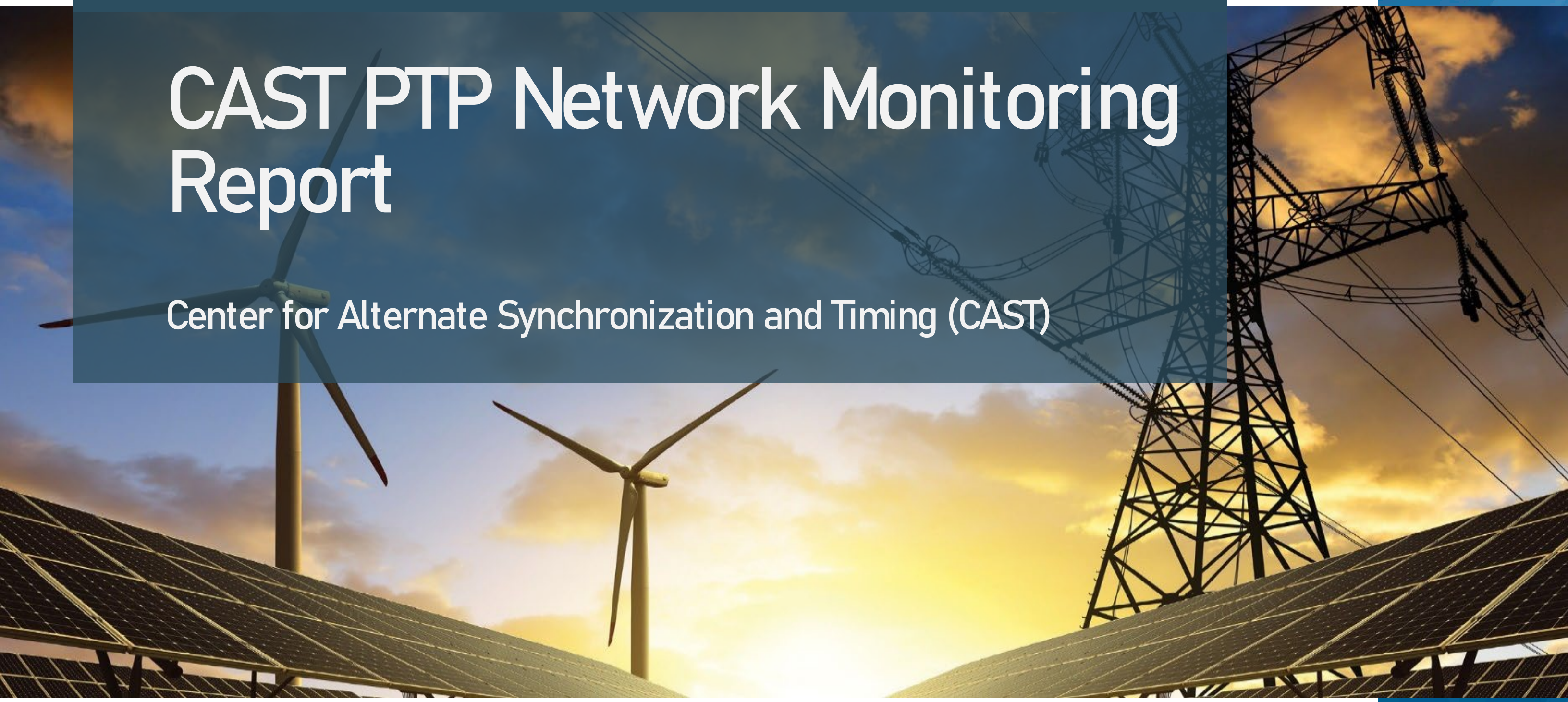
CAST PTP Network Monitoring Report

Center for Alternate Synchronization and Timing (CAST)



November 2025

Prepared by:
OAK RIDGE NATIONAL LABORATORY
and
U.S. DEPARTMENT OF ENERGY, OFFICE OF ELECTRICITY

ORNL IS MANAGED BY UT-BATTELLE LLC FOR THE US DEPARTMENT OF ENERGY

CONTENTS

LIST OF FIGURES

ABSTRACT

The Oak Ridge National Laboratory Center for Alternative Synchronization and Timing (CAST) performs research, development, testing, and evaluation of alternative terrestrial-based timing and synchronization infrastructure for the US power grid and other critical infrastructures. Alternative timing options reduce reliance on GPS and enhance the overall resilience of critical infrastructures. CAST infrastructure uses Precision Time Protocol (PTP) as the primary conduit for delivery of synchronization packets. CAST deploys PTP over long terrestrial links to synchronize a multitude of remote boundary clocks and downstream power grid components with the authoritative grand master clocks. Network traffic issues can severely degrade PTP accuracy. This report focuses on examining network traffic anomalies and their effects on PTP operation as well as the potential implications to CAST's high-precision remote synchronization operations.

1. BACKGROUND

The Center for Alternative Synchronization and Timing (CAST) team investigates ways to deliver time synchronization services over long terrestrial links to align grand master clocks (GMCs) with remote boundary clocks (BCs), which serve as the local timing authority for downstream power grid time-critical devices such as phasor measurement units. Specifically, phasor measurement units can report power delivery frequency at very high temporal resolution of more than 100 measurements per second. Timing skews in PTP packet irregularities caused by network traffic anomalies could induce out-of-sync power grid operation that leads to service interruption, breakdown, and other performance issues. [1]

There are multiple types of network traffic anomalies, mostly from non-malicious causes such as network architecture design issues, traffic congestion, or equipment failures. PTP traffic anomalies could also be the result of malicious causes such as network cyberattacks or insider attacks. Following are examples of network traffic anomalies that often result in packet delays or packet loss that interfere with PTP operation.

- Network traffic congestion: Unstable delayed traffic could lead to packet delay variation or jitters that affect PTP packet delivery timing.

- Network asymmetry: If the paths taken by PTP messages in opposite directions (master-to-slave [m–s] and slave-to-master [s–m]) are not symmetrical, then the PTP embedded delay calculations used for synchronization could become inaccurate due to different path delays. Different delays could be due to different network routing policies or internal switch delays.

- Symmetric stable delays: Although symmetric stable delays do not affect PTP internal offset calculations, they could skew time interval error (TIE), a periodic relative phase error measurement used to evaluate the long-term stability and accuracy of a clock signal.

- Packet loss: Congestion or errors on the network can lead to PTP packets being dropped altogether, causing synchronization issues and forcing PTP slaves to rely heavily on an internal clock or lose synchronization capability.

- Malicious network-based cyberattacks [2] such as distributed denial of service, man-in-the-middle, and bogus/impersonated GMC attacks: These attacks may overwhelm network resources, intercept or modify timing messages, or mislead slave clocks with false timing information; all can degrade synchronization accuracy or cause severe service interruption.

To understand how network traffic anomalies are linked to PTP operation monitoring, we first examine how the PTP-dependent timing devices (e.g. master-clock, slave-clock) calculate offset internally for synchronization using multiple transmission timing parameters. PTP devices calculate time offset by exchanging timestamped messages (in packets) between a master clock and a slave clock. Per the specification of PTP, the device first measures the round-trip path delays and processing delays. Then, it adjusts and uses these delay values in a calculation to derive the offsets between the clocks. The slave device then adjusts its clock to align with the master, typically with hardware-based timestamps for higher precision. As illustrated in Figure 1 [3],

1. The master sends a `Sync` message at time $t1$.
2. The slave receives the `Sync` message at $t2$.
3. The slave sends a `Delay_Req` message at $t3$.
4. The master receives the `Delay_Req` message at $t4$.

5. Thus, m–s transmission time is ($t2 - t1$), and s–m transmission time is ($t4 - t3$).
6. Calculate mean delay (or path delay) = (($t2 - t1$) + ($t4 - t3$))/2.
7. Calculate offset = ($t2 - t1$) − mean-delay = (($t2 - t1$) − ($t4 - t3$))/2.
8. Adjust slave with the offset to correct its internal time to synchronize with the master.

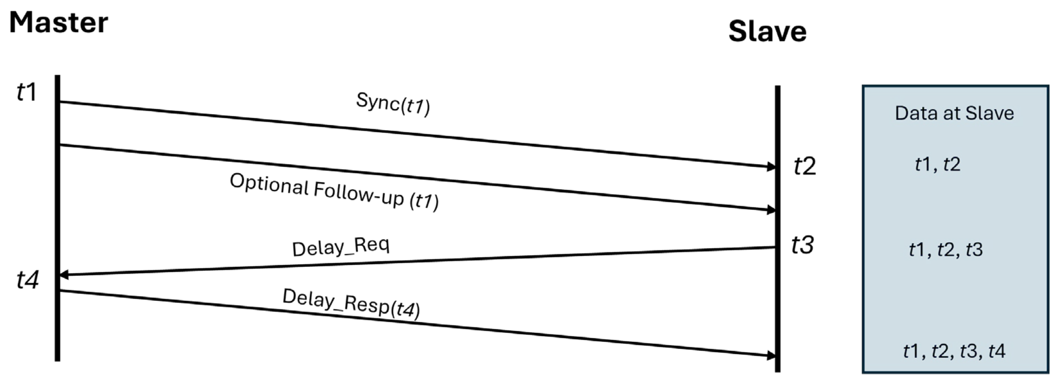


Figure 1: PTP synchronization messaging.

PTP offset calculation assumes network transmission symmetry. It averages out the bidirectional delays in Step 6 to obtain a mean delay, which is then used to calculate offset for slave clock synchronization in Step 7. Based on these calculations, we analyze how network anomalies in packet delay/loss disrupt relevant synchronization parameters. We then link them to the abnormal PTP clock observables that could affect CAST operation. For example, when transmission disruptions occur, they could affect `Sync` and `Delay_Req` message delivery speed differently on m–s and s–m, thus violating the symmetry assumption. Such disruption could potentially lead to inaccurate offset calculation and eventually synchronization failures.

Another PTP traffic-related CAST pertinent metric is TIE, a measurement of “wander”, or long-term variations in the phase of a clock signal. TIE is based on time error (TE) performance over time. TE is the instantaneous phase difference between the time on a PTP slave clock and the time on the PTP master clock. TIE plot is obtained by:

1. Using PTP, synchronize the slave clock to a master.
2. Measure at nth measurement TE(n) = t_slave(n) − t_reference(n), where reference is a high-precision clock, such as GPS or GPS-synced atomic clock.
3. Calculate TIE over k clock cycles, TIE(n, k) = TE($n + k$) − TE(n).

TIE plot could increase, decrease, fluctuate randomly, or drift over time. Parameters such as maximum time interval error (MTIE), denoting TIE plot peak-to-peak values, and time deviation (TDEV), for average time error over different averaging intervals, can also be derived. TE is affected by disruptions while PTP packets traverse through the network, and it can be captured by PTP-aware devices. Anomalies in network traffic disruptions thus contribute significantly to abnormal TIE behaviors.

2. NETWORK ANOMALY EXPERIMENTAL TESTBED

CAST presently maintains a PTP infrastructure prototype connecting Oak Ridge National Laboratory GMCs to Idaho National Laboratory BCs via combined private and public networks. On this platform, CAST conducts research, system development, testing, and evaluation tasks. Because this CAST network monitoring activity attempts to correlate traffic anomalies to PTP operation errors and their ramifications, we are required to manipulate the network to generate traffic anomalies for data collection. To prevent such traffic anomalies from disrupting regular CAST operation, we established an in-house testbed connecting one internal Master Clock (MC) (ADVA OSA 5422) to another in-house BC (also ADVA OSA 5422) through a Netropy network emulator, as illustrated in Figure 2.

This testbed is used to simulate network traffic anomalies with different attributes, allowing researchers to monitor and collect traffic data and PTP clock observables such as delays, offsets, and TIE. We then qualitatively analyze these data to correlate traffic anomalies to PTP internal parameters (e.g., mean delays, offsets) and their impacts to clock observables pertinent to CAST PTP operations. In parallel, we also evaluated one commercial off-the-shelf tool—Meinberg's PTP Track Hound [4]—exploring its PTP monitoring and analysis capability, particularly for the cases of PTP packet loss.

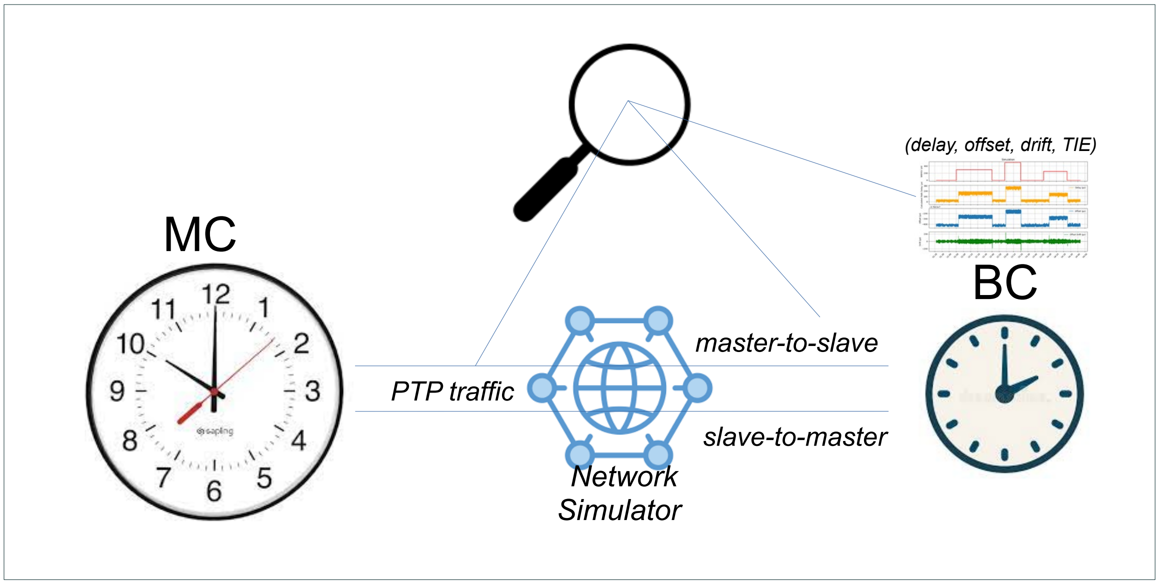


Figure 2: Network anomaly experimental testbed.

3. NETWORK ANOMALIES AND CORRELATION ANALYSIS OF THEIR IMPACT ON PTP

In this section, we examine multiple PTP network traffic anomalies and analyze their impacts to pertinent timing parameters. We divide PTP network traffic anomalies into two categories: packet delay and packet loss.

Packet delay can be caused by network architecture related issues or a combination of any of the following factors.

- Processing: Time taken for various networking devices (e.g., routers, switches, firewalls) to examine/process each PTP packet for the purpose of further routing.
- Transmission: Time to push the PTP packet into the physical link for transmission, pending on the packet size and link bandwidth.
- Queuing: Transmission delays from waiting in the device's routing queue, depending on device performance, link bandwidth, and traffic volume.

- Propagation: Time required to traverse through the physical link, depending on distance and media (e.g., optical vs. copper).

Packet loss can be attributed to different traffic conditions such as:

- Congestion: Excessive traffic volume overwhelms networking/routing constructs, resulting in PTP packets being dropped.

- Defective or outdated network components: Faulty or obsolete devices or cables, leading to PTP packets loss.

- Wireless instability: If wireless infrastructure is involved, then physical obstacles or signal reception quality challenges could cause the PTP packet to be discarded.

- Internet Service Provider (ISP) issues: If an ISP is involved, then network configuration or performance parameter adjustments such as quality of service or bandwidth throttling could lead to PTP packets being discarded.

Network anomalies depend heavily on the network architecture where the PTP is deployed. Simplified and direct networking connections with less traffic volume tend to lead to less network anomalies, thus higher PTP synchronization performance. Conversely, a more complex and congested networking environment likely contributes to less stable PTP operation.

PTP is a bidirectional protocol. It takes into consideration both m–s and s–m transmission delays to calculate clock synchronization offset. Asymmetric traffic delay between m–s and s–m presents a challenge. This difficulty stems from PTP assuming traffic flow symmetry and calculating offset required for synchronization by using equally divided m–s plus s–m delays. For this reason, we examined symmetric and asymmetric network anomaly cases separately. We further examined three levels of packet loss and analyzed how they affect PTP observables.

3.1 Symmetric Static Delay

Figure 3 illustrates our experiment of symmetric static traffic delay.

- We started with a stable and synced MC (master) to BC (slave) connection on the testbed.

- We then introduced a static traffic delay 300 µs (denoted as D) to both m–s and s–m PTP traffic, using the Netropy network emulator connecting them.

- Figure 3 top red line indicates the start and the stop time of the manually generated traffic delay.

- As expected, when the delay started, a stepped increase of the PTP calculated mean delay (or simply calculated delay) occurred (yellow line). PTP calculates mean delay as the average of the sum of m–s and s–m delays, which is $((t2 - t1) + (t4 - t3))/2$. Because delay D was added to both m–s and s–m, $(t2 - t1) + (t4 - t3)$ therefore equates to $2D$. As such, we saw the $2D/2 = D$ value yellow line step function.

- The $t1$, $t2$, $t3$, and $t4$ values were extracted from the capture PTP traffic packets' timestamp fields and were used to calculate and plot the subsequent delay, offset, and offset drift.

- We reasoned that the red (simulation) and yellow (calculated delay) graphs are correlated because they aligned in both start and end times, without other environmental interferences.

- PTP then calculated the clock synchronization offset by $(t2 - t1)$ − mean delay. Because $(t2 - t1)$ was increased by D manually, and mean delay is D, then $D - D = 0$. Consequently, the offset values are not affected by the symmetric network delay, as validated by the stable blue offset line. Essentially, the symmetric delays were canceled out.

- Slave does not need to adjust to sync with the master, which is correct, by PTP design.

- We then observed a relatively stable offset drift (defined as the change of the current offset with respect to the previous offset) line, in green, without spikes.

- However, because static network delay was indeed introduced to the (m–s) transmission, we observed a corresponding drop in the TIE graph at the bottom. This TIE graph was charted by a separate tool on a different time scale (UTC vs. local).

- This analysis suggests that symmetric and static delays largely do not affect PTP synchronization, except for the increased PTP internally calculated mean delay (but eventually canceled out during offset calculation) and the corresponding TIE step down.

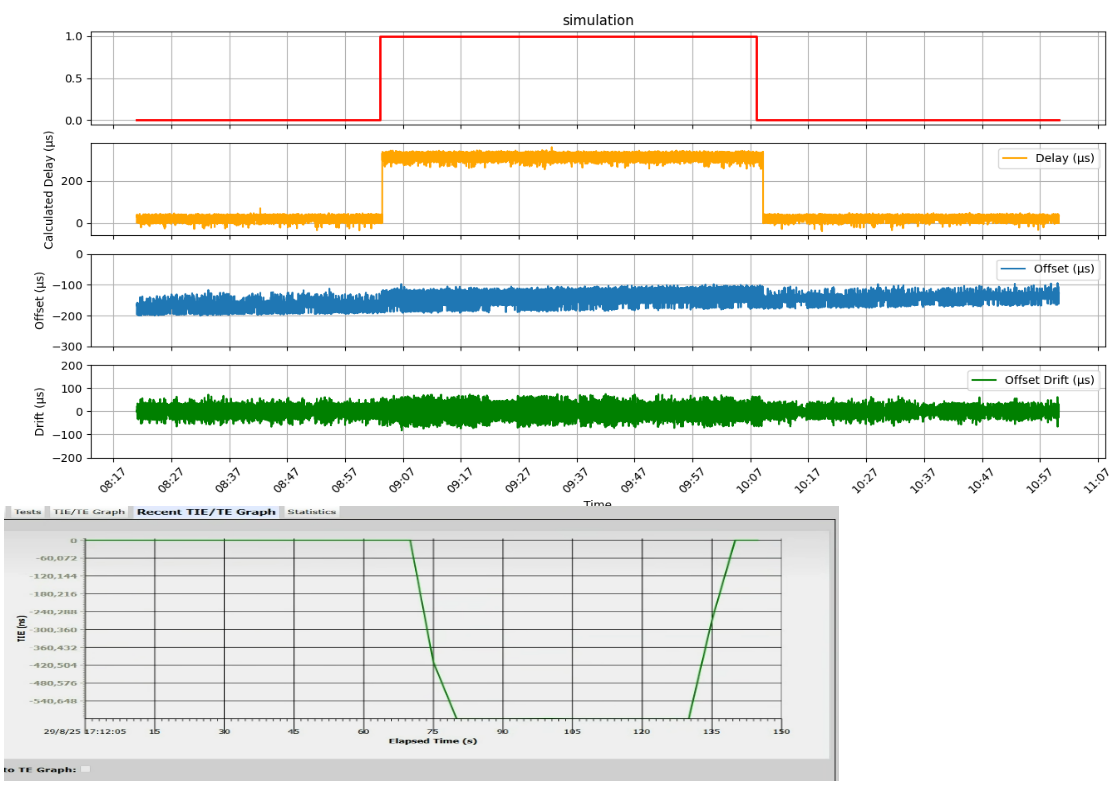


Figure 3: Symmetric Static Delay with associated TIE.

Figure 3 illustrates a simulated symmetric PTP traffic delay (in red step up function), triggering a corresponding calculated offset increase (in yellow step up), but did not affect the calculated delay (in blue) and drift (in green). However, it did result in a TIE decrease (bottom green step down).

3.2 Asymmetric Static Delay

An asymmetric static delay is problematic to PTP [5] because the protocol assumes network traffic symmetry. Figure 4 illustrates our observations.

- A static 600 µs asymmetric delay—selected arbitrarily yet sufficiently large to illustrate its impact on m–s PTP traffic—was introduced.

- PTP calculated mean delay (in yellow) to be $((t2 - t1) + (t4 - t3))/2 = (600 + 0)/2 = 300$, only half of the original 600 µs was added to the m–s traffic asymmetrically. The step increased in the packet timestamps-based calculation is in yellow.

- When PTP calculated the offset by $(t2 - t1)$ − mean delay, it became $600 - 300 = 300$. This resulted in an erroneous 300 µs offset, a problem for synchronization as the slave adjusts to sync for no other reason than asymmetric traffic delay. We visualized the resulting offset inaccuracy in the actual timestamp-based blue line step function.

- Correspondingly, the green line shows spikes in offset drifts when asymmetric static delays were applied and terminated.

- Offset drift is an important performance metric to monitor because it reflects how the offset between the master and slave clocks change over time. A constant offset is relatively straightforward to correct, whereas offset drift indicates a timing mismatch or network variability. In this experiment, the start and the stop of asymmetric delay causes sudden drift spikes/outliers.

- TIE changed from 0 to 600 µs in sync with the application of m–s PTP traffic delays with amplitude of 600 µs.

- We further validated the asymmetric static delay with three different experiments using different duration and amplitudes (described in Section 3.4). All three experiments confirmed our observation on the asymmetric delay and offset inaccuracy correlation.

- Our analysis shows that the timing and size (amplitude) of the artificially introduced asymmetric static delays are strongly linked to anomalies observed in PTP metrics—specifically mean delay, offset, drift, and TIE—largely because PTP assumes symmetric network delays, and when that assumption is violated (as in the experiment), errors occur. Although other factors might also cause PTP anomalies, the clear correlation between the injected asymmetry and the observed issues suggests that this asymmetry can serve as partial evidence when diagnosing such problems.

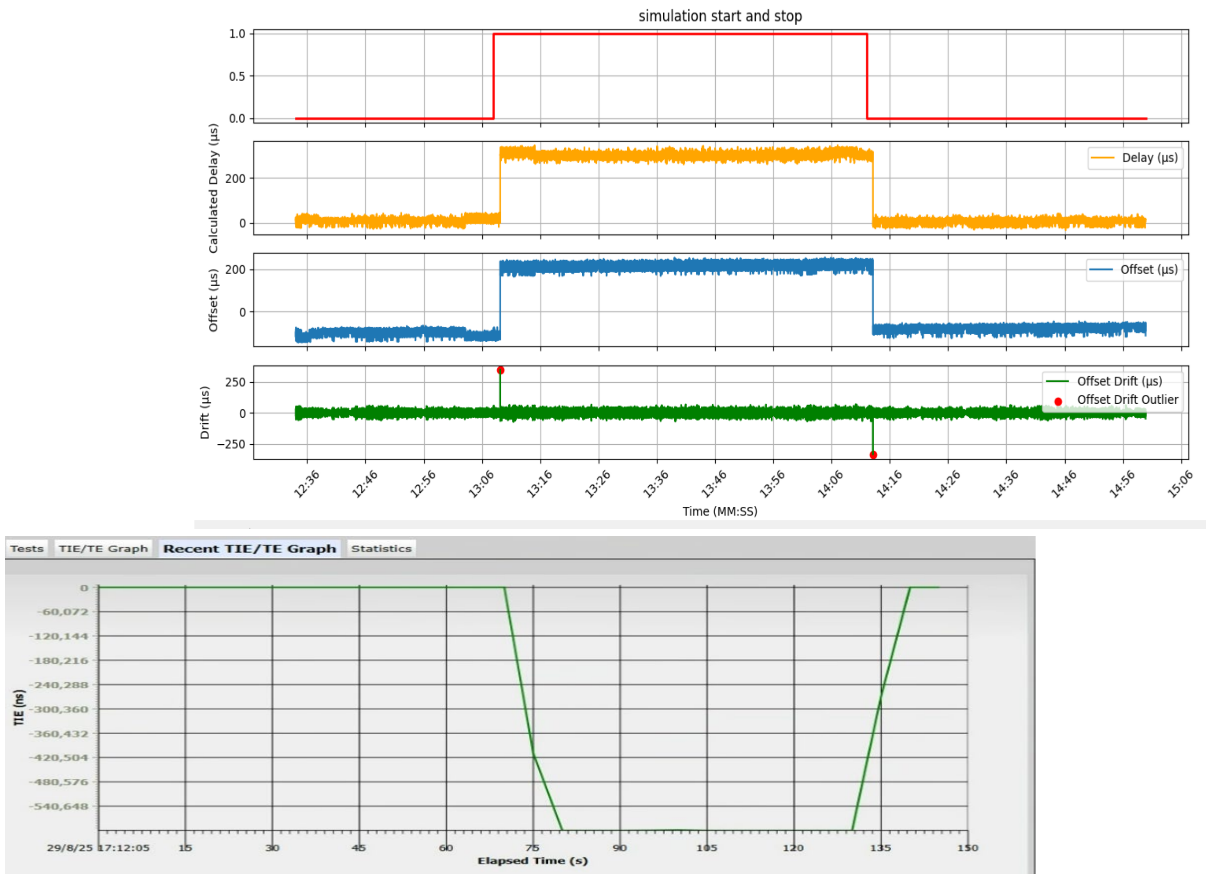


Figure 4: Asymmetric static delay with associated TIE.

Figure 4 above illustrates a simulated asymmetric PTP traffic delay (in red step up function), triggering a corresponding calculated offset increase (in yellow step up), also causing a calculated delay increase (in blue step up) and drifts (in green spikes). It also created a TIE decrease (bottom green step down).

3.3 Asymmetric Random Delay

Figure 5 illustrates the experiment with asymmetric random delays and their impacts.

- Asymmetric random delays were introduced into PTP m–s traffic, as illustrated in Figure 5.

- In this experiment, the red simulation line does not represent the random delay amplitude being applied. The network emulator inputs were the mean delay parameters of 600 µs with 150 µs deviation, and the red line step function here simply denotes the start and end time of the delay application.

- We observed corresponding TIE fluctuation during the same period.
- Because this simulated delay is random, we were unable to plot the corresponding mean delay, offsets, and drifts. The network emulator and the PTP parameter calculation/graphic we used originate from different software applications, and we did not allocate time to integrate them to pass the simulated random delay values to the PTP graphing software.
- Based on the prior observations on symmetric static delays and asymmetric static delays, we infer that corresponding mean delay, offsets, and drifts fluctuate accordingly and eventually result in the observed TIE diagram in Figure 5.

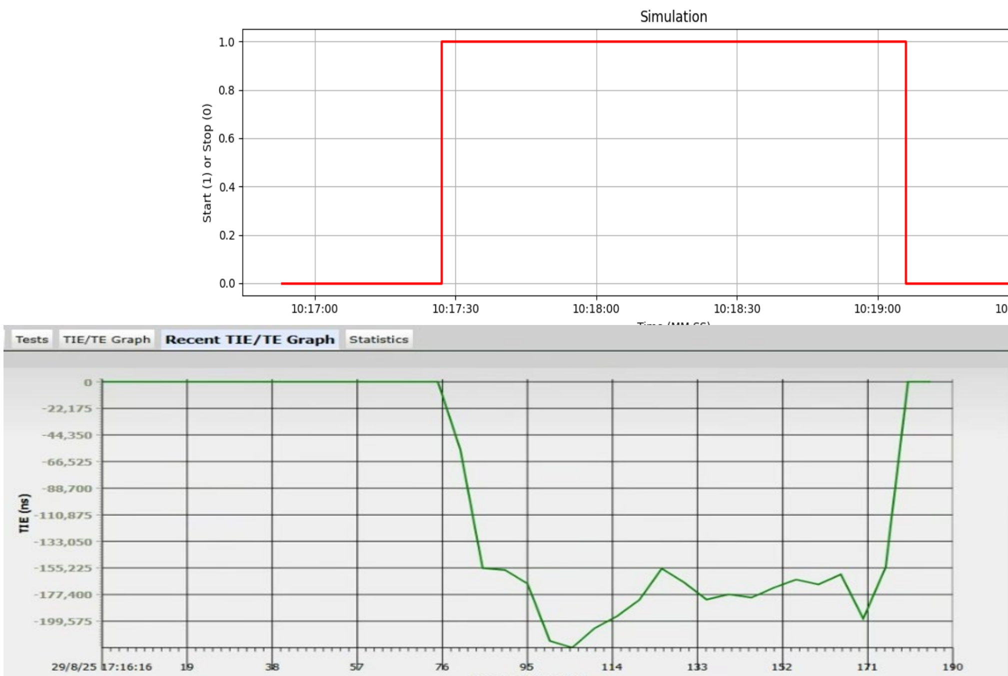


Figure 5: Asymmetric random delays.

Figure 5 above illustrates the start/stop time of a simulated asymmetric random PTP traffic delay (in red step up function), triggering a correspondingly random TIE decrease (green down line).

3.4 Asymmetric Jitters

Network jitters are common network behaviors could be caused by a wide variety of factors, including inconsistent delays from traffic congestion, wireless interference issues, or multitudes of faulty hardware or networking device misconfiguration. Jitters could essentially be characterized as collections of randomly occurred and varied duration and amplitude transmission delays.

- We simulated jitters in three static asymmetric delays with different duration and amplitudes.
- We simulated PTP m–s asymmetric jitters because two-way jitters incur complicated PTP mean delay calculations. At any given time, the instant combination effect of delays on m–s and s–m affects PTP offset calculations in a complex manner. If equal delays occur on both channels at the same time, then the jitter is equivalent to symmetric delay. Otherwise, the jitter is asymmetric and

could potentially induce inaccurate offset calculation, leading to synchronization errors. Phase and amplitude variances further complicate the calculation beyond meaningful analysis.

- Our asymmetric jitters experiment, shown in Figure 6, serves to validate the fundamentals of our observations on the effects of jitters.

- Observations on the PTP mean delays, offsets, drifts, and TIEs in this jitter simulation coincided with the results of the asymmetric static delay analysis. All three of the simulated delays were followed by the corresponding mean delay, offset, drift, and TIE results with correlated start/stop times and amplitudes.

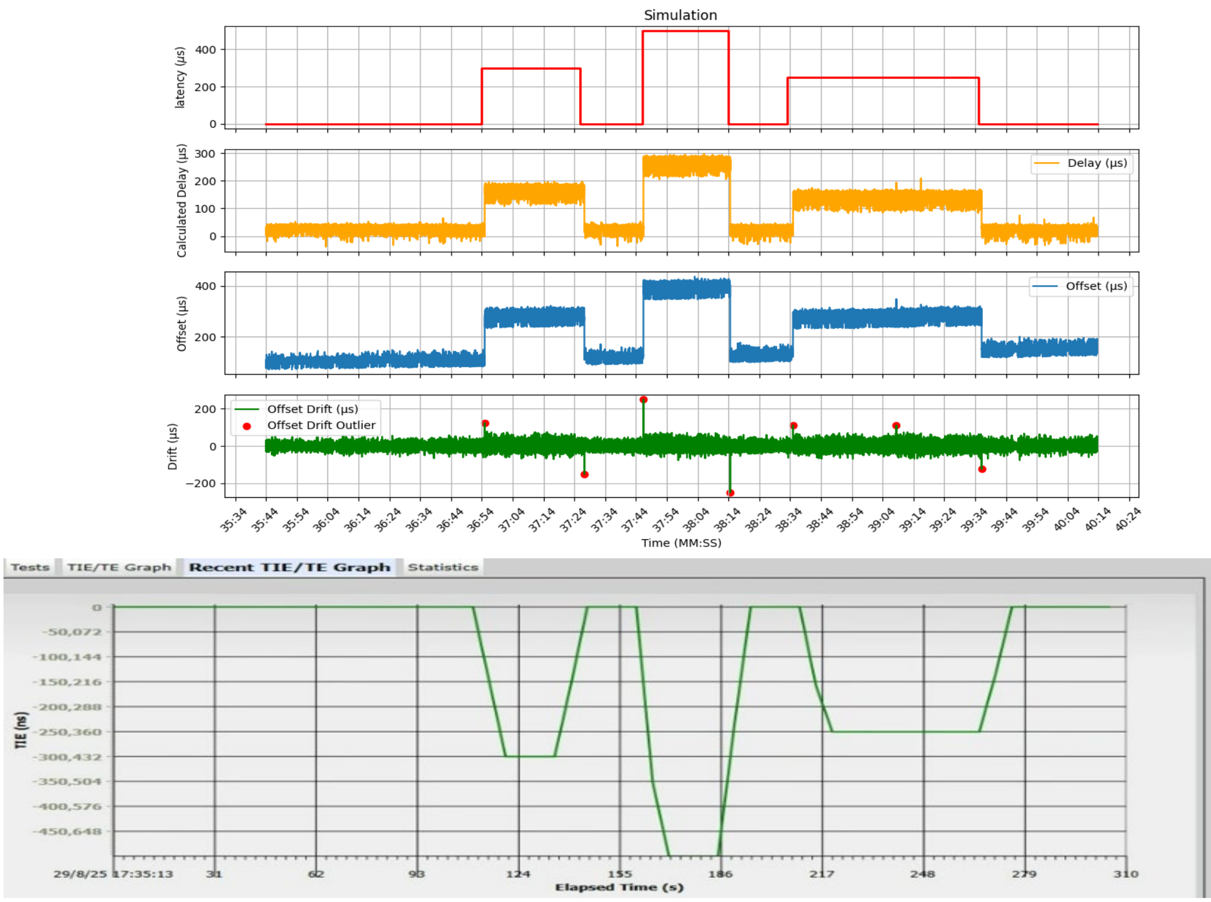


Figure 6: Asymmetric jitters.

Figure 6 above illustrates three generated random asymmetric PTP traffic delays, with different severity levels and duration (in red step ups), to simulate asymmetric jitter. This triggered three corresponding calculated offset increases (in yellow step ups), three corresponding calculated delay increases (in blue step ups) and six drifts (in green spikes), as well as three related TIE decreases (bottom green step downs).

3.5 PACKET LOSS

We examined packet loss to understand its effect on PTP synchronization. Packet loss could significantly disrupt PTP. Inaccurate delay measurements can be taken from the loss of `Sync`, `Delay_Req`, and `Delay_Resp` packets because they are needed for the timestamps to properly calculate $t1$, $t2$, $t3$, and $t4$

that contribute to the resulting offset and drift calculation. Furthermore, the offset adjustment directly affects slave clock synchronization. An inaccurate delay measure could degrade slave clock accuracy. It could also result in slave clock wander behavior, rendering downstream synchronization clients out of sync, which leads to overall system (e.g., power grid) operation errors.

PTP does not have a built-in packet loss recovery mechanism. However, the protocol is inherently resilient to a certain degree of packet loss. The protocol uses a filtering algorithm to maintain synchronization based on past timing history from prior packets. By design, PTP addresses packet delay variation, and the slave clock uses the algorithm to stabilize the fluctuations. When a packet loss occurs, the slave clock can rely on previous samples to maintain a stable time estimate and avoid over-correction due to inconsistent data. This process prevents sudden synchronization failure from packet loss.

However, under severe conditions (either by the time or by the amount of packet loss), the filtering algorithm’s hold over capability degrades. We explored symmetric packet loss scenario in three different severity levels (30%, 80%, and 95%) to understand PTP parameter behaviors.

3.5.1 Symmetric Packet Loss (30%)

At the Simulated 30% packet loss level, shown in Figure 7, we observed no `Sync` packet loss (because they originated from the master) but meaningful degradation on both `Delay_Req` and `Delay_Resp` transmissions. This degradation could have occurred because `Sync` was not received (and therefore not replied) or because of `Delay_Req` or `Delay_Resp` packet loss. The combined effect led to a TIE drop of about 700 ns during the 30% packet loss period, likely indicating PTP was effectively applying the filtering algorithm while sustaining synchronization.

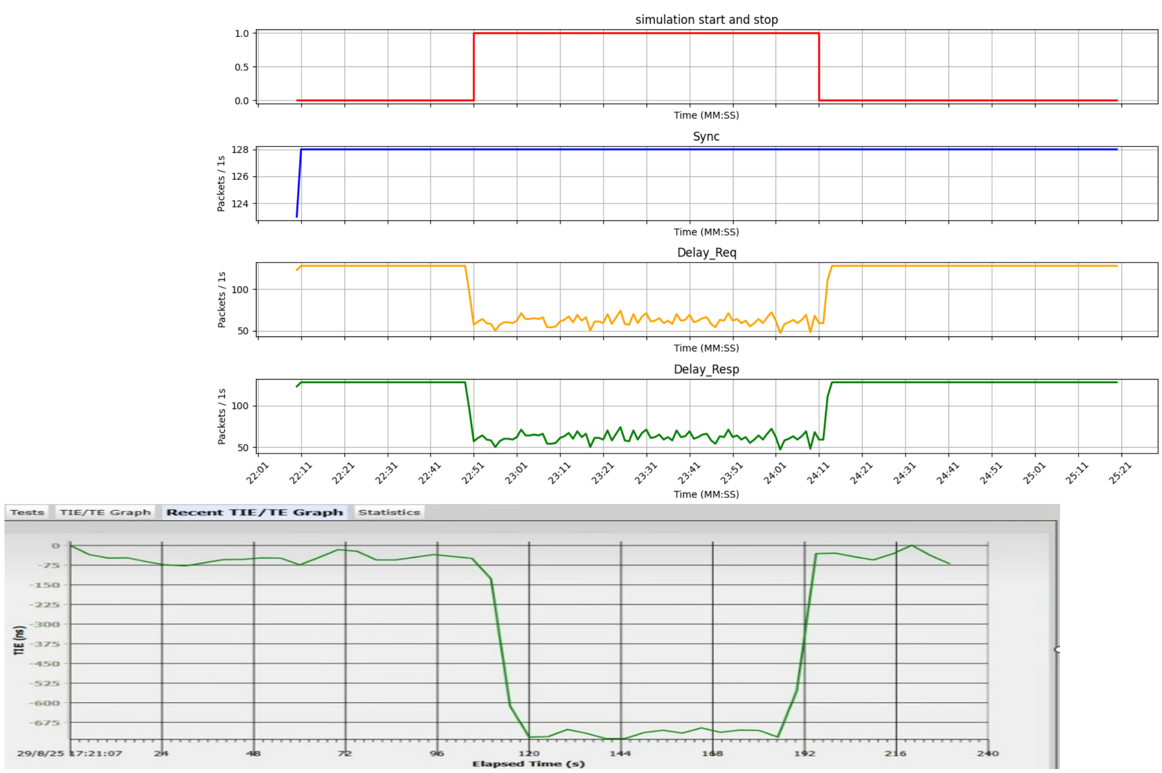


Figure 7: 30% packet loss and TIE impact.

Figure 7 above illustrates the start/stop time of an asymmetric 30% packet loss simulation (in red step up function). Packet loss was introduced by the network emulator, and it did not affect MC Sync packets (in blue). The 30% packet loss triggered a corresponding `Delay_Req` packet decrease (in yellow step down fluctuation), a `Delay_Resp` packet decrease (in blue step down fluctuation), and a TIE decrease (bottom green step down fluctuation).

3.5.2 Symmetric Packet Loss (80%)

At the 80% packet loss rate, the `Delay_Req` and `Delay_Resp` packet rates dropped to single digits per second (compared to 128 packets per second for `Sync` messages). This reduction occurs because the Netropy network emulator simulated 80% packet loss on both m-s and m-s paths. With averagely 20% `Sync` messages received at the slave and 20% of the responded `Delay_Req` received at the master, the observed `Delay_Req` and `Delay_Resp` packet rates dropped to roughly 4% (=20%*20%) and they fluctuated between 2-7 packets per second. The fluctuation is due to randomness in packet loss simulation. However, despite the significant packet rate reduction, the delay was still only 700 ns, which is not much different from that of 30% packet loss rate. That is, TIE dropped similarly during the period of 80% packet loss simulation. This similarity in TIE drop likely indicates that, as in the 30% loss case, PTP was still effectively applying the filtering algorithm to sustain synchronization.

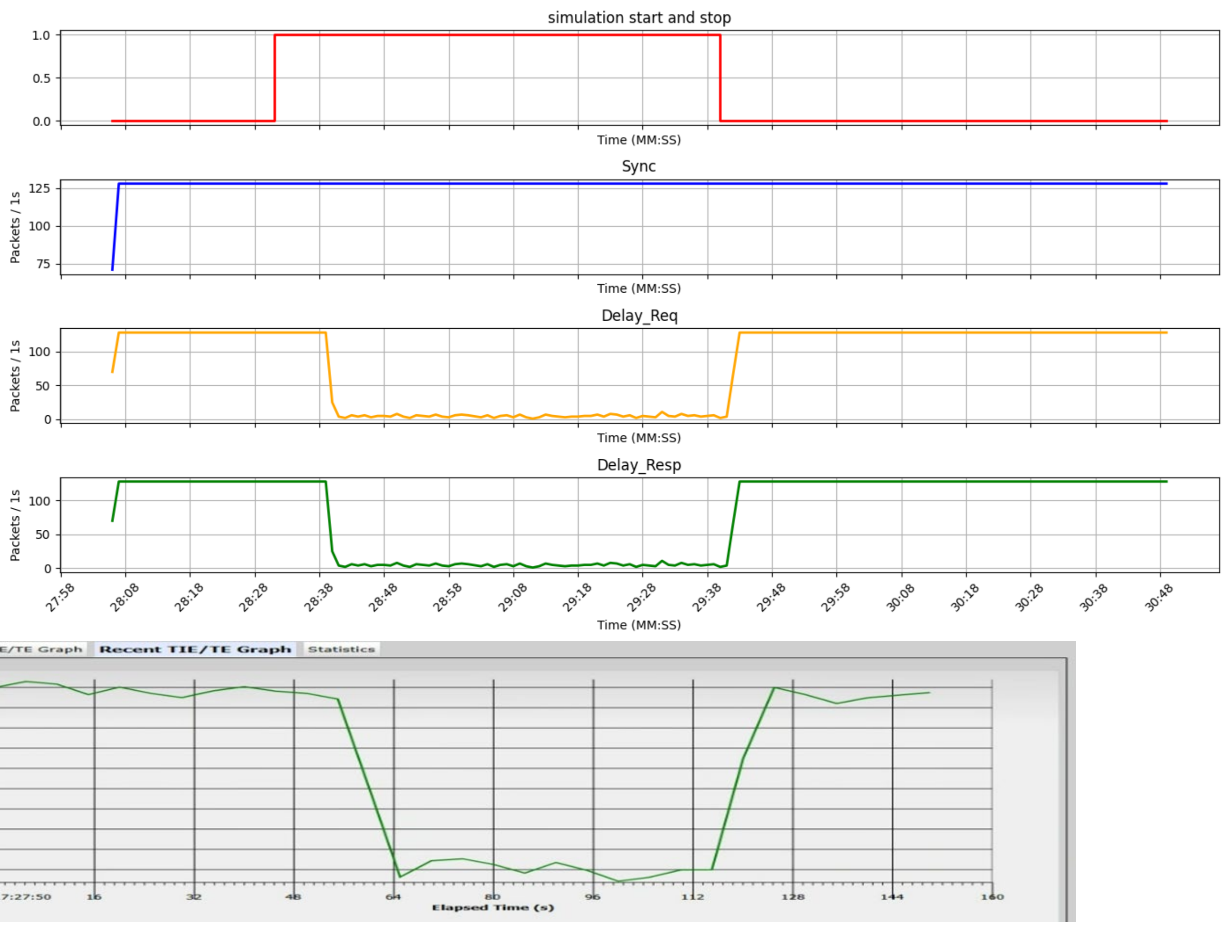


Figure 8: 80% packet loss and TIE impact.

Figure 8 above illustrates the start/stop time of simulated asymmetric 80% packet loss (in red step up function). Similarly to the 30% loss case, it did not affect MC Sync packets (in blue). The 80% loss triggered a corresponding `Delay_Req` packet decrease (in yellow step down almost to single digit per second), a corresponding `Delay_Resp` packet decrease (in blue step down almost to single digit), and a TIE decrease (bottom green step down fluctuation).

3.5.3 Symmetric Packet Loss (95%)

Approximately 10 seconds after the 95% packet loss simulation was applied, the red lines in the TIE plot in Figure 9 indicate a total shutdown of synchronization. Unlike the 30% and 80% cases, the loss of packets was beyond the failover recovery capability of the server. With 95% packet loss, the packets received were too sparse for the slave clock to accurately estimate offset/delay; therefore, the slave clock treated the delay as a loss of `Sync` signal and failed back to the “holdover” state.

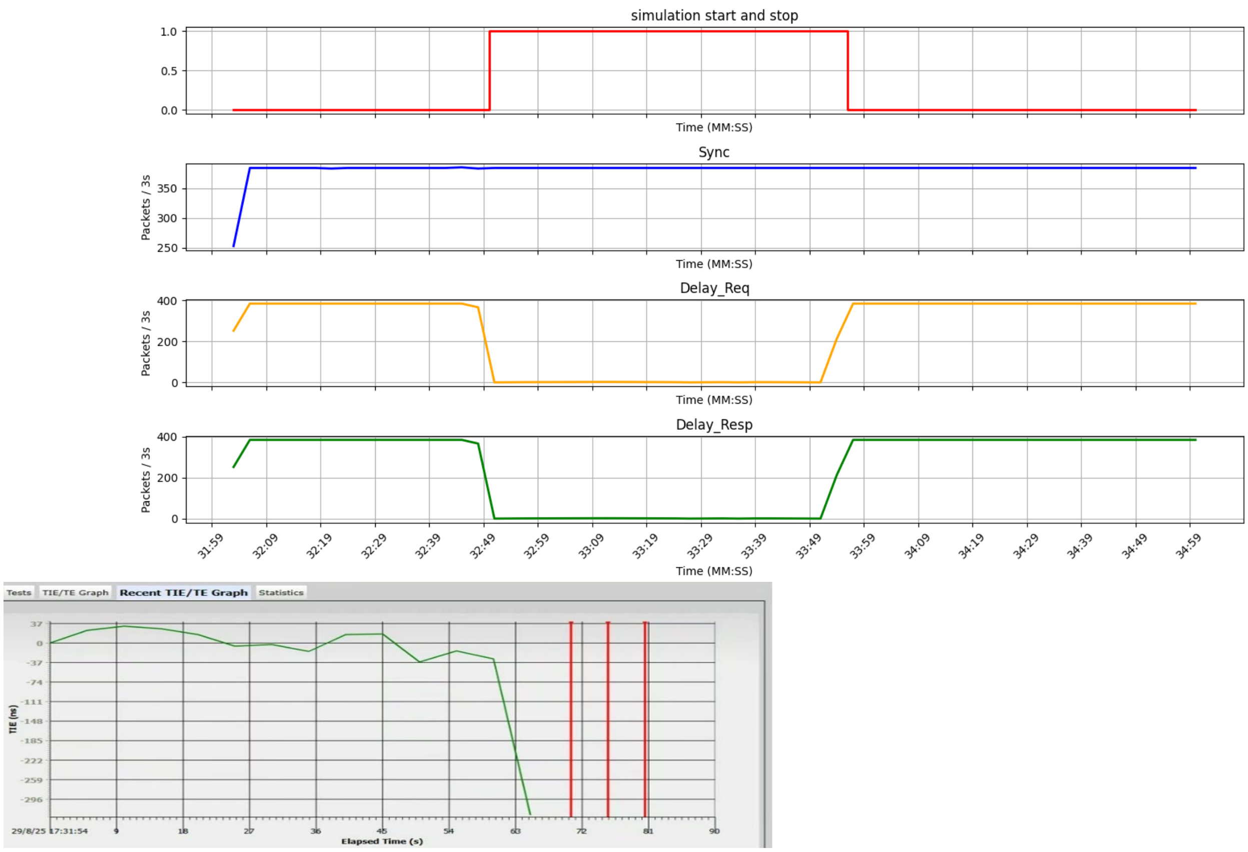


Figure 9: 95% packet loss and TIE impact.

Figure 9 above illustrates the start/stop time of simulated asymmetric 95% packet loss (in red step up function). Similar to the 30% case, MC Sync packets (in blue) were not affected. The 95% packet loss triggered a significant `Delay_Req` packet decrease (in yellow step down approaching zero), a corresponding `Delay_Resp` packet decrease (in blue step down approaching), and a total TIE dysfunction (bottom step down red bars).

4. PTP PACKET RATE MONITORING

In PTP, time-critical messages such as `Sync`, `Delay_Req`, and `Delay_Resp` play a central role in time synchronization. In addition to these, non-time-critical event messages—including `Announce`, `Signaling`, and optional `Management` messages—support clock selection, network management, and configuration.

Monitoring the transmission rates of both time-critical and non-time-critical PTP message types and comparing them to their configured values provides valuable insights into both network performance and clock behavior. Deviations in observed message rates may indicate issues such as packet loss, network congestion, clock misconfiguration, or even security threats such as impersonated GMC.

Figure 10 presents a screenshot of the PTP Track Hound tool under conditions simulating 80% packet loss. The top panel displays the average transmission rates for various PTP message types, and the bottom panel illustrates the traffic patterns of two selected message types. Pull-down menus allow users to choose the specific message types to view. In the bottom panel, the left figure plots the total packet rates over a short duration to highlight the onset of the packet loss simulation (i.e., when the rate dropped), as PTP track hound shows only the most recent 255 records. Meanwhile, the right graph depicts the packet rate of the `Announce` messages. Figure 11 presents the master clock configuration used in this test scenario.

Case 1: `Delay_Resp` rate drop indicates network congestion

In the top panel of Figure 10, the observed `Delay_Resp` message rate drops to 2.43 packets per second, far below the configured rate of 128 packets per second as shown in Figure 11. This significant reduction strongly suggests network-level issues, such as congestion, high packet loss, or queuing delays, affecting timely response delivery.

A key metric to monitor in such cases is the ratio between `Delay_Req` and `Delay_Resp` messages [6]. Under normal operation, each `Delay_Req` should elicit a corresponding `Delay_Resp`. Any sustained mismatch in this ratio may indicate dropped responses, delayed processing, or misrouted traffic and should trigger further investigation.

Case 2: `Announce` message rate validates Best Master Clock Algorithm stability

In PTP traffic, an `Announce` message is a PTP general message sent by a master clock to other clocks in a PTP domain. Its primary purpose is to inform other PTP devices about its own characteristics, such as its priority, quality, and accuracy, allowing the Best Master Clock Algorithm (BMCA) to determine the most suitable GMC and establish the PTP synchronization hierarchy.

As another PTP packet rate monitoring example, the `Announce` message rate is observed to be 8 messages per second, as shown in both the top panel and the bottom right graph of Figure 10. This value matches the configured transmission interval of the master clock, as shown in Figure 11.

Deviation of the observed `Announce` rate from the expected configuration may signify instability in the BMCA such as frequent re-elections or clock role changes. Alternatively, such a mismatch could indicate the presence of a rogue or impersonated GMC [7], which could disrupt time distribution and lead to synchronization errors. Such scenarios warrant immediate analysis and validation of the clock hierarchy.

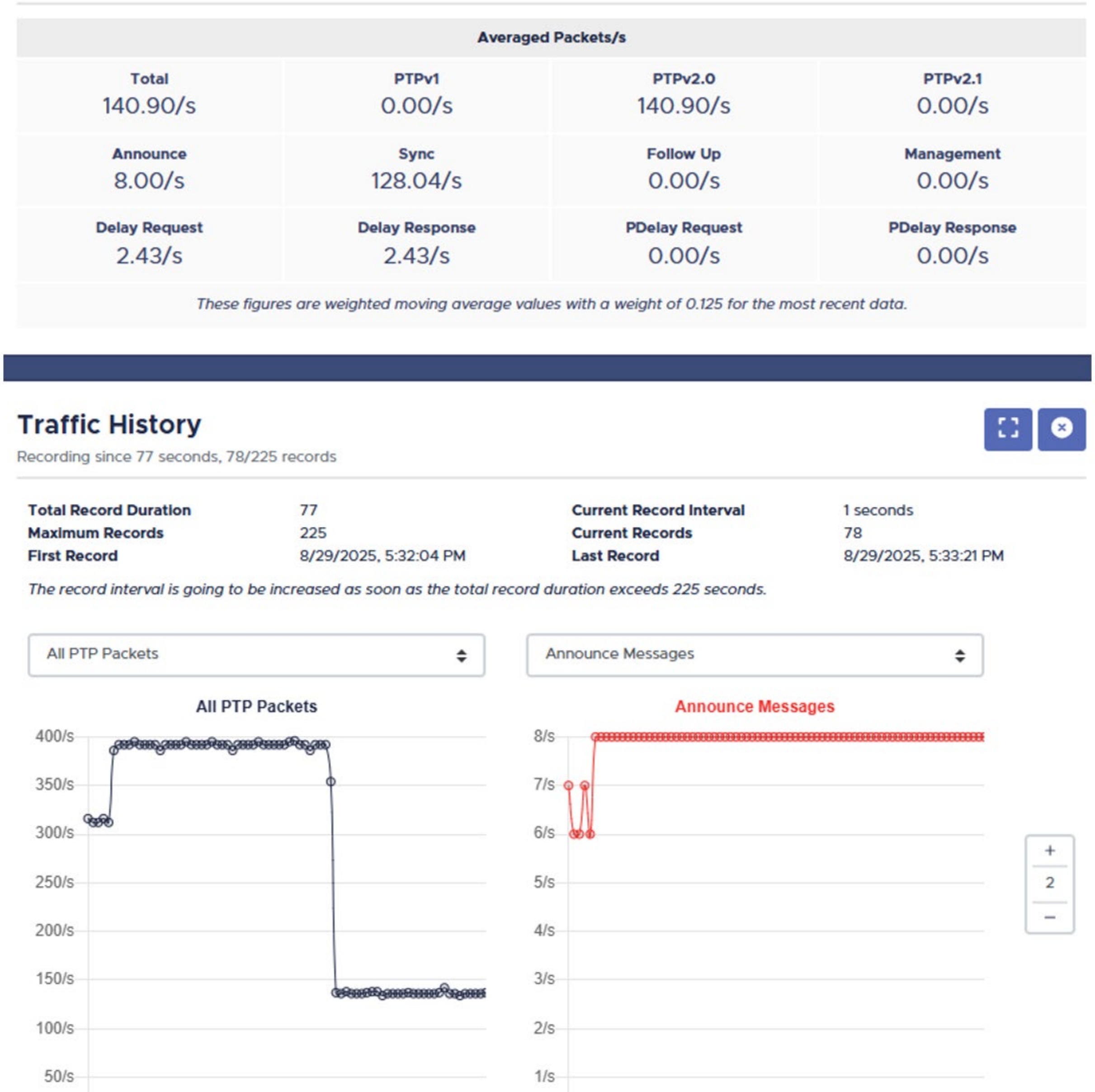
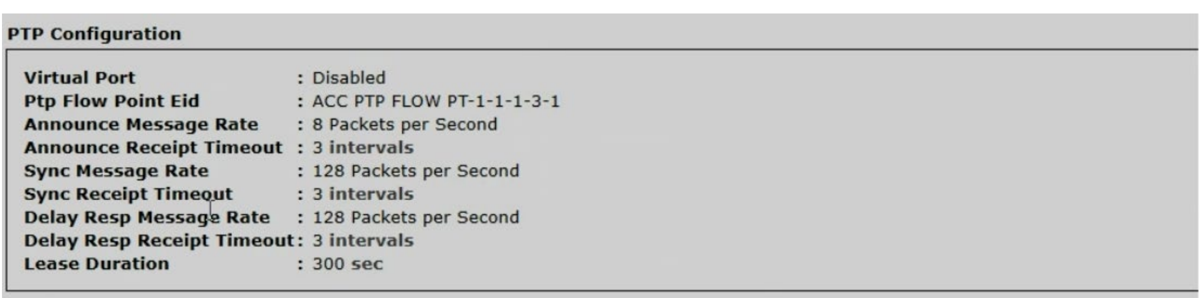


Figure 10: Meinberg PTP Track Hound screenshot.



PTP Configuration

Virtual Port	: Disabled
Ptp Flow Point Eid	: ACC PTP FLOW PT-1-1-1-3-1
Announce Message Rate	: 8 Packets per Second
Announce Receipt Timeout	: 3 intervals
Sync Message Rate	: 128 Packets per Second
Sync Receipt Timeout	: 3 intervals
Delay Resp Message Rate	: 128 Packets per Second
Delay Resp Receipt Timeout	: 3 intervals
Lease Duration	: 300 sec

Figure 11: PTP configuration at the MC.

5. CONCLUSION

In this report, we examined multiple PTP network traffic anomalies and correlated them to the PTP timing parameters calculation, and then to the impacts on CAST synchronization operation. We connected a MC to a BC through a network emulator. Using the emulator, we simulated controlled PTP traffic anomalies in both symmetric and asymmetric PTP traffic flows. We also simulated different severity levels of packet loss. Our experiment results and the analysis on them indicated that certain types of network anomalies such as increased asymmetric latency (delay), jitter, or bandwidth saturation (packet loss) could result in impaired PTP operation due to the inaccurate offset calculation errors. These errors could lead to reduced synchronization accuracy and potentially disable PTP or create operation errors on PTP-dependent networking devices, which could result in synchronization failures and interruptions to critical applications such as energy grids and industrial control systems.

In a related note pertinent to this investigation, to prevent network anomalies from disrupting PTP synchronization, a robust network design with PTP-aware devices is critical. Such designs could include employing PTP failover architectures with mechanisms like deploying redundant GMC, configuring dual-homed network connections using two separate paths, or using Parallel Redundancy Protocol (PRP) for PTP communication. Alternatively, configuring PTP-aware network devices such as switches and routers as BC or transparent clocks (TC) can help to minimize packet queuing or jitter. Dedicated PTP networks or VLANs could also improve PTP traffic quality, as well as applying Quality of Service (QoS) to prioritize PTP packets, or designing physical networks to be as symmetrical as possible. Finally, PTP operation coupled with network traffic monitoring is essential to maintain system integrity by proactively identifying and mitigating high-impact network anomaly causes. Consequently, PTP network anomaly observables identified in this report with the associated correlations, could serve as precursors for the investigative monitoring effort.

6. REFERENCES

[1] "Timing and Synchronization for Modern Power Grids", CAST Technical Bulletin #10, 2025. https://cast.ornl.gov/resources/

[2] W. Alghamdi, "An Analysis of Internal Attacks on PTP-based Time Synchronization Networks", Ph.D. Dissertation, National University of Ireland, 2021.

[3] "1588-2019 IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems", IEEE standard, https://ieeexplore.ieee.org/document/9120376

[4] "PTP Track Hound v2 – IEEE1588 Debugging &Monitoring", https://www.ptptrackhound.com/#/home

[5] David Spencer, "Asymmetry in PTP Networks - What It Is and What Do We Do About It", https://tf.nist.gov/seminars/WSTS/PDFs/7-4_Semtech_Dspencer_Asym_PTP_networks.pdf

[6] D. A. Popescu, et al, "Measuring Network Conditions in Data Centers Using the Precision Time Protocol", IEEE Transactions on Network and Service Management, Vo. 18, No. 3, September 2021

[7] T. Gunkel, "PTP Monitoring: A Must, Not An Option", Skyline blog, May 2022 https://skyline.be/skyline/news/ptp-monitoring-must-not-option#:~:text=PTP%20issues%20might%20originate%20from%20misconfiguration%20of,PTP%20slave%20devices%20becoming%20a%20PTP%20Grandmaster.