764
		30%	2.257	-0.004	0.061	0.601	0.279	4.827
	Mesh	10%	1.043	0.002	0.017	-0.143	0.145	0.888
		20%	1.769	-0.001	0.021	0.417	0.197	3.233
		30%	2.012	-0.019	0.028	0.779	0.326	5.204
Asymmetric Delay	Random	10%	7.351	0.008	0.046	0.069	0.152	0.590
		20%	7.511	0.016	0.055	0.034	0.206	0.990
		30%	8.458	0.024	0.045	0.015	**2.182**	0.121
	Scale-Free	10%	0.208	0.004	0.005	-0.019	0.018	0.168
		20%	0.259	0.008	0.006	-0.004	0.026	0.398
		30%	0.614	0.011	0.007	-0.023	0.028	0.531
	Mesh	10%	0.121	0.003	0.001	0.053	0.028	0.107
		20%	0.167	0.006	0.002	0.063	0.031	0.327
		30%	0.181	0.008	0.002	0.132	0.023	0.425

compromised switches introduce systematic and directionally biased delays. This behavior reveals a trade-off in which topological uniformity can unintentionally amplify minor, consistent timing manipulations, especially when attacker placement coincides with critical synchronization paths.

2) Robustness of CBS against Desynchronization: Deviations in gPTP messages can indirectly affect CBS performance, as traffic bursts may arrive earlier or later than expected at devices downstream, leading to uneven credit accumulation. If CBS performance degrades under such conditions, more tightly synchronized TSN mechanisms, such as TAS, are likely to experience even more severe degradation in the presence of synchronization attacks.

To investigate the impact of desynchronization on time-sensitive traffic delivery in CBS-scheduled networks, we performed a two-stage analysis. First, we evaluate how timing deviations of ± 5 µs per gPTP message across different network topologies impact deadline adherence for various TSN flows. Secondly, we conducted a fine-grained sensitivity analysis, introducing minimal alternating offsets between switches. This analysis illustrates how even standard-compliant offsets can lead to observable deadline violations.

a) Impact of Network Topology: To reflect the impact of synchronization disturbances due to an attack, we introduce timing irregularities derived from our observed attack scenarios in Table I. For this purpose, we generate a representative TSN traffic mix with different priority levels (Priority Code Points, PCP) corresponding to TSN traffic classes. Attacked high-criticality gPTP messages are assigned the highest priority (7), while video streams use medium priority (4) and best-effort flows low priority (0). Each flow type was configured with realistic transmission rates to reflect practical deployment scenarios. We first examine the baseline scenario without synchronization disruptions, measuring the per-packet delay deviation from each flow's median. The same evaluation is then applied to attack-affected scenarios to analyze how flow performance degrades under varying delay-tolerance thresholds.

Fig. 3 shows the percentage of missed deadlines as a function of the allowed deviation from the baseline median delay. Results indicate that higher-priority flows consistently outperform best-effort flows regarding missed deadlines across all topologies. CBS schedules traffic mainly based on accumulated credit rather than local clock deviations. Deadlines start to be missed only when timing errors accumulate enough to distort credit evolution and release times. As expected in TSN, higher-priority flows are given precedence in resource allocation, allowing them to maintain their schedule even when desynchronization happens. Results also indicate that under strict timing requirements (0-5% deviation), over 90% of all flows across all topologies miss their deadline. This high sensitivity persists at 10% tolerance, which marks the percentage where time-aware traffic in TSN is accepted in worst-case [22].

The evaluation further indicates that in regular topologies

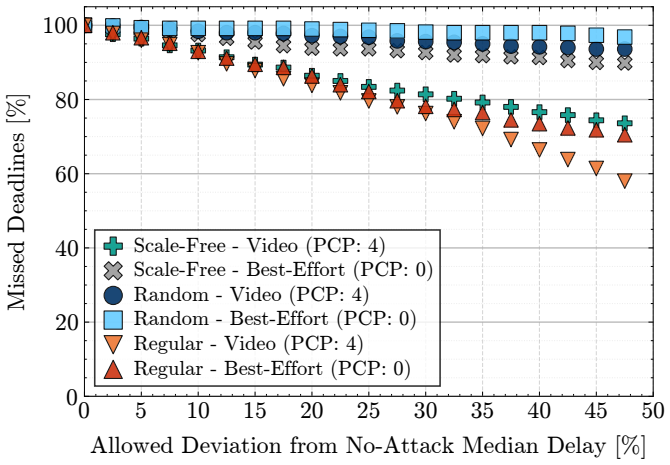

Fig. 3: Missed deadline percentages for mixed-criticality traffic in CBS-scheduled networks under attack.

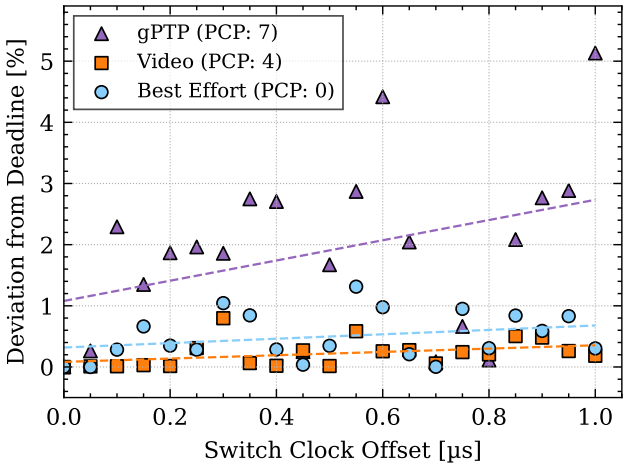

Fig. 4: Impact of switch clock offset on the deviation from deadlines for gPTP, Video, and Best Effort TSN flows in CBS-scheduled networks.

and at 20% deviation, packets from video flows experience about 85% missed deadlines, whereas best-effort flows are significantly worse at around 88% missed deadlines. Similarly to the time synchronization effects in Table I, random topologies consistently suffer the most from scheduling degradation due to timing disruptions. Regular topologies exhibit more robust behavior, particularly for high-priority flows. In scale-free topologies, video flows decrease to approximately 74% missed deadlines at 47% deviation, while best-effort flows remain above 90% across the evaluated range.

b) Impact of Minimal Switch Offsets: To complement the topology-focused analysis, we conduct a fine-grained sensitivity analysis that isolates the impact of minimal clock offsets between adjacent switches. These offsets remain within standard-compliant synchronization tolerances and reflect realistic imperfections in distributed time synchronization [1]. The sensitivity analysis is based on a minimal TSN topology consisting of two end hosts and three CBS-enabled switches. The same representative TSN traffic mix used in previous experiments is generated with diffferent priority levels: a high-priority gPTP stream, a medium priority video stream, and best-effort traffic.

As a baseline, we first simulated the scenario without clock offsets. Then, small, controlled clock offsets were injected into the switches, applied alternately to consecutive switches along the forwarding path, while all other parameters remained fixed. The offset magnitude was varied over the range 0 to 1 µ. For each offset, we measured the deviation from the deadlines across all packets of each flow and normalized the results relative to the no-offset baseline for the three traffic classes.

Under ideal conditions (no offsets), all flows meet their deadlines and deviations are negligible. Figure 4 shows the normalized deviation of per-packet delays from their no-offset baseline deadlines as a function of the injected switch clock offset for three traffic classes: high-priority gPTP, video, and best-effort traffic. Each data point is a simulation instance for a given switch clock offset. The lines show linear regression fits, to depict the overall sensitivity of each traffic class to increasing offsets. For zero offset, all flows meet their deadlines and deviations are negligible. Even sub-microsecond offsets, however, cause increasing deadline deviations for all classes. The effect is already visible below 0.2 µs and increases approximately linearly with the offset magnitude. High-priority traffic is most affected, followed by video and best-effort traffic. This reflects the tighter timing constraints of high-priority streams and aligns with prior work showing that such traffic is more sensitive to timing imperfections than lower-priority classes, while CBS maintains class prioritization [23], [24].

The sawtooth-like pattern of the high-priority curve results from the discrete nature of CBS credit updates and bursty traffic production. Small offset changes shift packet bursts relative to local credit evolution, causing abrupt transitions between credit windows and non-linear delay variations, as observed for CBS/ATS shapers under slight timing shifts [25]. Lower-priority flows, such as video and best-effort traffic, show flatter curves due to longer deadlines and lower scheduling priorities, making them less sensitive to these fine-grained burst credit alignments.

The sensitivity analysis results indicate that even clock offsets within gPTP specification can significantly impair hard real-time guarantees for high-priority flows, despite CBS prioritization. This highlights the importance of clock synchronization accuracy as a critical design parameter for TSN systems with stringent deadlines. It furthermore motivates the need for robust jitter suppression under realistic synchronization conditions and for hybrid shaping (CBS and TAS).

V. Conclusion

This work presents the first systematic study of gPTP time synchronization attacks and their cascading effects on TSN real-time scheduling across diverse network topologies. We implemented three realistic attacks in OMNeT++, each targeting different layers of the gPTP synchronization mechanisms:

a malicious GM distributing falsified time, a changing subset of compromised switches injecting offset errors, and malicious TSN switches introducing asymmetric delays.

We conducted extensive simulations across different topologies to evaluate how topological structure influences the severity of synchronization degradation. We quantified the end-to-end impact on mixed-criticality TSN traffic scheduled with IEEE 802.1Qav Credit-Based Shaper, showing that modest per-message offset injections of 5 µs can accumulate into significant synchronization errors exceeding 100 µs within only 1 s. This deviation exceeds the ± 1 µs precision threshold for TSN by nearly two orders of magnitude [1], causing substantial flow deadline violations, increased latency, and compromised deterministic guarantees.

These findings highlight the susceptibility of TSN deployments to subtle time synchronization attacks, and emphasize the need for stronger protection mechanisms within gPTP to maintain the integrity and performance of time-sensitive applications. Using analytical modeling, simulations, and multiple performance metrics, we provide the first end-to-end quantification of synchronization degradation on the IEEE 802.1Qav Credit-Based Shaper, a mechanism that schedules traffic on credit accumulation rather than precise timing. Future work includes hardware validation, broader attacker scenarios, and investigation of mitigation mechanisms for secure TSN scheduling.

REFERENCES

[1] IEEE, *IEEE 802 - Timing and Synchronization for Time-Sensitive Applications*, IEEE Std. IEEE 802.1AS-Rev/D8.0, 2019.

[2] R. Kakade, J. Chou, and S. Torcato, "Vulnerability Analysis of Time Synchronization in Automotive Ethernet," *ArXiv*, vol. abs/2208.11878, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:251800193

[3] H. Li, D. Li, X. Zhang, G. Shou, Y. Hu, and Y. Liu, "A Security Management Architecture for Time Synchronization Towards High Precision Networks," *IEEE Access*, vol. 9, pp. 117 542–117 553, 2021.

[4] IEEE 802.1 Working Group, *IEEE Standard for Local and Metropolitan Area Networks—Timing and Synchronization for Time-Sensitive Applications—Amendment 3: Hot Standby and Clock Drift Error Reduction (IEEE 802.1ASdm)*, IEEE Std., 2024.

[5] F. Fischer and D. Merli, "Security Considerations for IEEE 802.1 Time-Sensitive Networking in Converged Industrial Networks," in *2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, 2022, pp. 1–7.

[6] R. Annessi, J. Fabini, F. Iglesias, and T. Zseby, "Encryption is Futile: Delay Attacks on High-Precision Clock Synchronization," *ArXiv*, vol. abs/1811.08569, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:53761106

[7] D. Calero and E. Fernandez, "Characterization of chip-scale atomic clock for GNSS navigation solutions," in *2015 International Association of Institutes of Navigation World Congress (IAIN)*. IEEE, 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:21016150

[8] E. Itkin and A. Wool, "A Security Analysis and Revised Security Extension for the Precision Time Protocol," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 1, pp. 22–34, 2020.

[9] M. Fotouhi, A. Buscemi, A. Boualouache, F. Jomrich, C. Koebel, and T. Engel, "Assessing the Impact of Attacks on an Automotive Ethernet Time Synchronization Testbed," in *2023 IEEE Vehicular Networking Conference (VNC)*, 2023, pp. 223–230.

[10] W. Alghamdi and M. Schukat, "Precision time protocol attack strategies and their resistance to existing security extensions," *Cybersecurity*, vol. 4, no. 1, p. 12, 2021. [Online]. Available: https://doi.org/10.1186/s42400-021-00080-y

[11] H. Wang and D. Zhang, "Detecting SYN flooding attacks," vol. 3, 07 2002, pp. 1530– 1539. [Online]. Available: https://api.semanticscholar.org/CorpusID:11960636

[12] A. Buscemi, M. Ponaka, M. Fotouhi, F. Jomrich, C. Köbel, and T. Engel, "An Intrusion Detection System Against Rogue Master Attacks on gPTP," *IEEE Vehicular Technology Conference*, pp. 1–7, 06 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:260917954

[13] E. Lisova, E. Uhlemann, J. Åkerberg, and M. Björkman, "Monitoring of Clock Synchronization in Cyber-Physical Systems: A Sensitivity Analysis," in *2017 International Conference on Internet of Things, Embedded Systems and Communications (IINTEC)*, 2017, pp. 134–139. [Online]. Available: https://api.semanticscholar.org/CorpusID:4571086

[14] T. Mizrahi, "A Game Theoretic Analysis of Delay Attacks against Time Synchronization Protocols," in *2012 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication Proceedings*, 2012, pp. 1–6. [Online]. Available: https://api.semanticscholar.org/CorpusID:34133070

[15] F. Luo, Z. Wang, and B. Zhang, "Impact Analysis and Detection of Time-Delay Attacks in Time-Sensitive Networking," *Computer Networks*, vol. 234, p. 109936, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S138912862300381X

[16] F. Rezabek, M. Helm, T. Leonhardt, and G. Carle, "PTP Security Measures and their Impact on Synchronization Accuracy," in *2022 18th International Conference on Network and Service Management (CNSM)*, 2022, pp. 109–117. [Online]. Available: https://api.semanticscholar.org/CorpusID:254155784

[17] M. Fotouhi, A. Buscemi, F. Jomrich, C. Koebel, and T. Engel, "Evaluation of PTP Security Controls on gPTP," in *2023 IEEE Symposium on Computers and Communications (ISCC)*, 2023, pp. 783–789.

[18] A. Finkenzeller, O. Butowski, E. Regnath, M. Hamad, and S. Steinhorst, "PTPsec: Securing the Precision Time Protocol Against Time Delay Attacks Using Cyclic Path Asymmetry Analysis," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM 2024)*, 05 2024, pp. 461–470. [Online]. Available: https://api.semanticscholar.org/CorpusID:267061267

[19] C. Brülhart, N. S. Bülbül, N. O. Tippenhauer, and M. Fischer, "Transparent TSN for Agnostic End-hosts via P4-based Traffic Characterization at Switches," in *2024 IEEE 49th Conference on Local Computer Networks (LCN)*. IEEE, 2024, pp. 1–8.

[20] Rockwell Automation, *EtherNet/IP Network Infrastructure Recommendations*, Rockwell Automation, 2020, accessed: 2025-05-25. [Online]. Available: https://literature.rockwellautomation.com/idc/groups/literature/documents/rm/enet-rm001_-en-p.pdf

[21] "IEC/IEEE International Standard - Communication Networks and Systems for Power Utility Automation - Precision Time Protocol Profile for Power Utility Automation," *IEC/IEEE 61850-9-3 Edition 1.0 2016-05*, pp. 1–18, 2016.

[22] T. Zhang, G. Wang, C. Xue, J. Wang, M. Nixon, and S. Han, "Time-Sensitive Networking (TSN) for Industrial Automation: Current Advances and Future Directions," *ACM Comput. Surv.*, vol. 57, no. 2, 2024.

[23] L. Zhao, P. Pop, and S. Steinhorst, "Quantitative performance comparison of various traffic shapers in time-sensitive networking," *IEEE Transactions on Network and Service Management*, vol. 19, no. 3, pp. 2899–2928, 2022.

[24] M. M. Hasan, H. Feng, S. Khan, M. I. Ullah, M. T. Hasan, and B. Gain, "Timing analysis for optimal points in credit-based shaper of time sensitive network," in *2021 IEEE 6th International Conference on Signal and Image Processing (ICSIP)*, 2021, pp. 1102–1108.

[25] E. Mohammadpour, E. Stai, M. Mohiuddin, and J.-Y. Le Boudec, "Latency and backlog bounds in time-sensitive networking with credit based shapers and asynchronous traffic shaping," in *2018 30th International Teletraffic Congress (ITC 30)*, vol. 02, 2018, pp. 1–6.