Analysis of Time Synchronization for 6G-TSN Networks with Hot Standby

Mahin Ahmed*, Lucas Haug†, Raheeb Muzaffar*, Damir Hamidovic*, Armin Hadziaganovic*, Hans-Peter Bernhard*‡, Marilet De Andrade§, and János Farkas¶
*Silicon Austria Labs GmbH, Austria
†University of Stuttgart, Germany
‡Johannes Kepler University, Austria
§Ericsson Research, Sweden
¶Ericsson Research, Hungary
mahin.ahmed@silicon-austria.com

***Abstract*—Reliable time synchronization is of utmost importance within 6G networks, particularly when envisioning their role in future industrial automation scenarios that demand deterministic communication. In particular, we focus on the integration of 6G with time-sensitive networking (TSN). Traditionally, the best timeTransmitter clock algorithm (BTCA) is used to dynamically select a grandmaster (GM) for TSN nodes in a 6G-TSN network. However, BTCA's performance in case of link or device failures is categorized to be slow. The IEEE 802.1ASdm standard addresses this limitation by introducing a hot standby GM in the network. In this paper, we extend the hot standby amendment to 6G-TSN networks to support continuous time synchronization. For the analysis, we develop a simulation framework in OMNeT++. We analyze the performance in terms of clock drift and out-of-sync time for different time synchronization scenarios in case of failures. Having two synchronized GMs in the network reduces the out-of-sync time in case of failure when compared to BTCA.**

***Index Terms*—Time synchronization, 5G/6G, TSN, 5G-TSN, 6G-TSN, BTCA, BMCA**

I. Introduction

In the rapidly advancing realm of 6G networks, time-sensitive networking (TSN) has become an essential framework for enabling deterministic communication with extremely low latency and jitter [1], [2]. This is crucial for applications such as autonomous driving, industrial automation, smart grids, and mission-critical communications, where even the slightest timing inaccuracies can lead to undesirable behavior. TSN aims to provide highly accurate and reliable time synchronization across network devices to ensure coordinated actions and seamless data flow. To address this, the IEEE 802.1AS [3] standard defines a precision time protocol (PTP) profile [4], the generic precision time protocol (gPTP). gPTP uses a simplified version of the best timetransmitter clock algorithm (BTCA) to dynamically select a grandmaster (GM) clock within a domain [4]. In the upcoming sections, we follow the inclusive terminology as specified in [5] amendment to describe the time synchronization. Whereby the following should be noted: master is replaced with timeTransmitter (tT), slave is replaced with timeReceiver (tR), GM remains grandmaster, and best master clock algorithm (BMCA) is replaced with BTCA.

The integration of 5G with TSN networks has been standardized by 3GPP, where the 5G system (5GS) acts as a (virtual) time-aware system for the time synchronization perspective [6] (as shown in Fig. 1). This integrated network is referred to as a 5G-TSN network. The TSN integration with 6G networks relies on the backward compatibility of 6G specifications. The gPTP time synchronization of TSN nodes (i.e., end stations and bridges) connected via the 5G virtual TSN bridge is managed by the two functional entities at the boundaries of the 5GS. As in any gPTP network, BTCA is used for selecting a GM for the TSN nodes in the 5G-TSN network. However, BTCA's response to link or device failures is considered slow. During the election of a new GM, the clocks of the tR nodes continue running independently, resulting in an unsynchronized network. The election process can take several hundred seconds, primarily due to BTCA and clock servo mechanism's recovery time [7]. Additionally, it is unable to recognize transient faults in clocks, hence leading to an unstable time synchronization [8]. The IEEE 802.1ASdm standard addresses these shortcomings by disabling BTCA and enabling a redundant hot standby GM and time domain for wired networks.

For 6G-TSN network, a hot standby redundancy in GM has not been defined yet. There are a few open questions that need to be addressed when extending the hot standby to 6G-TSN networks. In this paper, we address these questions by extending the hot standby in 6G-TSN networks. We first provide an overview of the important considerations for the placement of the two GMs in 6G-TSN networks. In particular, our contributions include the following.

- We build upon the concept of a hot standby GM and extend it to 6G-TSN networks.
- We provide a simulation framework in OMNeT++ for the analysis. It uses real-world measurement data for 6G packet delay [9]. The simulator will be provided open-source on GitHub.
- We analyze the time synchronization performance with hot standby in 6G-TSN networks for different time synchronization scenarios. The simulation results show a reduced out-of-sync duration when using hot standby

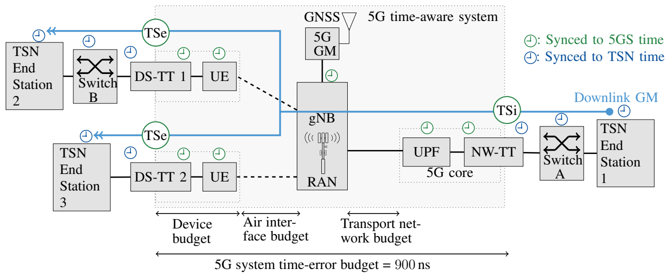


Fig. 1. Time synchronization in 5G-TSN network where 5GS operates as a time-aware system. Time synchronization with a downlink GM is shown.

compared to traditional BTCA, thus supporting continuous time synchronization.

The rest of the paper is organized as follows: Section II presents related work on time synchronization in 5G-TSN networks. Section III explains the time synchronization mechanisms in 5G-TSN networks with a focus on BTCA. Section IV details the hot standby GM operation in 6G-TSN networks. Section V presents the simulator design. Section VI analyzes different time synchronization scenarios in 6G-TSN network. Lastly, section VII concludes the paper with some discussions.

II. Related work

Time synchronization in 5G-TSN network is an extensively researched topic. As related work, we present the research focusing on the performance of BTCA and GM locations. Gutiérrez et al. [10] demonstrated the cumulative impact of PHY jitter and clock granularity as synchronization information propagates further from the GM. Their research, conducted within a TSN network, showed that synchronization accuracy degrades from $0.6\,\mu$s to approximately $2\,\mu$s at the final end station in a 100-hop chain. In the context of 5G-TSN networks, Schüngel et al. introduced an enhanced BTCA, which takes into account the relative positioning of the GM [11]. This algorithm factors in network-related data to better account for architectural influences when selecting a GM. Additionally, the authors studied synchronization performance in a 5G-TSN network, considering different GM placements [12]. Their results indicated that the error introduced by combining 5G with TSN adds an extra synchronization error equivalent to around 36 additional hops. To address error accumulation in traditional BTCA, Jeon et al. proposed using a pre-configured ordered list of GMs to reconfigure the GM, helping to mitigate the effects of successive message exchanges [13]. To this end, time synchronization with hot standby GM in 6G-TSN networks has not been studied so far. Our previous work [14], [15], focused on presenting the first ideas for extending the hot standby GM for 6G-TSN networks. In this paper, we analyze the time synchronization performance in the presence of a hot standby.

III. Time synchronization in 5G-TSN networks

Within a 5G-TSN network, different synchronization mechanisms work together to achieve end-to-end (E2E) time synchronization, as shown in Fig. 1. The components of the 5GS, e.g., the gNodeB (gNB), user plane function (UPF), and user equipment (UE), are synchronized via internal 5GS synchronization mechanisms [16], [17]. These components synchronize to the internal 5GS clock, which receives time from a global navigation satellite system (GNSS). On the other hand, the TSN nodes synchronize using the PTP profile gPTP [3] to a GM clock located either within the TSN network or in the 5G network. An instance of the gPTP protocol operating in a single device within a single domain is termed a PTP instance. A device containing one or more PTP instances is termed a time-aware system. The gPTP synchronization is implemented by the two functional entities at the boundaries of the 5GS: the device side TSN translator (DS-TT) at the UE and the network side TSN translator (NW-TT) at the UPF. The DS-TT and NW-TT generate gPTP messages for their respective tT ports. The NW-TT generates gPTP messages on behalf of the DS-TT, if the DS-TT does not support this. The internal 5GS synchronization mechanisms are used to synchronize the UE/DS-TT [16] and the NW-TT [17]. The PTP messages are timestamped at the ingress (TSi) and at the egress (TSe) of the 5GS to calculate the residence time, as shown in Fig. 1. A time error contribution between the ingress and egress of the 5GS for the synchronization messages has been specified by the 3GPP to be no more than 900 ns [18].

The GM location introduces dependencies among the different synchronization mechanisms [10]. To address this, 3GPP has standardized three options for a GM to synchronize the TSN nodes in 5G-TSN networks [6]. First, in *downlink synchronization* (as illustrated in Fig. 1), the GM for TSN nodes is located on the network side (connected to the UPF) of the 5GS . The synchronization error for TSN nodes beyond the 5GS is influenced by the accuracy of the 5GS residence time. This accuracy greatly depends on the 5GS clock based synchronization of NW-TT and DS-TT. Second, in *uplink synchronization*, the GM for TSN nodes is located on the device side (connected to a UE) of the 5GS. Here the gPTP messages required to synchronize other TSN devices on the device side must traverse the wireless link (air interface, shown with dashed line in Fig. 1) between gNB and UE twice. Similar to downlink synchronization, the synchronization error is greatly influenced by the accuracy of residence time estimation, which relies on the 5G clock. For the air interface, the time error can increase substantially, potentially exceeding the required 900 ns time error budget. To mitigate this, 3GPP has proposed adding the air interface propagation time to the time reference, thus minimizing the time error by accounting for the propagation delay [11]. Third, the *time synchronization as a service* enables external networks, such as TSN networks to access the 5G timing as introduced in Rel-17 [19]. As the quality of 5GS clocks is constantly improving, it is becoming less dependent on the GNSS or even can be considered as a terrestrial backup for GNSS. With Rel-18, 3GPP has introduced a timing resiliency service for an external network, which monitors the performance of the 5G time synchronization and provides a status report. If a status change (degradation, failure, or improvement) is detected, the next generation-radio access network (NG-RAN) sends either a high level status report or detailed clock quality metrics to the UE using the service [19]. This support for timing resiliency allows for the detection of faults and degradation in quality of the time synchronization. Additionally, it allows to report the 5G clock quality metrics to external networks hence allowing for the 5GS clock to be used as GM in external networks employing BTCA.

A. Best timeTransmitter Clock Algorithm (BTCA)

When using gPTP based time synchronization, BTCA is used to dynamically select the GM within each gPTP domain in a time-aware network, as specified in [4] and [3]. The synchronization hierarchy is established based on the port decisions made independently at each PTP instance. In other words, the PTP instances do not negotiate which instance should be the GM. The synchronization hierarchy is established as spanning tree to prevent synchronization loops.

BTCA consists of two parts: a dataset comparison algorithm and a port decision algorithm. The *dataset comparison algorithm* makes use of the information contained in the *Announce* messages. Each time-aware system sends out *Announce* messages to other time-aware systems within a gPTP time domain. The *Announce* messages contain information about the clock and topology attributes like *priority1, clockClass, clockAccuracy, offsetScaledLogVariance, priority2, and clockIdentity*, ordered from the most to the least significant attribute [4]. The dataset comparison algorithm compares the information contained in the *Announce* messages received at each port to the local clock to find the best clock at each port. The dataset comparison algorithm is leveraged again to find the overall best clock for each PTP instance. The *port decision algorithm* runs on each PTP instance and sets the port states based on the best data set per PTP instance, the best dataset per port, and the local dataset. A port can either be tT, tR, or passive. tT ports send out synchronization messages, whereas the tR ports receive them. Lastly, the passive port states help avoid loops in the synchronization hierarchy.

If the current GM fails or ceases to transmit timing information, BTCA is triggered to select a new GM after a *announceReceiptTimeout*. On the one hand, BTCA's response in the event of a failure is considered to be slow as it introduces a downtime until a new GM can be found [7]. This would be undesireable for many time-critical applications. This downtime in case of a failure is presented in Sec.VI. Moreover, BTCA does not have a detection mechanism to detect transient faults within a GM clock. These transient faults, when not promptly identified, can lead to undesirable consequences, including what is colloquially referred to as a "ping-pong effect". This phenomenon unfolds as BTCA tirelessly searches for the best tT in the network, often switching between multiple candidates without achieving stable synchronization [8].

IV. Hot standby in 6G-TSN networks

Both 3GPP and IEEE 802.1 standards support multiple GMs and time domains to meet the needs of industrial applications requiring both universal and working time, as well as to enhance fault tolerance. Each domain is identified by a unique domain number. These time domains are currently independent, meaning the different GMs do not synchronize with one another. Furthermore, 3GPP allows for local configuration to determine the GM and the synchronization spanning tree.

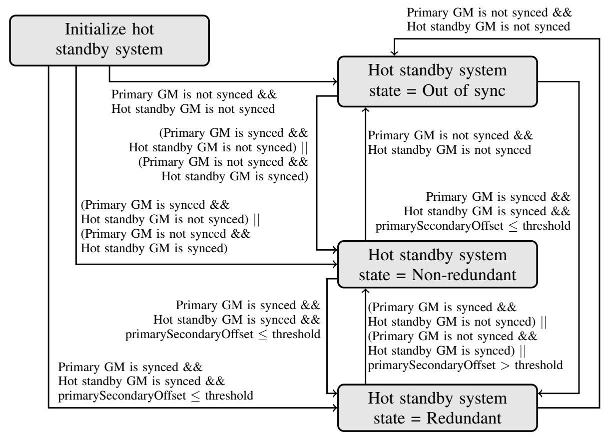


Fig. 2. Hot standby system state machine, abstracted from [20].

The IEEE 802.1ASdm standard [20] (the hot standby amendment) addresses the limitations of BTCA. It specifies the process of selecting and configuring multiple GMs in a time-aware network, hence enabling redundancy in GM with a hot standby system. The hot standby amendment eliminates the use of BTCA within a domain. Instead, it relies on the external port configuration option to implement a static configuration. The two GMs are selected by an external management entity (e.g., the TSN network controller). With hot standby system enabled, each time-aware system simultaneously operates two PTP instances, each within its own domain. The two domains are referred to as the primary domain and the hot standby domain, with their respective GMs referred to as the primary GM and the hot standby GM. A time-aware system uses the primary domain for its application as long as the hot standby system is in the redundant state (i.e., both domains are available). Only if the primary domain is no longer available, the time-aware systems begins using the hot standby domain. The hot standby GM synchronizes itself with the primary GM before transmitting timing messages within its domain. This procedure ensures that the time provided by both GMs remains within a tolerance range, resulting in consistent and continuous time provision. The hot standby system keeps track of the difference between the primary and the hot standby GM time using the *primarySecondaryOffset* variable. If the *primarySecondaryOffset* is above a threshold, the hot standby system transitions from a redundant to a non-redundant state (i.e., only one domain is available). The hot standby state machine is depicted in Fig. 2. Here "is synced" means that the synchronization requirements of the PTP instance are fulfilled. The hot standby amendment proposes an optional split functionality which provides an interworking function that transfers synchronized time from a synchronized PTP instance to an out-of-sync PTP instance. Thus, the split functionality brings the *primarySecondaryOffset* within the threshold and allows the hot standby mechanism to recover from a non-redundant state to a redundant state (i.e., two domains are available).

While both the 3GPP [21] and IEEE 802.1 [22] standardization bodies acknowledge the need for continuous time synchronization by introducing support for multiple GMs and multiple synchronization paths, limitations persist as mentioned above. Redundancy in GM could be a key enabler for achieving resilient, fault-tolerant, and continuous time synchronization in the event of failures. Hence, in the context of 6G networks looking into a solution similar to IEEE 802.1ASdm, i.e., a hot standby GM could be of interest to reduce the downtime period in case of failures for the time-critical applications. When enabling hot standby for 6G-TSN networks, some important considerations related to the network architecture and clock quality should be taken into account. These considerations are presented in our previous work [14]. Here we present a short overview for completeness. The placement of the two GMs in 6G-TSN networks could introduce some dependencies in the time synchronization quality as also pointed out by the related work [10]–[13]. When considering possible options for the placement of the two GMs in the light of the standardized time synchronization mechanisms (see Sec. III), following should be considered:

- Using the 6G GM as one of the GMs could benefit from the higher quality (access to GNSS time) 6G clock. In the case of failure of the primary (6G) GM (e.g., loses access to GNSS time), the gNB can still support time synchronization for a certain time while being in a holdover state.
- The 6G clock is used to synchronize the other 6G elements like the gNB, UPF, and other UEs for their proper functioning. Hence, it is not desirable that the 6G GM synchronized to an external network (non 6G clock), making it not a suitable choice for a hot standby GM.
- Similarly, for use cases involving scenarios where one of the GMs is the 6G GM, the optional split functionality should be set to FALSE. This will prevent the hot standby GM from attempting to synchronize the 6G GM's clock once the 6G GM or 6G connection is restored after a failure.
- Having the GM on the network side helps keeping the time-error low for device side TSN end-stations given the timing messages only travel once over the air interface.

V. Simulator design

In order to analyze the hot standby in 6G-TSN networks in the case of failures, we extend the INET simulation framework [23] for TSN features and the DetCom simulator framework [24] for 6G-TSN networks in the OMNeT++ simulator [25]. In this section, we introduce our extensions to these frameworks to allow for a full evaluation of BTCA as well as hot standby. Our implementation will be open-sourced along with this paper on GitHub.

A. Extensions to INET's gPTP implementation

The INET framework already provides a gPTP implementation which includes two distinct device types: TsnDevices and TsnSwitches, representing PTP ordinary and boundary clocks, respectively. Each device can have one or more local clocks, each equipped with different types of oscillators. Real oscillators are affected by imperfections like oscillator aging, temperature gradient, and vibrations causing the clocks to drift. The oscillator of a clock determines its drift behavior, including the drift rate relative to simulation time and its variations. The clock itself can have additional functionalities, such as adjusting its time or compensating for drift by modifying the oscillator settings. Multiple clocks within a single device can be grouped into a *multiClock*. While each clock in a *multiClock* continues to operate independently, the *multiClock* presents itself externally as a single clock. The *activeClock* parameter specifies which clock is currently in use. Each device can have one or multiple gPTP instances, which manage the processing of incoming and outgoing gPTP messages. Each gPTP instance has a unique domain number and has configurable parameters, such as the *syncInterval*. Additionally, each gPTP instance is linked to a single clock, which provides timestamps for gPTP messages. Upon receipt of a *SYNC* message, the gPTP instance instructs its associated clock to adjust its time according the clock's implementation.

As these basic features are insufficient for our evaluation needs, we extend the INET gPTP implementation by clock servos, a BTCA, and a fully-functional hot standby operation.

1) Clock servo: The clock servo is responsible for correcting the drift between tR and tT clocks, thus minimizing clock jitter and maintaining the clock drift within the required limits [26]. Currently INET's gPTP implementation, does not provide any clock servos. A clock without a clock servo is termed as a *SettableClock*. It jumps to the time calculated by the gPTP component and estimates the drift rate using the last *SYNC* message. The clock drift for the *SettableClock* is shown in Fig. 3 in green color. This jumping behavior can introduce several challenges. If a device event or a message is scheduled for transmission at a specific time, a sudden forward jump in time may cause the event to be skipped or, if executed immediately, may lead to conflicts with other events. Conversely, a backward time jump could result in events being processed multiple times if not handled correctly. To mitigate these issues, we introduce a proportional-integral (PI) controller based clock servo. The PI controller adjusts the clock drift by speeding up or slowing down the clock and thus maintains the synchronization to the tT without performing any jumps [27]. Given, the simplicity of a PI controller, it is a common choice for PTP-synchronized clocks and is also used in real-world gPTP implementations, such as gPTP for Linux (ptp4l) [28]. The behavior of the PIServoClock is shown with orange color in Fig. 3.

2) BTCA: Currently, INET's gPTP does not provide an implementation of the BTCA. We therefore extend the INET framework by adding a complete BTCA implementation. Our implementation adheres to the fundamental requirements of BTCA as presented in Sec. III-A, and specified in [3]. In the event of a link or device failure that results in the loss of synchronization with the current GM, the BTCA is automatically re-executed after a predefined *announceReceiptTimeout* interval. This ensures that the network can dynamically reselect a new GM when necessary, thus maintaining clock synchronization and minimizing disruption.

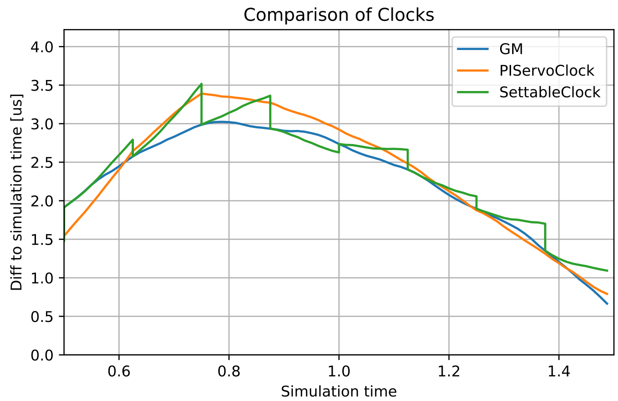


Fig. 3. A comparison of different clock servos. The tT clock is presented with the blue line. A tR clock without a clock servo is presented with the green line, whereas a tR clock with a PI controller based clock servo is presented with the orange line.

3) Hot standby: INET already supports multiple gPTP instances for different domains in a single network device. By assigning each gPTP instance to a separate clock, a device can achieve independent synchronization in each of its gPTP domains.

In order to evaluate hot standby in the 6G-TSN network, we incorporate the hot standby system into the INET's gPTP implementation. The synchronization state of a domain changes following a *syncReceiptTimeout* event. By default of the gPTP standard [3] this *syncReceiptTimeout* equals three consecutive *syncInterval* periods. The hot standby system orchestrates the switch between a redundant and a non-redundant by updating the *activeClock* parameters based on the hot standby system state. If the *primarySecondaryOffset* is above a threshold, the hot standby system transitions from a redundant to a non-redundant state (i.e., only one domain is available), as shown in Fig. 2.

B. Extensions to the 6G DetCom simulator framework

As the 6G system (6GS) integrates with TSN as a time-aware system, the main difference from the TSN network perspective is the added packet delay inside the 6GS. This packet delay is compensated for using the residence time (see Sec. III). The DetCom simulator framework provides an implementation of a 6G-TSN bridge with DS-TTs and the NW-TT. It hides the 6G internals by replacing them with real-world measurements of the characteristic packet delay of different 5G testbeds and trials, thus facilitating evaluation of concepts for 6G-TSN networks. The details on the measurement data sets is provided in [24].

For the analysis of time synchronization performance, we extend the DetCom simulator framework, with gPTP support at TSN translators. In particular, we implement the timestamping of gPTP messages at the two TSN translators at the boundaries of the 6GS. These timestamps allow for the calculation of the rate ratio between the 6GS and TSN nodes, as well as the determination of residence time.

In order to provide a complete framework for the evaluation of gPTP in 6G-TSN networks, we combine our new INET gPTP implementation with our new TSN translator functionality. To this end, we developed a modified gPTP implementation for the use in TSN translators. This implementation prevents the transmission of *Pdelay_Req* and *Pdelay_Resp* messages between TSN translators, as delay measurement within the 6G system are handled by the newly added timestamping mechanism. Additionally, it manages synchronization between the 6G and TSN clocks when a TSN translator is designated as the GM of a domain.

VI. Analysis of hot standby operation

Using the simulator presented above, we analyze the performance of time synchronization in 6G-TSN networks with hot standby in the case of failures for two different time synchronization architectures. The simulation results are compared against a 6G-TSN network with BTCA enabled.

A. Simulation setup

The 6G-TSN network is simulated as shown in Fig. 4 and Fig. 5. The TSN end station 1 is connected to the NW-TT via TSN switch A. The TSN end station 2 is connected to the DS-TT 1 via TSN switch B. The TSN end station 3 is connected to DS-TT 2. In the following we use the term station when referring to a TSN end station. The 6GS is simulated by the 6G DetCom node [24] with uplink and downlink delays as per the *PD-Wireless-5G-1* dataset [9]. The 6G clocks of the TSN translators are configured to drift apart by a maximum of $\pm 450\,\mathrm{ns}$, to ensure compliance with 3GPP's budget of $900\,\mathrm{ns}$ [18] between two TSN translators. The drift rate for TSN clocks is randomly assigned in alignment with the time deviation (TDEV) requirements specified in the IEEE 802.1AS standard [3], but simplified to a maximum TDEV of $5\,\mathrm{ns}$ over an observation interval of $1\,\mathrm{s}$ and a maximum frequency offset of $100\,\mathrm{ppm}$.

All gPTP compliant devices are configured according to the parameter settings recommended in the IEEE 802.1AS standard. Specifically, we set the intervals for the *SYNC* messages to $125\,\mathrm{ms}$ and to $1\,\mathrm{s}$ for the *Announce* message, each with a timeout after 3 intervals. We simulate two different BTCA implementations. One, BTCA where the *Announce* messages are sent immediately to neighboring nodes if BTCA execution results in the selection of a new GM. And an alternative implementation, BTCA2 where the PTP instance waits until the end of the current announce interval to send out the *Announce* messages.

B. Simulation scenarios

In the view of the important consideration for GM location as presented in Sec. IV, we consider three different time synchronization scenarios in 6G-TSN network.

First, in hot standby scenario 1, BTCA is disabled and hot standby system is enabled with a static configuration of GMs. The NW-TT, synchronized to the 6G GM, is chosen as the primary GM. It synchronizes the TSN nodes on either side of the 6GS. Whereas, the network side TSN end station 1 is chosen as the hot standby GM. It also synchronizes the TSN

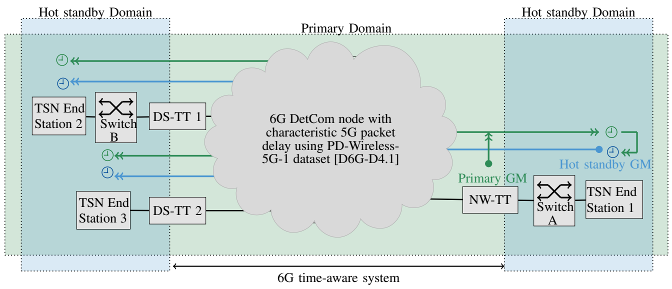


Fig. 4. Hot standby scenario 1: NW-TT as primary GM and network side TSN end station as hot standby GM.

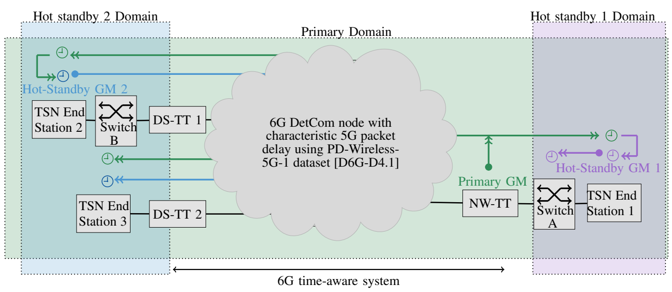


Fig. 5. Hot standby scenario 2: NW-TT as primary GM and the hot standby GM 1 on a network side TSN end station and the hot standby GM 2 on a device side TSN end station.

nodes on either side of the 6GS. The hot standby scenario 1 is depicted in Fig. 4. Here the primary domain is depicted in green color, whereas the hot standby domain is depicted in blue color. Second, in hot standby scenario 2, BTCA is disabled and hot standby system is enabled with a static configuration of GMs. The hot standby scenario 2 is shown in Fig. 5. The NW-TT, synchronized to the 6G GM, is chosen as the primary GM, as in the first scenario. It synchronizes the TSN nodes on either side of the 6G time-aware system. In this scenario, two independent hot standby GMs are used for different parts of the network. The hot standby GM 1 resides on the TSN end station 1 on the network side of the 6GS. It synchronizes the TSN nodes only on the network side of the 6GS (purple domain in Fig. 5). Whereas the hot standby GM 2 resides on the TSN end station 2 on the device side of the 6GS. It synchronizes the TSN nodes only on the device side of the 6GS (blue domain in Fig. 5). Third, in the BTCA scenario, we only use one time domain which uses BTCA to determine the GM. In order to allow for a fair comparison, we set clock attributes in the BTCA scenario that prioritize the NW-TT in the selection of the GM clock. Additionally, the *clockAccuracy* is also kept low for the TSN end station 1 and TSN end station 2, such that BTCA eventually selects the same GM as predefined in the hot standby scenarios.

For each time synchronization scenario, we simulate three different failure cases:

1) The NW-TT loses access to 6G timing. Hence, the NW-TT becomes GM incapable and stops transmitting *SYNC* messages as the GM. In the simulation, this failure occurs from 10 s to 50 s.

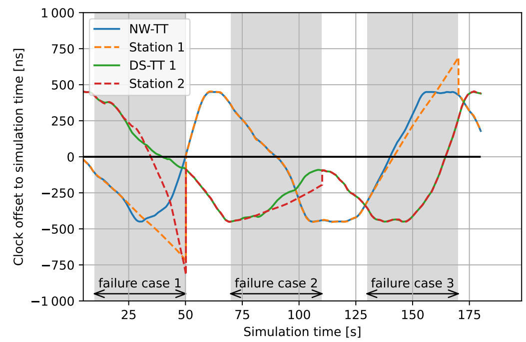


Fig. 6. Clock drifts for devices in the hot standby scenario 1 with drifting 6G clocks. The clock offset $= 0$ is shown with a black line.

2) The 6G network availability is compromised. In other words the network connection between the NW-TT and DS-TTs is lost. In the simulation, this failure occurs from 70 s to 110 s.
3) The connection between NW-TT and switch A is lost. In the simulation, this failure occurs from 130 s to 170 s.

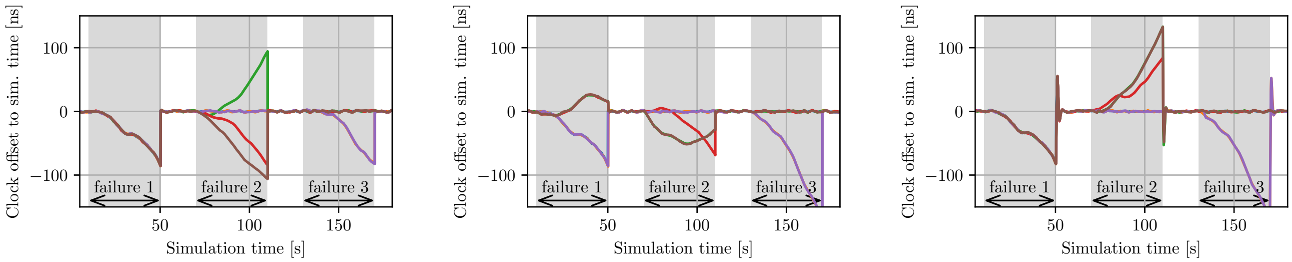
C. Simulation Results

In this section, we analyze the scenarios and failure cases introduced above based on different metrics, namely the influence of the 6G time-aware system, the adaptability to failures and the out-of-sync time.

1) Influence of the 6G time-aware system: First, we evaluate influence of the 6G system on the synchronization accuracy of the TSN system in 6G-TSN networks for hot standby scenario 1 in all three failure cases. Fig. 6 shows the clock drift of the 6G clocks located in the TSN translators and the drift of the TSN clocks located in stations 1 and 2. A clock offset of 0 is the ideal time. As can be seen, the total drift error for the 6GS remains within the 900 ns budget. During intervals without failure, e.g., between 50 s and 70 s, the clock drift of the TSN clocks tightly follows the clock drift of the 6G clocks. This indicates that a main source for the TSN clocks drifting apart is the inaccurate synchronization of the 6G system. Since the net effect of the failure cases is partly obfuscated by this drift of the 6G clocks, all subsequent simulations are based on an ideal 6G clock, i.e., free from any drift or offset.

2) Adaptability to failures: Next, we compare the adaptability in the different time synchronization scenarios across all three failure cases. Our analysis focuses on the clock offset of TSN nodes within the 6G-TSN network relative to the ideal clock under these scenarios, with results presented in Fig. 7. Since we consider idealized 6G clocks for this analysis and the NW-TT is the primary GM, the NW-TT always has a zero clock drift.

In the first failure case, where the NW-TT becomes GM incapable, a key distinction between the two hot standby scenarios becomes evident. In hot standby scenario 1, all devices seamlessly switch to the hot standby domain, with station 1 assuming the role of hot standby GM. Here, all TSN nodes are perfectly synchronized with each other. In contrast, in hot standby scenario 2, the network splits into two



(a) Hot standby scenario 1

(b) Hot standby scenario 2

(c) BTCA scenario

Fig. 7. Drift of TSN clocks in a setup with ideal 6G clocks. Each grey area corresponds to one of the three failure cases.

separately synchronized parts. This occurs because the device side synchronizes to hot standby GM 2 (station 2), while the network side synchronizes to hot standby GM 1 (station 1). As a result, the two sections of the network drift apart from each other, despite all network links remaining intact and theoretically capable of maintaining synchronization. In the BTCA scenario, station 1 is selected as the GM, thus resulting in a similar behavior of the TSN nodes as in hot standby scenario 1.

In the second failure case, where the connection between the NW-TT and DS-TTs is lost, a network split into multiple segments is unavoidable due to the lack of link redundancy. In both hot standby scenarios, the network side TSN nodes remain synchronized to the NW-TT as the primary GM. However, the device side TSN nodes respond differently depending on the hot standby scenario. In hot standby scenario 1, the device side TSN nodes lack an alternative synchronization source, causing their clocks to drift randomly apart. In contrast, in hot standby scenario 2, station 2 and switch B remain synchronized by the hot standby domain 2 and only station 3 drifts apart, as the link between the DS-TT 1 and DS-TT 2 is broken and there is no available hot standby GM to synchronize to. The BTCA scenario behaves similar to the hot standby scenario 2, eventually selecting station 2 as the new GM to keep switch B synchronized.

In the third failure case, where the link between the NW-TT and switch A is broken, we observe a similar behaviour for all three scenarios. The device side TSN nodes remain synchronized to the NW-TT. In both hot standby scenarios, station 1 is the hot standby GM for the network side TSN nodes. Similarly, in the BTCA scenario, station 1 is eventually selected as the GM by the BTCA algorithm.

In summary, when comparing BTCA and hot standby, we observe that in the BTCA scenario, in any failure case, there is only one group of clocks drifting away in the same direction from the ideal GM at the NW-TT per disconnected network part. This shows that BTCA is able to find exactly one new GM for all interconnected parts of the network as long as there is at least one GM-capable device in each separate part of the network. Hot standby with a static configuration, on the other hand, leads to parts of the network being without a GM if the hot standby domain is separated as well. Having different hot standby domains can additionally lead to devices synchronizing to two different GMs even though the network topology would still allow a single GM to be present.

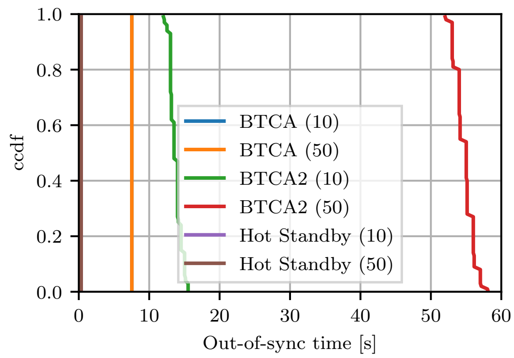


Fig. 8. Complementary cumulative distribution of the out-of-sync time of switch A. The number in parentheses indicates the number of hops between the switch A and the GM.

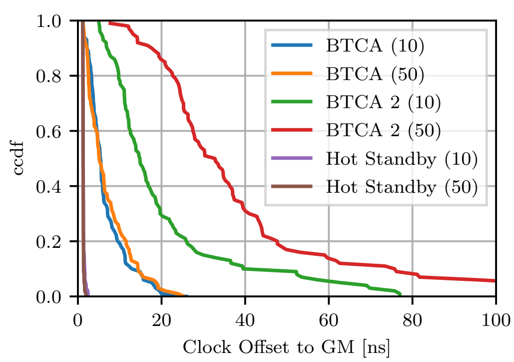


Fig. 9. Complementary cumulative distribution of the maximum offset between switch A and newly selected GM station 1. The number in parentheses indicates the number of hops between the switch A and the GM.

3) Out-of-sync time: Even though BTCA is able to adapt to changing network environments, its reaction time to failures is categorized as slow. In order to analyze this claim in more detail, we compare the out-of-sync time and the resulting clock offset for switch A in hot standby scenario 1 against the two BTCA implementations in failure case 3. In this failure case, in hot standby scenario 1, the primary domain is no longer available, and the switch A switches to the hot standby domain. In contrast, in the BTCA implementations, BTCA needs to select new GM.The out-of-sync time is defined as the time between a *syncReceiptTimeout* and a new *Sync* message.

In order to also analyze the impact of multiple hops when BTCA searches for a new GM, we generate results for 10 and 50 intermediate hops between switch A and station 1. The following results are based on 100 simulations for each simulation scenario. In Fig. 8, we present the out-of-sync time between switch A and the newly selected GM station 1. Additionally, Fig. 9 shows the cumulative distribution of the maximum offset between switch A and station 1. As expected, in the hot standby scenario, the out-of-sync time is minimal, as devices are able to switch to the hot standby domain almost immediately after the sync timeout. The short out-of-sync time also leads to a negligible maximum offset between switch A and station 1. Similarly, the effect of the number of hops is negligible, as the sync timeout occurs independently on every device and no new GM needs to be selected. The BTCA implementations, on the other hand, take longer to adapt to network failures compared to hot standby scenario 1. This longer out-of-sync time also leads to a greater clock offset. There are two main reasons why this happens. First, the Announce timeout it greater than the Sync timeout. Thus, the time until BTCA starts to react to the failure in the network is greater than in the hot standby scenario. The second reason is that BTCA need time to exchange Announce messages in order to select a new GM, while the hot standby GM is already preconfigured in the hot standby scenario. This effect especially becomes clear in BTCA2, which is highly affected by the number of intermediate hops. If Announce messages are not sent immediately, every additional hop introduces a longer out-of-sync time and thus, a greater maximum offset.

In summary, our results confirm the assumption that BTCA reacts slower to network failures and thus hot standby outperforms BTCA with regards to out-of-sync time and, as a consequence, also in the clock offset.

VII. Discussion and Conclusion

In this paper, we addressed the need for continuous and fault-tolerant time synchronization in 6G-TSN networks, which are poised to play a pivotal role in future industrial automation scenarios requiring deterministic communications. By extending the IEEE 802.1ASdm amendment to 6G-TSN networks, we demonstrated the effectiveness of employing a hot standby GM to overcome the limitations of BTCA in case of failures. We developed a simulation framework, which allows for detailed analysis of the time synchronization aspects of 6G-TSN networks. Scenarios with a hot standby GM outperform BTCA in terms of clock drift and out-of-sync time in case of failures. However, our analysis indicates that in certain failure cases, parts of the network may be left without a GM when using static GM configuration (hot standby). In contrast, BTCA is able to dynamically select a new GM in the event of failure, albeit with some delay. This leads us to conclude that the decision to use a static GM configuration with hot standby or a dynamic GM configuration with BTCA should be based on the network topology and the presence of other redundancy measures, such as link redundancy. For instance, in a network with multiple redundant links, a static GM configuration with hot standby might be more effective. Conversely, in a network with fewer redundancy measures, a dynamic GM configuration could provide better resilience by a dynamically selecting new GM in case of failure.

VIII. Acknowledgements

This work was supported by the European Union's Horizon Europe project DETERMINISTIC6G under grant agreement No. 101096504.

References

[1] G. P. Sharma et al., "Toward deterministic communications in 6G networks: State of the art, open challenges and the way forward," *IEEE Access*, vol. 11, pp. 106 898–106 923, 2023.

[2] R. Muzaffar et al., "5G deployment models and configuration choices for industrial cyber-physical systems–a state of art overview," *IEEE Trans. on Ind. Cyber-Physical Systems*, 2023.

[3] IEEE, "IEEE Std 802.1AS-2020: IEEE standard for local and metropolitan area networks–timing and synchronization for time-sensitive applications," 2020.

[4] ——, "IEEE Std 1588-2019 (Revision of IEEE Std 1588-2008): IEEE standard for a precision clock synchronization protocol for networked measurement and control systems," 2019.

[5] ——, "IEEE Std 1588g-2022: IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems Amendment 2: Master-Slave Optional Alternative Terminology," 2023.

[6] M. K. Atiq et al., "When IEEE 802.11 and 5G meet time-sensitive networking," *IEEE Open J. of the Ind. Electron. Soc.*, vol. 3, pp. 14–36, 2021.

[7] A. Bondavalli et al., "Master failure detection protocol in internal synchronization environment," *IEEE Trans. on Instrum. and Meas.*, vol. 62, no. 1, pp. 4–12, 2013.

[8] Y. Kozakai and M. Kanda, "Keeping clock accuracy on a master clock failure in substation network," in *Proc. IEEE Int. Symp. on Precis. Clock Synchronization for Meas., Control and Commun.*, 2010, pp. 25–29.

[9] L. Haug et al., "Deterministic6g/deterministic6g_data: D4.1 - first detcom simulator framework release (datasets)," Dec. 2023. [Online]. Available: https://doi.org/10.5281/zenodo.10405085

[10] M. Gutiérrez et al., "Synchronization quality of IEEE 802.1AS in large-scale industrial automation networks," in *Proc. IEEE Real-Time and Embedded Technol. and Appl. Symp. (RTAS)*, 2017, pp. 273–282.

[11] M. Schüngel et al., "Advanced grandmaster selection method for converged wired and wireless networks," in *Proc. IEEE Int. Conf. on Ind. Technol. (ICIT)*, 2021, pp. 1007–1014.

[12] ——, "Analysis of time synchronization for converged wired and wireless networks," in *Proc. IEEE Int. Conf. on Emerg. Technol. and Factory Automation (ETFA)*, 2020, pp. 198–205.

[13] Y. Jeon, J. Lee, and S. Park, "An efficient method of reselecting grand master in IEEE 802.1AS," in *Proc. IEEE Asia-Pacific Conf. on Commun. (APCC2014)*, 2014, pp. 303–308.

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

[15] ——, "First Analysis of Time Synchronization for TSN Networks with Hot Standby GM," in *Proc. IEEE Int. Conf. on Factory Commun. Syst. (WFCS)*, 2025, pp. 1–4.

[16] 3GPP, "TS 38.331: NR; Radio Resource Control (RRC); Protocol Specification," Release-16, v16.1.0, 2020.

[17] ITU-T, "G.8275.1: Precision time protocol telecom profile for phase/time synchronization with full timing support from the network," 2022.

[18] 3GPP, "TS 22.104: Service requirements for cyber-physical control applications in vertical domains," Release-18, v18.0.0, 2021.

[19] ——, "TS 23.501: Technical specification group services and system aspects, system architecture for the 5G system (5GS)," Release-18, v18.0.0, 2022.

[20] IEEE, "IEEE Std 802.1ASdm-2024: IEEE standard for local and metropolitan area networks–timing and synchronization for time-sensitive applications amendment 3: Hot standby and clock drift error reduction," 2024.

[21] I. Godor et al., "A look inside 5G standards to support time synchronization for smart manufacturing," *IEEE Commun. Standards Mag.*, vol. 4, no. 3, pp. 14–21, 2020.

[22] 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, 2022.

[23] L. Mészáros, A. Varga, and M. Kirsche, *INET Framework*. Springer International Publishing, pp. 55–106. [Online]. Available: https://doi.org/10.1007/978-3-030-12842-5_2

[24] L. Haug and F. Dürr, “Deterministic6g/deterministic6g: Detcom simulator framework v2.0,” Mar. 2024. [Online]. Available: https://doi.org/10.5281/zenodo.10871314

[25] A. Varga and R. Hornig, “An overview of the OMNeT++ simulation environment,” in *Proc. ICST Int. Conf. on Simul. Tools and Techn. for Commun., Networks and Syst. & Workshops (Simutools)*, 2008, pp. 1–10.

[26] R. Exel and F. Ring, “Improved clock synchronization accuracy through optimized servo parametrization,” in *in Proc. Int. Symp. on Precis. Clock Synchronization for Meas., Control and Commun. (ISPCS)*. IEEE, 2013, pp. 65–70.

[27] X. Chen et al., “Frequency-tracking clock servo for time synchronization in networked motion control systems,” *IEEE Access*, vol. 5, pp. 11 606–11 614, 2017.

[28] R. Cochran, “GitHub - richardcochran/linuxptp: User space PTP stack for the GNU/Linux operating system. — github.com,” https://github.com/richardcochran/linuxptp, Accessed: Jan. 21, 2025.