

Beyond USB Packet Arrival: Frame-Identified Time Transfer to macOS

Jonathan Yantis

Engineer

Research preprint, revised 15 October 2026

Final USB / user-level chrony experiment; not submitted

Abstract

A user-level clock can follow an embedded reference without steering the operating system clock or using USB packet arrival as a timestamp. We capture full-speed USB frame boundaries in hardware, identify their host-frame counterparts independently of clock fitting, and compose a causal counter-to-Mach map with an immutable source map and a retained UTC translation. The fine estimate feeds a private chronyd instance on macOS. In a completed 300.065 s experiment, qualified pre-ingest base-USB counter forecasts achieved 328.413 ns RMS across 9,437 observations. In the predeclared 60–300 s settled window, chrony’s reported estimator RMS ranged 23–339 ns (median 129 ns, 265 queries). A separate common-instant comparison of the delivered clock with its fine source estimate measured 414.351 ns RMS, with 182 valid and 54 withheld observations; its maximum absolute difference was 1.315 μ s. There were no USB request errors or actual daemon restarts. We explain the resolved repeated UTC-translation defect, the fit and authority contracts, and the remaining availability and calibration work. These results establish sub-microsecond settled relative RMS on one setup; shared-source agreement and an internal estimator do not independently establish absolute UTC accuracy.

1 Introduction

The useful timestamp is the time of an event, even when the packet describing it arrives later. This study brings an embedded BLE-derived clock to a macOS application over USB. TC-01 supplies time to the TC-04 receiver over BLE; the Mac receives only USB. A hardware-captured USB start-of-frame (SOF) event links the receiver’s counter to the Mac’s Mach clock. USB control transfers carry the captured evidence, rather than supplying the phase observations through their callback times.

The implementation now supplies an experimental fine UTC estimate to a private user-owned chronyd instance and reads back its virtual clock. The operating system clock is unchanged. The clock’s source is the BLE-derived USB refclock, identified as BLE; chronyd reports Stratum 1 when it selects that source. Stratum is a hierarchy indicator, not a statement of calibrated accuracy. The integration discovers compatible USB timing relays on any port; the present experiment tests TC-04 on one direct full-speed XHCI connection.

Three measurements answer different questions. A pre-ingest USB forecast tests the causal bridge using only preceding observations. Chrony’s reported tracking RMS describes its internal estimate. A simultaneous delivered-clock-minus-fine-estimate difference measures the application’s agreement with its own source. We report all three, their populations, and the availability denominator. The main result is a usable sub-microsecond settled RMS clock estimate with stable daemon operation, while startup, occasional withheld output and absolute endpoint bias remain explicit.

Earlier work on this setup found quiet short-window monotonic phase but millisecond variation in reported UTC. Code review traced repeated noise to recomputing an integer UTC translation from coarse observations independently of the fine monotonic map. Binding that translation to one coherent source interpretation removed

the repeated variation in the final capture. The original experiment remains valid historical evidence, rather than the description of the current result.

2 Related work and comparison criteria

2.1 gPTP, PTP and the Mac clock endpoint

IEEE 802.1AS specifies timing and synchronization for time-sensitive networks, using IEEE 1588 mechanisms where applicable. The IEEE working group’s current published list includes 802.1AS-2025; historical evaluations below use their stated earlier editions [2]. gPTP supplies a network timing architecture with synchronization messages, rate relationships and link-delay handling. The USB prototype in this paper does not implement its message set, peer-delay machinery, election or conformance requirements. The conceptual connection is the transport of event timestamps and the need to cross clock domains without treating software delivery delay as event phase.

A particularly relevant implementation is Clark’s *mac-gptp*. It exposes a Mac’s AVB gPTP virtual clock through the private TimeSync framework and can feed a chrony reference-clock socket. Its documented setup uses an AVB-capable Ethernet path and a Linux boundary clock. The README reports roughly 250 ns agreement between successive Mach/gPTP mapping estimates over a few seconds and explicitly retains uncertainty in a fixed adapter-to-Mach cross-timestamp offset [3]. This is closely related to our host-endpoint problem, despite the different transport and API.

The project’s refclock account reports 0.6 μ s RMS over half an hour with tuned chrony settings and approximately 2 μ s excursions during its fastest wander. It identifies oscillator wander by comparing the Mac mapping with a separate adapter-clock mapping and explains the importance of short frequency-estimation histories [4]. These are author-reported system-clock-to-gPTP residuals, not our measurements and not a calibrated UTC benchmark. We did not run that project, configure AVB, or steer the system clock. Its distinct adapter clock also provides an attribution experiment that our USB run lacks: our relative rate changes cannot yet be assigned uniquely to the Mac, USB reference, TC-04 or its upstream source.

Val et al. evaluate an FPGA-based Ethernet/wireless TSN clock-domain translation with an external Calnex measurement. Their Table III reports -4.3 ns mean and 18.7 ns standard deviation for a three-hop Ethernet setup, and -5.5 ns mean and 16.6 ns standard deviation for one 802.11 plus two Ethernet hops [5]. These are externally observed synchronization distributions in their implementation, not residuals after independently fitting short intervals. The numerical similarity of any short-window figure in our study would not establish an equivalent result.

NIST’s calibration study illustrates the missing metrology more directly. Technical Note 2030 reports a 36 ns maximum offset through two transparent clocks with a 6 ns one-sigma uncertainty, and separately discusses source calibration and a power-related anomaly that increased receiver-calibration uncertainty to 100 ns [6]. This is evidence for the importance of reference calibration, not a performance baseline that our experiment can claim to meet.

2.2 USB synchronization and causal delay bounds

Lim, Rosdi and Yap use USB 3.0 Isochronous Timestamp Packets to coordinate device triggering. Their publisher abstract reports approximately 200 ns trigger delay and operation through up to three consecutive hubs [7]. The endpoint is relative triggering among SuperSpeed devices, rather than a full-speed device-to-Mach clock association. It is nevertheless clear prior evidence that USB timing can be much finer than a software transaction interval. Patent literature also describes recovering and synthesizing device clocks from periodic USB structures [8]; we do not present SOF-based synchronization as a new general principle.

WALT’s primary clock-synchronization notes develop causal upper and lower clock-offset bounds from messages in both directions. They report interval widths around $100\ \mu\text{s}$ on one Linux USB 2 path and $800\ \mu\text{s}$ on a USB 3 path [9]. Those are bound widths on named configurations, not standard deviations or physical accuracy floors. We use a related causal principle for *integer identity*, while hardware boundary observations supply the fine phase evidence. Our periodic UTC request intervals deliberately remain wide rather than pretending that a request midpoint is an exact event time.

NTP is a separate network clock-discipline system. RFC 5905 discusses capabilities extending to tens of microseconds on suitable workstations and fast LANs [10]. The Internet pool checks in our experiment do not test that regime. A noisy network observation is not evidence that the underlying NTP-disciplined clock is noisy by the same amount.

3 Timing architecture and coordinate contracts

Let C be the receiver’s free-running counter, $M_g(C)$ the source-monotonic projection for immutable map generationg, and O its retained integer UTC translation. The source coordinate is

$$U(C) = M_g(C) + O. \quad (1)$$

Let A denote Mach absolute time, K Mach continuous time, and W local macOS wall time. These coordinates have separate meanings and failure boundaries. The USB bridge estimates $A = F(C)$ from hardware SOFs. To evaluate source time at a current host event, the client uses the inverse causal map $C = F^{-1}(A)$ and applies the exact eligible M_g and O . A bracketed association relates A to K ; local W is used to deliver the SOCK sample and interpret the daemon’s clock, not to train the SOF fit.

3.1 Hardware capture and transport

TC-04 uses an nRF52840 TIMER3 at 16 MHz (62.5 ns ticks). A USB D SOF event triggers PPI capture into CC3, so a delayed userspace callback cannot move the original hardware timestamp. Nordic documents a synchronization stage before PPI; timer resolution does not calibrate that stage or the physical USB path[11]. Capture records preserve the timer, frame ordinal, scope and epoch. Acceptance requires coherent frame sampling, strictly advancing sequence/counter, a consistent frame increment and capture age below 16,000 ticks (one full-speed frame).

Vendor EP0 requests retrieve precision SOF/source records at 32 Hz and periodic source information at 1 Hz. The tested interface is FF/54/01, not a standard USB time class. UTS1 kind 1 uses request 0x53; kind 2 uses 0x52. Marker OUT 0x54 and immutable fetch IN 0x55 carry a nonce-bound 64-byte marker tuple. The helper uses the public IOUSBHost APIs, including `referenceMicroframeWithTime:error::`; runtime Mach conversion is 125/3 ns per tick on this setup[12, 13]. The 1 ms frame cadence and $125\ \mu\text{s}$ microframe numbering are event spacings, not accuracy floors for timestamped edges.

3.2 Identify the frame before fitting phase

A smooth affine fit cannot distinguish two host/device associations separated by a whole frame. The implementation therefore constrains the integer host-microframe origin using causal zero-data OUT marker transactions and separately recorded host frame-layout checks. Request and completion brackets bound when the marker could have been observed; the marker fetch retrieves the immutable captured tuple later. On this direct full-speed XHCI topology, the intersection of these bounds identifies one integer origin. A $1\ \mu\text{s}$ model allowance is part of this identification procedure; it is not an independently calibrated total uncertainty bound.

The bridge requires at least eight valid layout checks and sixteen causal markers, verifies the omission of the three microframe bits in the full-frame API, and retains expiry and consistency fences. A frame-origin change, boot change, USB/capture epoch change or invalid host association invalidates the affected model. Matching this identity is distinct from estimating fine phase. Sleep/resume, hubs and alternate controllers have not been qualified by this experiment.

3.3 Fit a clock using its past

The shipping bridge retains at most 1,024 accepted SOF tuples, approximately 32 s at the requested rate. A causal forecast requires at least 64 prior points. For a tuple (C_i, A_i) , the prior window gives F_{i-1} ; the counter-to-host innovation is evaluated before the tuple is ingested. Only then may the model be refitted. The qualified inverse-counter forecast population additionally applies the existing hardware and eligibility checks. Its signed error is expressed in nanoseconds; it is not a retrospective fitted residual.

For signed error e_i , we report

$$\bar{e} = \frac{1}{N} \sum_i e_i, \quad \text{RMS} = \sqrt{\frac{1}{N} \sum_i e_i^2}, \quad \sigma = \sqrt{\frac{1}{N} \sum_i (e_i - \bar{e})^2}. \quad (2)$$

RMS keeps bias. A whole-run detrend, a last-window residual, a forward innovation and an inverse forecast are different populations. In particular, the final 1,024-point fitted residual is not substituted for full-capture prediction performance. No request midpoint, local-wall residual or future tuple trains the fine bridge.

3.4 Source permission remains original

Each usable source record carries the exact map, full root/boot/source/path scope and original permission deadline. The client evaluates whether the captured or projected counter lies in that map's interval and clips host use to the original deadline. A later map cannot retroactively authorize an unavailable event. Source restart, interpretation change, invalid UTC, malformed input, unknown classifications and original expiry retain their withdrawal fences.

The .53 correction distinguishes missing historical-SOF geometry from current source withdrawal. Under the existing publication lock, a reason 7/USM1 classification is emitted only when projecting the current counter under the same original source map succeeds. Its flags and captured UTC remain zero: it is an invalid time sample. The host withholds output and cancels pending sends while retaining private filter training only within the old grant's unchanged deadline and exact provenance. This observation cannot extend permission or install a new UTC map. True withdrawal and expiry still discard the relevant state. No GM behavior or authority deadline was changed by this RC/host correction.

4 User-level chorny integration

The normal CLI starts one owned private chorny 4.9 instance with `-x -U -u CURRENT_USER`, a private command socket and a SOCK refclock. The documented `-x` mode disables system-clock adjustment[15]. The local manager uses a contained known configuration, waits for both sockets, and reaps only its own child. No administrator service, system timing configuration or direct Mac Bluetooth receiver is involved. Compatible relay discovery spans USB ports; identity checks and exclusive ownership prevent accidental mixed-device records.

The default fine feed composes the causal SOF/Mach bridge, the immutable source projection and the retained UTC translation at a common host instant. Integer UTC differences avoid epoch-sized floating-point subtraction. Each admitted fine observation retains a wall/Mach bracket no wider than $2\mu\text{s}$, its original grant deadline, source

generation and packet provenance. This admission limit does not become a $2\mu\text{s}$ absolute accuracy claim. A generation fence cancels pending feeds on authority transitions.

The application monitors chrony’s tracking correction and rate to project its user-level clock in local wall coordinates, subject to Mach freshness and wall-step guards. The recorded SOCK configuration uses `poll 0 filter 4 precision 0.05 delay 0.2`, with `minsamples 4` and `maxsamples 64`. The 50 ms precision and 200 ms delay are provisional configured allowances, not measured uncertainty bounds or tracking RMS. The SOCK refclock has stratum 0; the observed selected virtual clock reports stratum 1. No forced local stratum, fabricated PPS, trust override or network fallback creates a successful selection. Tracking RMS is the daemon’s historical offset statistic[15]; the UI labels it CHRONY EST RMS and uses whole nanoseconds below $1\mu\text{s}$.

Presentation and authority remain separate. For a display glitch strictly shorter than 500 ms, the large clock/RMS hold the last visible values with an immediate loss indicator; at 500 ms the outage presentation applies. Held pixels never feed chrony or serve clock APIs. The chrony page exposes source, frequency, freshness, filter/feed and lifecycle statistics. A smooth screen is not evidence of continuous time availability.

5 Experiment and predeclared analysis

The final trial ran for 300.065 s using firmware 0.15.53 (commit 1a16110a662ae66109b21bafa9958baf48d0465a) and the frozen CLI 0.21 package. The operator date is 15 October 2026; capture launch was 16 October 2026 at 04:08:13 UTC. Hardware is one Apple silicon Mac, macOS 26.7 build 25G229, and one direct full-speed XHCI connection to TC-04 following TC-01. Selected source scope remained constant. RF/GNSS placement, temperature and load were not controlled treatment variables.

Before capture, the test declared full $0 \leq t < 300\text{s}$ and settled $60 \leq t < 300\text{s}$ windows. The settled boundary is not selected from the observed convergence trace. New tracking queries and fresh selected status estimates are counted separately; zero-update intervals are not selectively removed. Approximately 1 Hz status samples supply explicit valid/withheld denominators. Missing observations are not filled and their counts do not imply measured continuous outage durations.

The direct comparison computes delivered chrony UTC minus the fine estimate at the same admitted host wall/Mach instant, using integer differences. This is a shared-source relative comparison: source or endpoint error common to both clocks may cancel. It is independent of chrony’s reported RMS statistic, but it is not an independent UTC reference. The trial disabled external NTP queries and made no operating-system clock changes.

Exact reviewed app-only installation preceded the run: actual new-application bytes, all 672 protected blocks and supported identical-UF2 unattended return were verified. The fixed helper budget was 9,600 samples. It completed at 299.990533 s; the following USB-session fence and daemon epoch end were planned finite cleanup, with no successor daemon. All samples in the fixed windows, including the final cleanup milliseconds, remain retained. This is not an unplanned source outage or an actual restart. Owned processes, descriptors, sockets, writers and locks were closed after the run.

6 Results

6.1 Current base USB prediction

Table1 gives the final capture’s base bridge results. The complete shipping replay runs forward over the unchanged raw capture. This recovers the final three hardware tuples beyond the last live status snapshot, without retroactively training any forecast. The native end record resets the bridge during finite cleanup; its zero counters are not the capture’s performance summary.

Table 1: Base USB bridge, complete final capture. Forecasts use preceding data.

Measurement	Result
Hardware tuples: accepted / rejected	9,558 /42
Qualified pre-ingest inverse forecasts	9,437
Inverse forecast RMS / population σ	328.413 /314.672 ns
Inverse forecast mean / minimum / maximum	-94.003/ - 897.863/ + 722.899ns
Separate forward innovations: N / RMS	9,494 /327.448 ns
Forward innovation population σ	313.840 ns
Final 1,024-point fitted residual σ (diagnostic)	103.409 ns
Whole-run detrended counter σ / relative drift	4.527 μ s /+5.602ppm
Host frame-layout checks / failures	1,200 /0
Accepted / rejected causal markers	1,198 /2

Of 9,600 raw hardware tuples, 9,558 were accepted and 42 rejected. Qualified inverse forecasts numbered 9,437; 24 were rejected by hardware validation and 139 were skipped or unqualified. Their RMS was 328.413 ns with bias retained. Separate forward phase innovations gave 327.448 ns RMS. These two similar values are not pooled. The whole-run 4.527 μ s detrended counter spread shows the slower rate wander that a local causal fit follows. The final 103.409 ns rolling fitted residual describes only its current window.

All 9,866 valid source records carried one identical UTC translation; its observed range was zero. Source-map mismatch count was zero. The removed repeated translation noise is therefore absent in the current observed scope. A constant translation can still carry an unknown initial bias.

6.2 Chrony estimator and delivered clock

Table 2 separates chrony's reported estimator from the application's direct comparison. All 265 new tracking queries in the settled window were below 1 μ s RMS. Their median was 129 ns, 95th percentile 263 ns by nearest rank, and last value 98 ns. The separate fresh selected status population numbered 182 and had median 128.5 ns and 95th percentile 278 ns. Repeated observations of a filter statistic are not independent accuracy trials.

Table 2: Predeclared settled window, $60 \leq t < 300$ s.

Statistic / population	Result
New tracking queries	265
Reported tracking RMS: minimum / median / maximum	23 /129 /339 ns
Reported tracking RMS: P95 / last	263 /98 ns
Direct chrony-minus-fine comparison: valid / withheld	182 /54 of 236
Direct comparison RMS / population σ	414.351 /413.841 ns
Direct comparison signed bias / maximum absolute	-20.549ns /1.315 μ s
Source-valid / withheld status observations	235 /1 of 236
USB request errors / actual daemon restarts	0 /0

The delivered-clock comparison achieved 414.351 ns RMS across 182 valid points, with 54 withheld of 236 status observations. Its maximum was 1.315 μ s: sub-microsecond RMS does not mean every difference was below 1 μ s. The full 0–300 s population had 220 valid comparisons and 76 withheld of 296, with 399.373 ns RMS. Chrony's initial tracking RMS reached 32.072201 ms; that startup transient is retained in Figure 1 and is not folded into a claim that all readings were sub-microsecond.

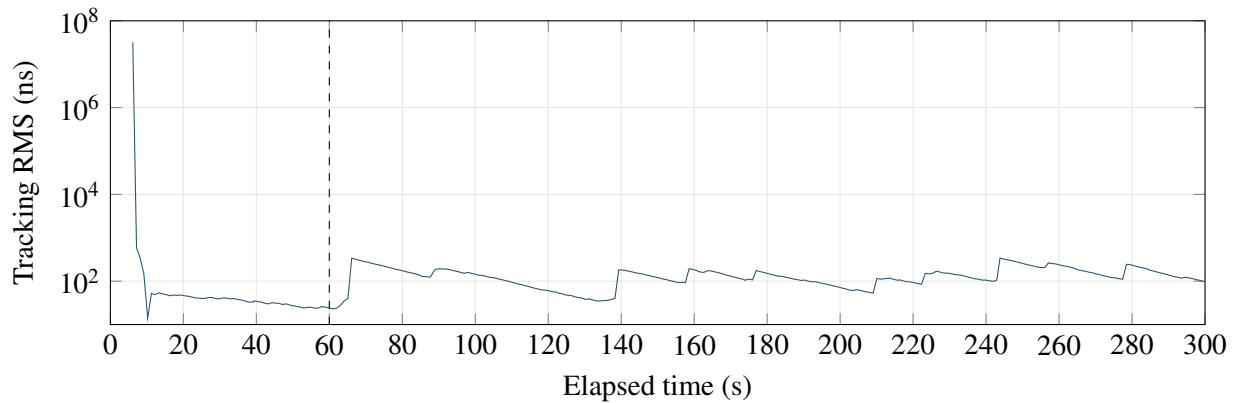


Figure 1: All positive new tracking-RMS queries in the full window. Logarithmic ordinate retains the 32.072 ms startup transient. Dashed line is the predeclared 60 s settled boundary; no transient exclusion changes that boundary.

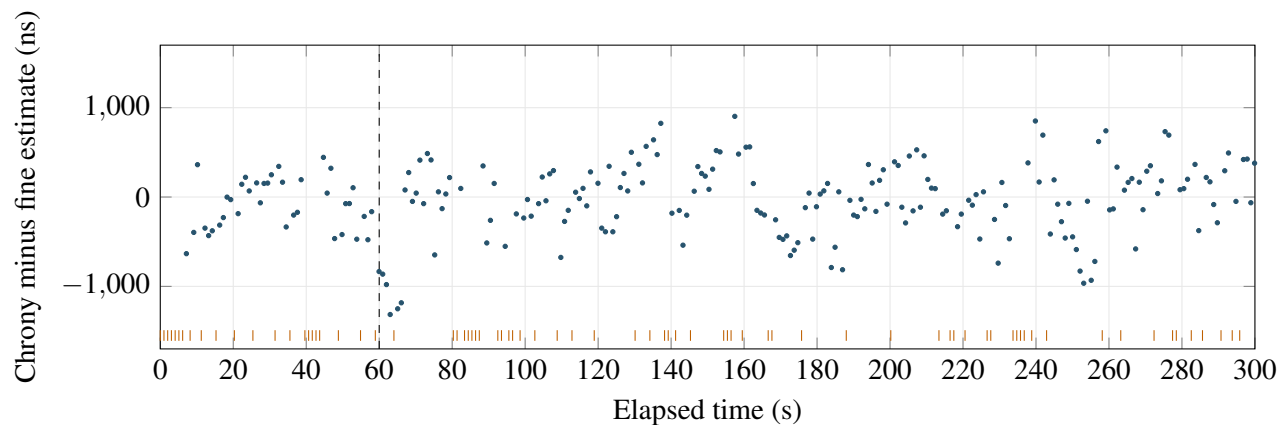


Figure 2: Every valid full-window shared-source comparison, with withheld status observations shown as orange ticks at -1550 ns (their vertical position is a plotting key, not an error value). No line interpolates missing observations. Settled RMS is 414.351 ns, and the observed maximum absolute difference is $1.315\mu\text{s}$.

6.3 Transport, provenance and remaining availability

The capture recorded 2,102 successful fine feeds, zero feed errors/quarantines, zero USB request/transport errors, zero helper restarts and zero source rotations. Independent reconstruction found 9,866 genuine source grants, exactly matching native acceptance. All 2,102 feed provenance checks and 291 fine-observation deadline/bracket/integer checks passed without violations. These checks verify the recorded contract, not timing truth between recorded events.

The settled source was valid in 235/236 status observations, yet direct delivery was available in 182/236. Of the 54 withheld comparisons, 45 had an unidentified delivery gate (in-flight feeder work is possible), seven had stale reference, one lacked a current anchor and one lacked source permission. The log does not expose the feeder-busy predicate; the 45 cannot be conclusively attributed to that path. Counts are point-sampled availability, not a continuous coverage guarantee. Improving this interface is more urgent than advertising a smaller internal RMS number.

7 Discovery, corrections and separate historical evidence

7.1 Repeated UTC translation noise: found and resolved

The earlier 600 s build recomputed `utc_offset_ns = utc - ns` in radio publication from fresh coarse/raw UTC observations while the precise source-monotonic map evolved independently. Its reported UTC phase had 1.465 ms detrended variation despite quiet short-window monotonic phase. Review localized and corrected this concrete publication behavior in the accepted .51 source change (commit5b7087d).

The time owner now binds one integer translation using a coherent uptime/counter observation within 100 μ s, retains that interpretation across ordinary map/publication revisions, and applies it consistently to precision publication, preferred JOIN and serving display. Contradictory or invalid associations and source restart fence before replacement. The existing 2 s contradiction threshold is a rejection policy, not an accuracy bound. The final 9,866 valid records show one retained translation, resolving the repeated translation variation in this run. Initial PPS/RMC second association and fixed bias still require independent commissioning; this change does not create a verified UTC anchor or repair genuine GNSS reception loss.

7.2 USB continuity and map classification

The .52 USB fix addresses pinned EP0 busy/ZLP retirement and superseded completion generations. The original physical trigger of the earlier startup hang was not established. The .53/CLI.21 change adds the bounded missing-SOF classification described above, preserving invalid-sample refusal and original expiry. Focused C/native checks and a real private-daemon synthetic lifecycle test cover brief retention, unchanged old deadline, true withdrawal, malformed input and required recycling.

The previous separate 300 s .52/CLI.20 trial had zero USB errors but 46 actual chrony restarts, 47 precision-only no-map responses and 103 valid/133 withheld settled comparisons. Its settled direct comparison RMS was 414.137 ns. The final run has zero restarts and 182 valid/54 withheld, while direct RMS stays 414.351 ns. The observed improvement is daemon continuity and sampled deliverability, not a reduction in already sub-microsecond direct agreement. No natural reason 7, retention, refusal or original-expiry event occurred in the final run. Therefore the new classification's physical branch coverage and its causal responsibility for the restart improvement are not proved by this comparison. RF/GNSS and temperature were uncontrolled and may have differed between sequential trials; populations are not pooled.

7.3 Earlier 600 s base-USB experiment

The original 600.013 s phase-only experiment is retained as a separate method/history population. It had 19,093 accepted hardware tuples and 18,332 valid source observations. All 20 disjoint 30 s windows had detrended phase σ below 0.288 μ s (median 0.155 μ s). Fitting each preceding 30 s block and predicting the next without refit gave 1.053 μ s RMS, 3.285 μ s absolute P99 and 3.662 μ s maximum across 17,439 predictions. A single full-run fit left 3.956 μ s source σ and +4.635ppm drift. Window choice was exploratory after capture. Those retrospective/local-block results are not the current causal 328 ns forecast or the current chrony statistic.

Ten historical read-only NTP checks had reported uncertainty ± 52.884 – ± 96.552 ms, too broad to calibrate this USB endpoint. No matched NTP or gPTP comparison was performed in the final trial. The earlier millisecond UTC variation is discovery evidence, not an unresolved description of the corrected current stream. Earlier raw failures, source gaps and rejected observations remain preserved in their original archives.

8 Limits and room for improvement

8.1 What the final results establish

The final run establishes a causal base-USB RMS near 328 ns and a settled delivered user-clock agreement RMS near 414 ns, on one composite TC-01/TC-04/Mac path. It demonstrates a practical userspace chrony integration without operating-system clock control. It does not measure independent UTC error. Shared-source/common-mode error can cancel, chrony’s tracking RMS is internal, and unique frame identity, 62.5 ns timer ticks and a narrow admission bracket are not a full uncertainty budget. Apple/controller/PHY/cable-to-Mach offset, the source’s initial UTC bias and qualified PPS/RMC second association remain unknown.

There is one finite direct-XHCI trial, with no controlled alternate-Mac, hub, suspend/reconnect, thermal or CPU/I/O stress cohort. GNSS/RF conditions were not independently recorded as metrology truth. The experiment covers neither leap-second behavior nor adversarial source integrity. Exclusive USB ownership and identity checks avoid accidental mixing; they are not cryptographic authentication of an attached malicious device. Absolute UTC traceability, continuous precision availability, long-run durability and superiority over NTP/gPTP are not claimed.

8.2 A concrete next sequence

First, instrument every delivered-clock refusal with an explicit gate reason and admission generation. Investigate whether a stable atomic snapshot can permit ordinary same-authority updates without an unnecessary busy refusal; preserve all reset, wall-step, source-transition and expiry fences. The 45 unidentified settled refusals are a diagnosis target, not proof that a specific gate is faulty. This aims to close the 235-source-valid versus 182-deliverable gap without extending authority.

Second, evaluate bridge-window adaptation against frozen causal criteria, with controlled load and temperature and fresh validation runs. Report bias, RMS, tails and availability together. The difference between the whole-run $4.527\mu\text{s}$ counter variation and 328 ns causal forecast supports following local rate, but does not by itself select a universally optimal window. Do not substitute the final 103 ns fitted residual for an end-to-end result.

Third, independently commission the PPS-to-UTC association and calibrate the physical endpoint/Mach bias. A second timed host event or external reference is needed to expose errors shared by the fine feed and chrony. Finally, compare USB, NTP and gPTP at a common independently timed endpoint with matched horizons and availability. The comparisons in related work motivate those experiments; unlike-for-like published values do not rank transports.

9 Conclusion

The corrected stream supports a working macOS user-level clock. In the declared settled window, reported chrony estimator RMS remained 23–339 ns, and direct delivered-clock agreement with its fine source was 414.351 ns RMS. The base USB bridge separately achieved 328.413 ns qualified pre-ingest forecast RMS. There were zero USB request errors and zero actual daemon restarts. Binding the UTC translation removed the repeated coarse-to-fine publication noise observed earlier, while causal fitting followed slower relative rate variation. The next engineering priority is delivery availability; independent endpoint and UTC calibration are the next metrology priorities.

Data, code and reproducibility

This is a local research preprint, not a submitted or accepted paper. The companion paper/`usb-macos-evidence-20261015.json` records exact metrics, separate capture hashes and software versions. The final raw JSONL SHA-256 is `c401eb777976634f7717aeb9637530d111dd72525142661cbaf474aba13022d3`. Its sealed run identifier is `MACOS_SOF_MAP_FIVE_MINUTE_20261015_01`; the baseline is `MACOS_FIVE_MINUTE_20261015_01`. Historical identifiers `TC04_DIRECT_USB_600S_20261015_01` and `TC04_SHORT_WINDOW_STABILITY_20261015_01` remain separate.

The standalone LaTeX source embeds every plot coordinate and the bibliography. Derived CSVs contain elapsed time and relative measurements without physical identities or epoch-sized UTC labels. The derivation checks all rows in the fixed windows and retains withheld markers. Raw captures, protected readbacks, private sockets and hardware identifiers remain outside website assets. New CLI images replay the actual last eligible fresh status at 299.904023 s through the native renderer; only the preview’s duration is corrected from 600 to 300 s. They are labeled replays, not physical display captures. Build, literature, claim-review and figure manifests accompany the review package. No public raw-data release or off-host archival guarantee is asserted.

Author contributions and assistance disclosure

Jonathan Yantis supplied the hardware, original protocol/firmware, experimental direction and research decisions. OpenAI Codex assisted implementation, tooling, literature research, evidence review, analysis, figure/table generation, writing and LaTeX preparation. The same Sol 6.1 Extra High implementation engineer reviewed this revision’s claims against code and final capture evidence. No AI system is listed as a human coauthor. The human author retains responsibility for the evidence, claims, references, privacy and any venue requirements. No affiliation, acceptance, DOI or arXiv identifier is claimed.

References

- [1] J. Yantis with disclosed OpenAI Codex assistance, *Passive Precision Time over BLE: V14 Boundary Clocks, Causal Timestamping, and Prospective One-Relay Validation*, local review preprint, 12 October 2026. Repository paper/`ptob-v14-preprint.tex`. Prior radio evidence is not pooled into this USB result.
- [2] IEEE 802.1 Working Group, *IEEE Std 802.1AS-2025: Timing and Synchronization for Time-Sensitive Applications*, published-standard listing and maintenance project scope. <https://1.ieee802.org/tsn/>; <https://1.ieee802.org/maintenance/802-1as-2025-revision-timing-and-synchronization-for-time-sensitive-applications/>. Accessed 15 October 2026. Public project material inspected; no claim of a clause-by-clause conformance review.
- [3] J. Clark, *mac-gptp: Use gPTP on macOS for timing*, README and implementation documentation, commit `e133922d72282e1a41835a19e575663d292a21c5`, 4 September 2026. <https://github.com/jclark/mac-gptp/tree/e133922d72282e1a41835a19e575663d292a21c5>. Primary project report, accessed 15 October 2026; not independently reproduced here.
- [4] J. Clark, *gptp-refclock*, technical notes at the same pinned revision. <https://github.com/jclark/mac-gptp/blob/e133922d72282e1a41835a19e575663d292a21c5/docs/gptp-refclock.md>. The reported 0.6 μ s RMS is a system-clock residual against the project’s gPTP reference, not calibrated absolute UTC error.
- [5] I. Val, Ó. Seijo, R. Torrego and A. Astarloa, “IEEE 802.1AS Clock Synchronization Performance Evaluation of an Integrated Wired-Wireless TSN Architecture,” *IEEE Transactions on Industrial Informatics*, 2022, doi:10.1109/TII.2021.3106568. Author manuscript arXiv:2109.09409, 2021, Table III. <https://arxiv.org/abs/2109.09409>.

- [6] M. A. Weiss, Y. Li-Baboud, D. Anand, K. G. Brady, P. A. Boynton, C. T. Nguyen, M. J. Burns and A. M. Gopstein, *A Calibration of Timing Accuracy in the NIST Cyber-Physical Systems Testbed*, NIST Technical Note 2030, December 2018, doi:10.6028/NIST.TN.2030. <https://www.nist.gov/publications/calibration-timing-accuracy-nist-cyber-physical-systems-testbed-0>. The Technical Note summary, rather than the differing conference abstract, supplies the selected maximum/uncertainty wording.
- [7] C. H. Lim, B. A. bin Rosdi and C. F. Yap, “Synchronization of multiple USB 3.0 devices using Isochronous Timestamp Packet,” *Computer Standards & Interfaces*, 2017, doi:10.1016/j.csi.2016.06.006. <https://www.sciencedirect.com/science/article/pii/S0920548916300423>. Publisher abstract/highlights inspected; full experimental article not available in this review.
- [8] P. G. Foster, *USB based synchronization and timing system*, US8943351B2, 2015. <https://patents.google.com/patent/US8943351B2/en>. Primary disclosure of USB periodic-signal clock synchronization; a patent disclosure is not an independently reproduced benchmark.
- [9] Google WALT contributors, “Clock Synchronization,” primary implementation notes preserved in Android Open Source Project. <https://android.googlesource.com/platform/external/walt/+HEAD/android/WALT/app/src/main/jni/README.md>. Accessed 15 October 2026; named-machine interval widths retained as reported.
- [10] D. Mills, J. Martin, J. Burbank and W. Kasch, “Network Time Protocol Version 4: Protocol and Algorithms Specification,” RFC 5905, June 2010. <https://www.rfc-editor.org/rfc/rfc5905.html>.
- [11] Nordic Semiconductor, *nRF52840 Product Specification*, v1.2, USB-D isochronous transactions / SOF event and TIMER capture descriptions. https://docs-be.nordicsemi.com/bundle/nRF52840_PS_v1.2/raw/resource/enus/nRF52840_PS_v1.2.pdf. Accessed 15 October 2026; source documents 16 MHz synchronization before PPI, not a calibrated installed-device timing bias.
- [12] Apple Inc., “referenceMicroframeWithTime:error:,” IOUSBHostObject documentation and local macOS SDK declaration. <https://developer.apple.com/documentation/iusbhost/iusbhostobject/referencemicroframewithtime:error:>. Accessed 15 October 2026. The public API is distinguished from DriverKit interfaces and private TimeSync.
- [13] Apple Inc., “frameNumberWithTime:,” IOUSBHostObject documentation and local SDK declaration, accessed 15 October 2026. <https://developer.apple.com/documentation/iusbhost/iusbhostobject/frameNumberWithTime:>. The inspected SDK explicitly describes omission of the microframe component.
- [14] J. Yantis with OpenAI Codex assistance, *TC-04 USB and macOS chrony evidence*, sealed local experiments, 15 October 2026 operator date. Companion paper/usb-macos-evidence-20261015.json records the separate historical600s, baseline300s and final300s populations and their hashes. Raw captures are private.
- [15] Chrony contributors, *chronyd(8)*, *chronyc(1)* and *chrony.conf(5)*, version 4.9 documentation. <https://chrony-project.org/doc/4.9/chronyd.html>; <https://chrony-project.org/doc/4.9/chronyc.html>; <https://chrony-project.org/doc/4.9/chrony.conf.html>. Accessed 16 October 2026 UTC. Local pinned implementation and configuration are retained with the experiment.