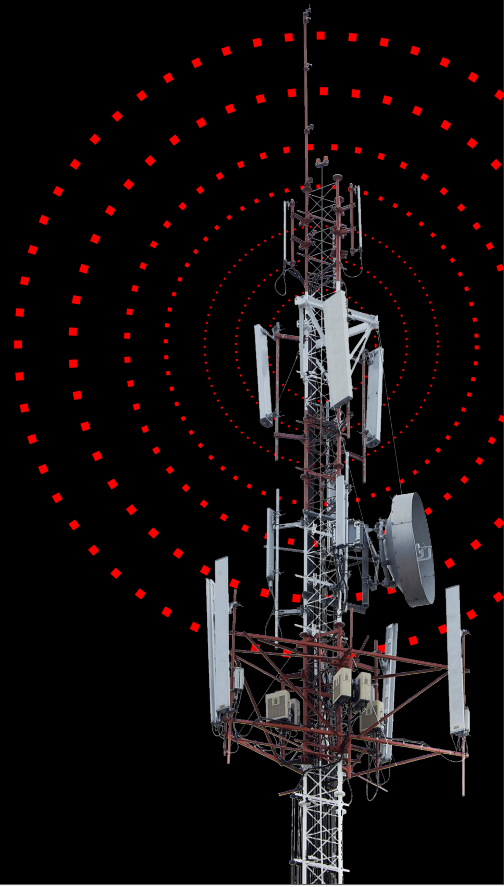
Getting in Sync with Open Source - to SyncE or not to SyncE

International Timing and Sync Forum
November 8, 2022
Düsseldorf, Germany



Pasi Vaananen
Systems Architect, Office of the CTO
Telco Enablement & Solutions

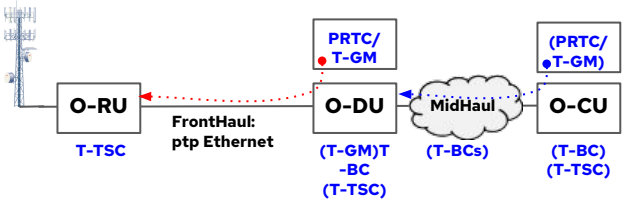
Timo Jokiaho
Chief Technologist
Telecom, Media & Entertainment - EMEA

Red Hat

Agenda

- Synchronization requirements in context of (O-)RAN
- RAN synchronization network topologies
- D-RAN site example recap
- RAN Server node synchronization HW & SW implementation
- Testing Capabilities @ Red Hat
- Impact of SyncE to Synchronization performance (test results summary)

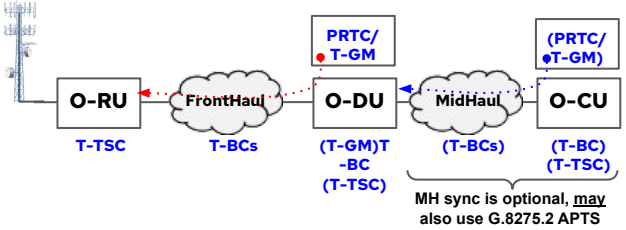
O-RAN “LLS-Cx” Synchronization Reference Configs



WG4 Configuration LLS-C1

FH Direct connection to O-RU(s) from O-DU; sync source in O-DU

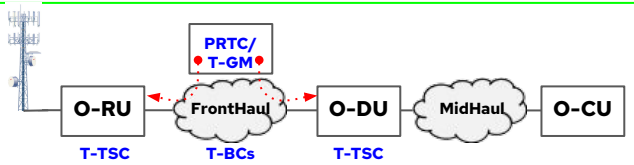
WG9 C1, Option A: GNSS/T-GM Embedded in ~~O-DU~~ (Cloud node)
WG9 C1, Option B: T-GM Directly Connected to O-DU
WG9 C1, Option C: T-GM connected to O-DU via chain of network nodes)



WG4 Configuration LLS-C2

FH Network connection to O-RU(s) from O-DU; sync source in O-DU

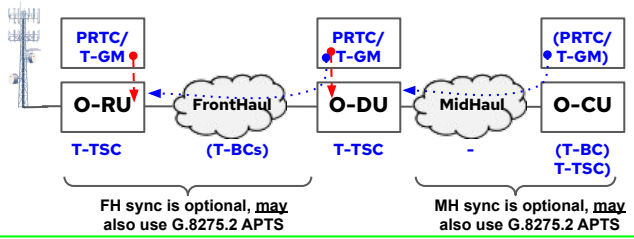
WG9 C2, Opt. A: ~~O-DU~~ (Cloud node) is the nearest common T-BC
WG9 C2, Opt. B: nearest common T-BC not O-DU



WG4 Configuration LLS-C3

Network connection to O-RU from O-DU & sync source in FH network

WG9 C3, Option A: T-GM is the nearest common master
WG9 C3, Option B: nearest common master is not T-GM
WG9 C3, Option C: T-GM in Mid/Back-haul
WG9 C3, Option D: T-GM in Mid/Back-haul with T-BC chain



WG4 Configuration LLS-C4

Network connection to O-RU from O-DU & local sync sources

WG9 C4, Option 1: GNSS at Cell Site (e.g. in O-RU / xNB)
WG9 C4, Option 2: GNSS at Cell & Edge + APTS network

Note:
In clouds, synchronization is part of cloud infrastructure, and decoupled from RAN instances

Red Hat

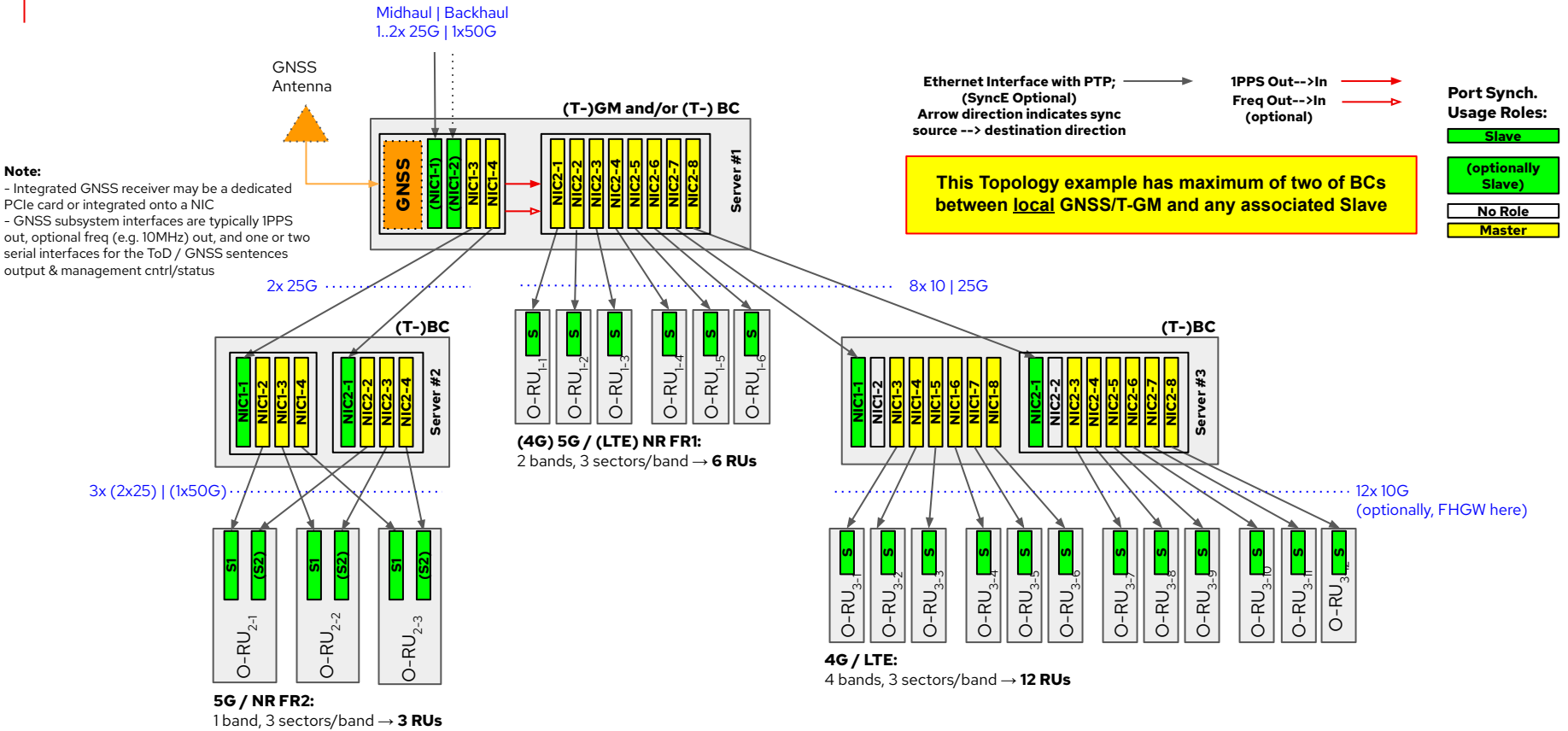
O-RAN LLS-Cx Decomposition for Cloud Features

Key Features	O-RAN CUS LLS-C1; FH Directly connect to O-RU from O-Cloud; Sync source in O-cloud	≈ O-RAN CUS LLS-C2; FH Network connection to O-RU from O-Cloud; sync source in O-Cloud	O-RAN CUS LLS-C3; Network connection to O-RU from O-Cloud & sync source in FH network	O-RAN CUS LLS-C4; Direct or Network connection to O-RU's from O-Cloud & sync sources in RU and O-Cloud
Mgmt Cfg/Metrics/Events (local and system evt. notifications)	Yes			
Node Linux Clock Synchronization from NIC Clock (e.g. phc2sys)	Yes			
G.8275.1 Full Timing Support T-TSC (G.8273.2 class A-D)	Yes (class A/B or better)	Yes (Class **B/C** or better)	Yes (class A or better)	Opt. (class A/B or better)
G.8275.2 Partial Timing Support (T-TSC-A/P)	Alternative Option vs. G.8275.1, less requested esp. on LLS C2/C3 configurations due to accuracy uncertainties			
G.8275.x FTS/PTS T-TSC redundancy	Yes, when redundant network connectivity available (mode common in C-RAN than D-RAN, but applies to both)			
PPS out/in (x-connect in multi card configs)	Yes, when supported in HW (O/I)		Yes, If supported in HW (O)	If supported in HW (O/I)
10 Mhz out/in (x-connect in multi-card configs w/ SyncE)	Yes, when supported in HW (O/I)		Yes, If supported in HW (O)	If supported in HW (O/I)
G.8275.1 FTS T-BC (G.8273.2 class A-D); one/multiple NICs	Yes (typ. class B or better)	Yes (typ. **class C** or better)	(N/A in node, in-RU-path node T-BC's typ. Class C or better)	Opt. (class B/C or better)
G.8262 (or G.8262.1) SyncE; one/multiple NICs	Yes w/ 8275.1 T-BC, when required to support O-RU		Option, typ. not in nodes	Opt. typically w/ T-BC
G.8275.2 PTS T-BC; one/multiple NICs	Alternative to G.8275.1, accuracy concerns in large configs		-	Alternative for G.8275.1
G.8275.1 FTS T-GM (Ext/Int >=PRTC-A)	Option, if supported in HW (GNSS receiver)		-	Yes, if supported in HW (GNSS receiver)
G.8275.2 PTS T-GM (Ext/Int,>=PRTC-A)	Option, if supported in HW (GNSS receiver)		-	Yes, if supported in HW (GNSS receiver)
G.8275.x FTS/PTS T-GM redundancy	Yes, when redundant HW available (more common in C-RAN than D-RAN, applies to both)		-	Yes, when redundant HW available
G.8275.2 APTS	-	-	-	Optionally Yes

Note: CUS LLS-Cx Configs are sufficiently "nebulous" to be un-usable as spec due to large variety of options within ea. - trust but verify, i.e. **focus on feature set & specific reqs. vs. LLS-Cx**

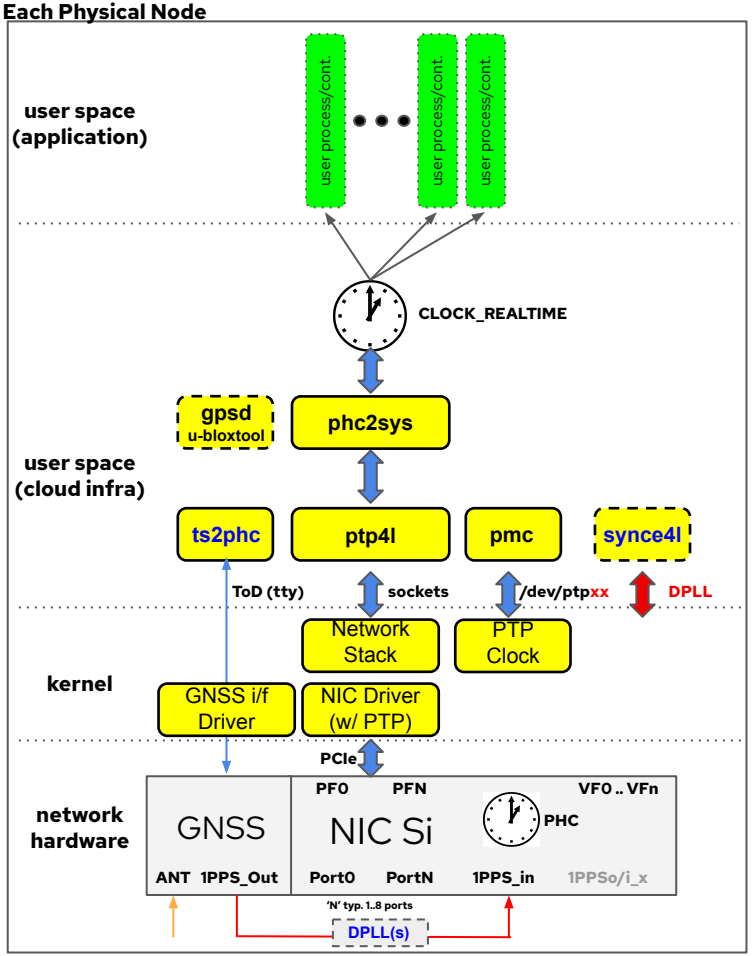
Red Hat

Dense, High-Capacity 21-sector D-RAN site example



Red Hat

Synchronization Components in Linux / k8s nodes



Key Components of the node PTP implementation

- HW specific synchronization SW support features are implemented in HW device drivers
- HW Clock (PHC) support in NIC Si is required for high accuracy; SyncE requires DPLLs
- **Linuxptp** is an Open Source project implementation of the PTP SW stack for Linux
- **ptp4l** implements Boundary Clock (BC) and Ordinary Clock (OC), it synchronizes PTP hardware clock (PHC) to remote master clock
- ptp4l is very flexible, and can be configured to support specific profiles, assuming that HW & driver supports associated features (e.g. PHC, L3 vs. L2 transport, accuracy targets)
- **phc2sys** synchronizes two or more clocks in the system, typically used to synchronize the system clock from PTP / PHC
- **pmc** - PTP management client; 1588 basic management access for ptp4l
- **ts2phc** synchronizes PHC(s) from external reference signals, such as 1PPS_in and ToD messages - used in multi-card T-BC, and GNSS driven T-GM configurations
- **synce4l:** implements ESMC protocol and associated state transition controls
- **gpsd/u-blox tool** can be used to interface w/ GNSS receivers / u-blox specifics
- In k8s clusters, synchronization functions are configured and monitored with k8s, and associated general CM, PM and FM event and metrics tools.
- O-RAN WG6 specifies a notification interface to inform synchronization users about node synchronization state changes

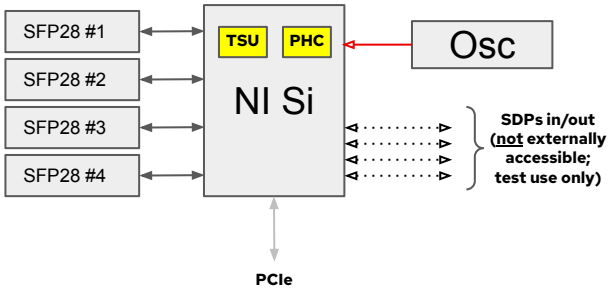
intel. NVIDIA. XILINX ... And many other Si & HW Partners

BROADCOM ublox Red Hat

2 kinds of typical HW from Sync. & SW Perspective

- Synchronization optimized designs add various functions, typically DPLL and PHY clock recovery/generation, as well as phase/freq physical inputs and outputs, and GNSS options all which need SW enablement for configuration, monitoring and control purposes
- Various design specific ways to achieve equivalent things, none is “right” or “wrong”, they are just different; goal is to abstract to the “right” level through Linux interfaces and device drivers
- FPGA cards and SoCs with embedded NI functionality are generally similar in design from sync perspective (i.e. incorporate “NI Si” block below)

“Basic” PTP HW-timestamping capable HW Without Synchronization Enhancements Features



Phase Out-->In

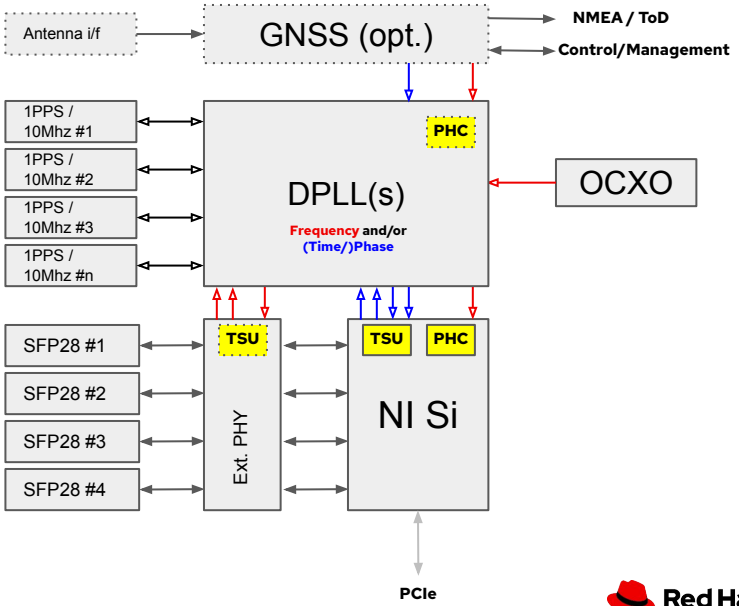
Freq Out-->In

Freq or Ph Out or In (design specific and/or configurable)

Ethernet i/f in/out

PHC: PTP HW Clock
TSU: TimeStamping Unit
1PPS: 1 Pulse Per Second
GNSS: Global Navigation Satellite System
DPLL: Digital Phase Locked Loop
OCXO: Oven Controlled Crystal (Xtal) Oscillator
ToD: Time of Day
NMEA: National Marine Electronics Association
PCIe: Peripheral Component Interconnect express

HW With Synchronization Enhancements Features



Red Hat

The State of the Open Source SW for Sync(E) Support

- **1PPS / 10Mhz input/output pins**
  - 1PPS in/out & 10Mhz out can already be supported by existing kernel interfaces
  - 10Mhz in will be expected to go to freq DPLL, dependent on the DPLL kernel interface (see below)
- **DPLLs as clock devices**
  - Can be exposed using existing interfaces, some DPLL vendors already implement this
- **DPLL new kernel Interface**
  - Work in progress; these are complex devices with lots of functionality
  - Will enable common way to configure & conduct basic operations on DPLLs
  - Will enable monitoring of DPLL operations and performance (state changes, performance metrics, ...)
  - Goal is to abstracts the HW implementations from user space processes (like ESMC protocol / sync management)
  - DPLLs are applicable to both frequency (SyncE) and phase (e.g. GNSS, 1PPS, PHCs ...) use cases in isolation/in combination
- **SyncE - NIC driver and/or firmware additions**
  - Presently OOT, available in sourceforge for Intel cards
  - Full upstreaming is dependent on the kernel interface for DPLL
- **SyncE - ESMC protocol & control: Intel**
  - This is very "LinuxPTP-like", similar approach to other LinuxPTP processes
  - This was 1st released in LinuxPTP mailing list, it is now available in github
- **SyncE - ESMC protocol & control: Renesas**
  - Renesas has also made an implementation available
  - Somewhat different in terms of interfaces / look-and-feel as compared to other LinuxPTP processes
  - This **was** available in github; looks like it has been taken private since initial release
- **Having alternative implementations is both good and bad**
  - This is progress - last year there was nothing yet on public, now there's already two to choose from :)
  - We (i.e. Red Hat) do want avoid the need for multiple implementations for the same problem
  - Preferably the community settles to / around ONE implementation and/or merges the best of initial submissions
  - First step is to get the required kernel interfaces (DPLL) in place first & ASAP

Note:
Intel implementation was used as a basis for SyncE & ESMC and associated tests in this work

Synchronization Lab Test Setup at Red Hat TelcoLab

Path-delay calibrated Optical single-mode Switch for remote/automated test reconfiguration

Spirent TestCenter for Bulk Traffic testing (≤1.2Tbps)

Spirent TC for Detailed Perf. Testing (≤800Gbps), w/ PTP/SyncE slave emulation

Calnex Attero-100G Network Emulation System

Calnex Paragon Neo PTP/SyncE analyzer w/ comprehensive SW feature set for Telco PTP & SyncE

Keysight 8ch Scope w/ time&freq analysis SW

Cisco&Dell PTP/SyncE capable Switches & routers (back)

Primary GNSS signal splitters (roof antenna + simulator)

1PPS & 10Mhz reference distribution amplifiers

Primary ePRTC (TP4100 + 5071A w/ hi-performance CBT)

Secondary ePRTC (TP4100 + 5071A w/ hi-performance CBT)

Spirent GNSS Constellation Simulation System

22x servers w/ various NICs, FPGA, SoC etc. PCIe cards w/ Sync. Support ("DUT" servers)

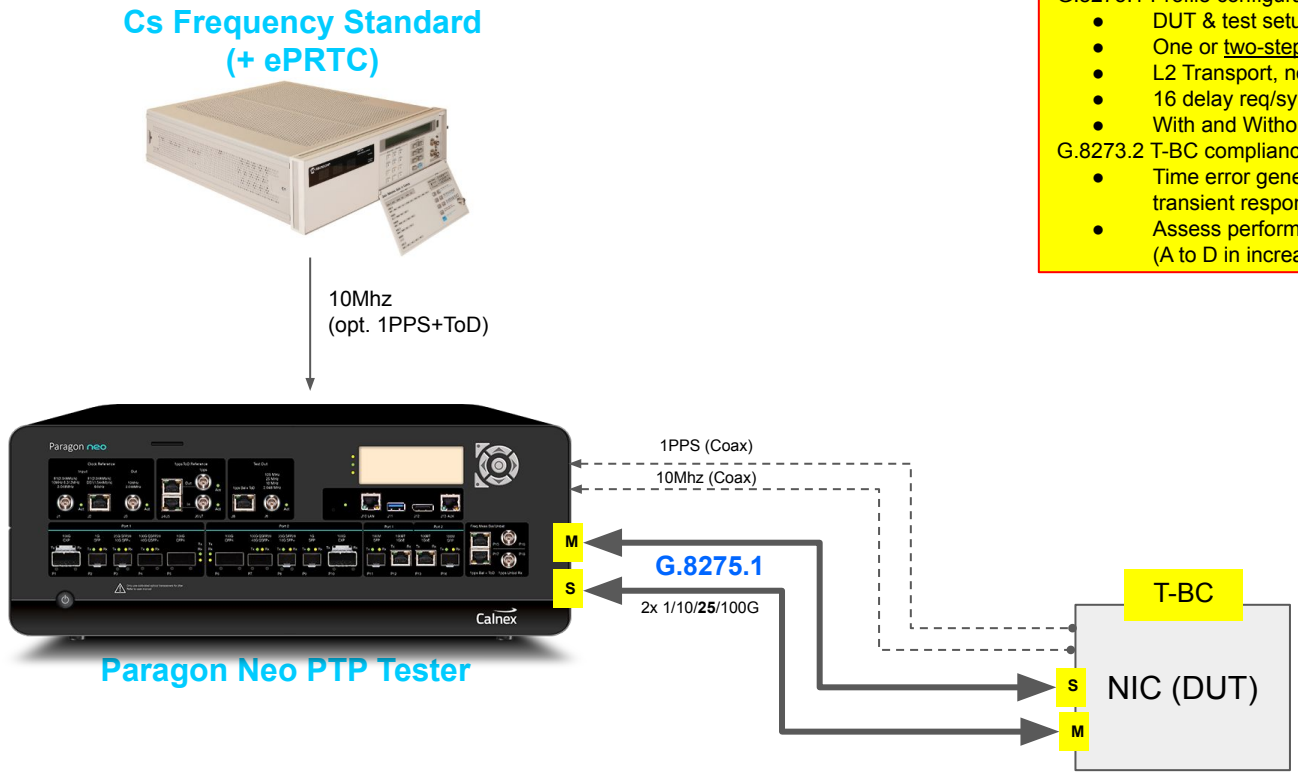
<u>Not</u> Shown: O-RU & UE emulation systems, Test O-RU's, 50Ghz Spectrum Analyzer, very high-end scope and OTA performance test gear etc.

Red Hat

Set-up for a 1-NIC; G.8275.1 & G.8273.2 T-BC

NOTES:
G.8275.1 Profile configuration
- DUT & test setup configured for T-BC
- One or two-step Timestamp (depending on DUT HW)
- L2 Transport, no VLAN tagging
- 16 delay req/sync, 8 announces per second
- With and Without SyncE

G.8273.2 T-BC compliance
- Time error generation / tolerance / transfer, short-term transient response, holdover, ...
- Assess performance of implementation to perf Classes (A to D in increasing level of accuracy)



Red Hat

The Test applicability – non-SyncE and SyncE capable HW

Test Cases	Basic HW	Enhanced HW (w/ SyncE)
G.8273.2 7.1.x: TE Noise Generation	Yes, all; 7.1.4.x TE_R defined for Class-C only	
G.8273.2 7.2.x: Noise Tolerance	Yes 7.2.1 (A/B) vs. 7.2.2 (C)	Yes, 7.2.2 (Class C)
G.8273.2 7.3.x: Noise Transfer	7.3.1 PTP-PTP only	Yes, 7.3.1 PTP-PTP & 7.3.3 phy. layer freq to PTP (C/D)
G.8273.2 7.4.1 Transient Response	(7.4.1.2 PTP only; perf is FFS)	Yes
G.8273.2 7.4.2 Holdover Performance	(7.4.2.1 PTP only; perf is FFS)	Yes; Class-C mask FFS

- We do also conduct the applicable tests for 1PPS and/or 10Mhz phase/freq. Reference outputs if those outputs are supported as physical interface with connectors
- In “standard” cards without connectorization, we may also use these for testing if available e.g. through pin headers, but they are expected to **not** be used/usable/supported in end application
- Also, G.8262.1 eEEC test sequence typically applies to SyncE capable cards, not shown here
- We view eEEC tests more of DVT tests, minimum SW impact; primarily FW/driver/DPLL configs

G.8273.2 T-TSC & T-BC Noise Gen Reqs Summary

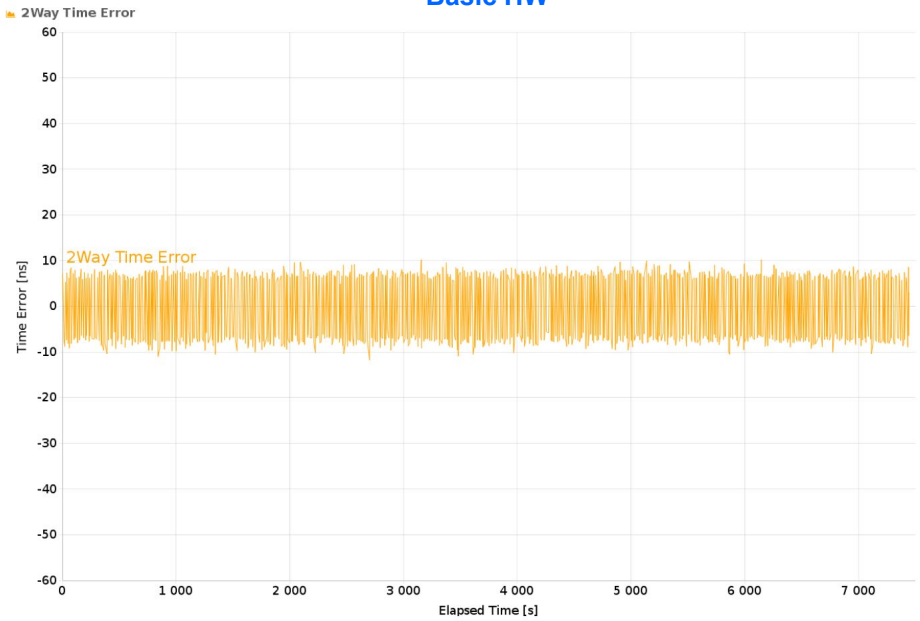
Parameter	Class-A	Class-B	Class-C	(Class-D); Still WIP in ITU-T	Notes
7.1 Max. Absolute Time Error; max|TE|	≤ 100 ns	≤ 70 ns	≤ 30 ns	(≤ 15 ns)	Unfiltered measurement, absolute value
7.1 Max. Absolute Time Error; max|TE_L|	-	-	-	≤ 5 ns	0.1Hz low-pass filtered, 1000s measurement, absolute value
7.1.1 Max. Constant Time Error; cTE	≤ ±50 ns	≤ ±20 ns	≤ ±10 ns	(≤ ± 4 ns)	cTE Averaged over 1000s
7.1.2 Max. dynamic Time Error, 0.1Hz Low-Pass Filtered; dTE_L (MTIE)	≤ 40 ns		≤ 10 ns	(≤ 3 ns)	MTIE Mask, 1000s observation interval constant temp., (10000s for A/B variable temp.)
7.1.2 Max. dynamic Time Error, 0.1Hz Low-Pass Filtered; dTE_L (TDEV)	4 ns		2 ns	(≤ 1 ns)	TDEV Mask, 1000s observation interval at constant temp.
7.1.3 Max. dynamic Time Error, 0.1Hz High Pass Filtered; dTE_H	70 ns p-p		FFS (30ns p-p?)	(15 ns p-p)	Peak-to-peak value, 1000s measurement
7.1.4.1 Relative constant Time Error Noise Generation; cTE_R	FFS		≤ ±12 ns	FFS	cTE averaged over 1000s
7.1.4.2 Relative dynamic Time Error Low-Pass Filtered Noise Generation; dTE_{RL} (MTIE)	FFS		≤ 14 ns	FFS	MTIE Mask; 1000s observation interval at constant temp.

Note: Accuracy required is primarily determined by specific Use Case requirements & number of elements on the synchronization transfer path

Red Hat

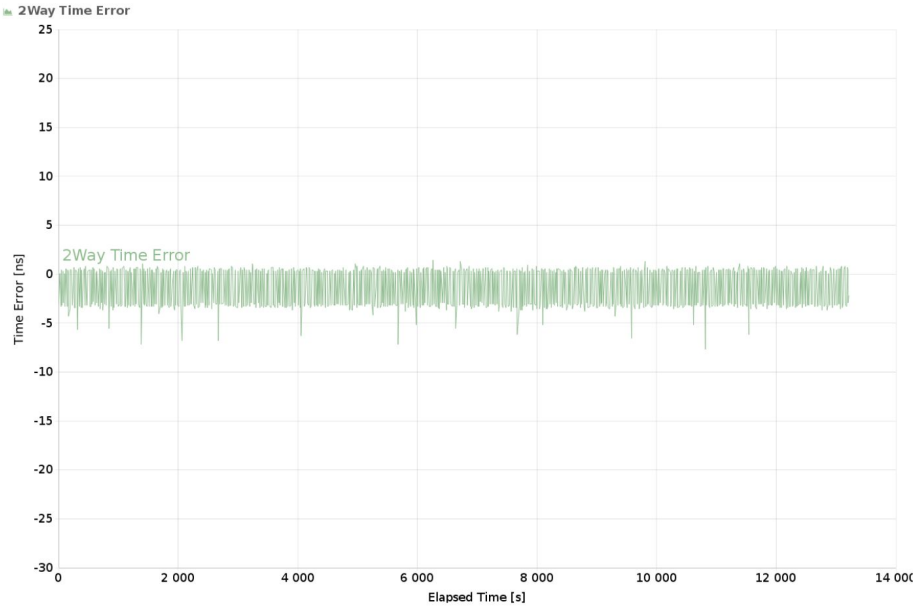
7.1 T-BC Noise Generation - Unfiltered TE

Basic HW



Pk-Pk	22 ns
Mean	-0.374 ns
Min	-11.805 ns
Max	10.195 ns
Max-Min	22 ns

Enhanced HW With SyncE



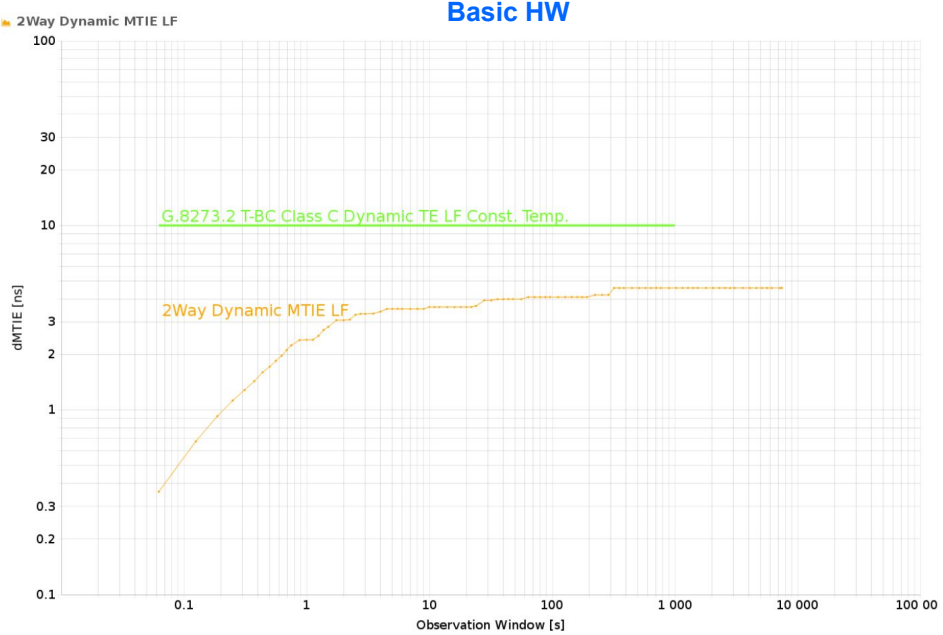
Pk-Pk	9.125 ns
Mean	-1.461 ns
Min	-7.68 ns
Max	1.445 ns
Max-Min	9.125 ns

- Both DUTs do pass Class-C for TE with lots of margin (Class-C requires 30ns for Unfiltered |TE|)
- The performance with SyncE in this test is ~2x better then without it in the same configuration

Red Hat

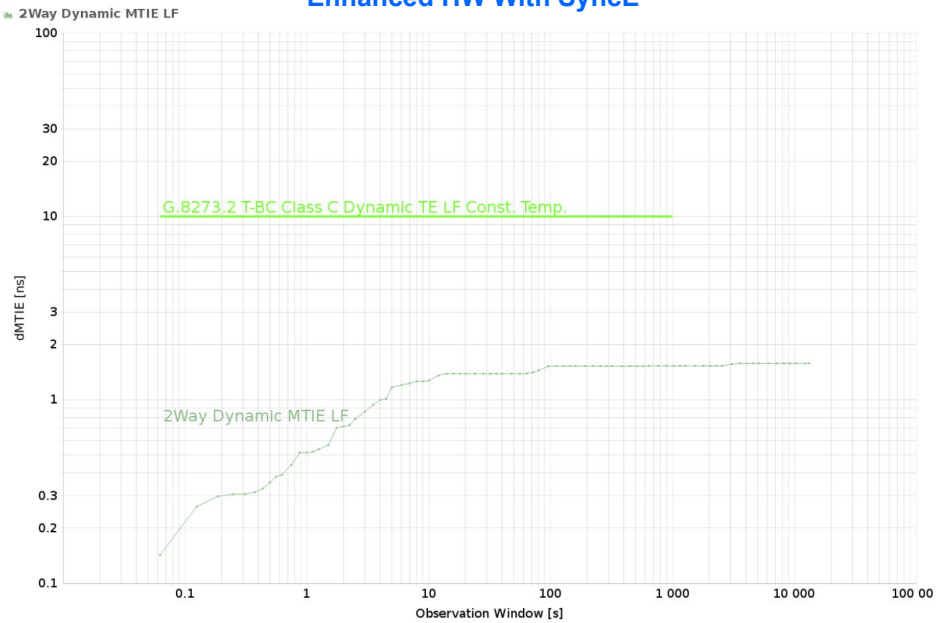
7.1.2 T-BC Noise Generation - Dyn. MTIE LF Class-C mask

Basic HW



Min	0.361 ns
Max	4.58 ns
Max-Min	4.219 ns

Enhanced HW With SyncE



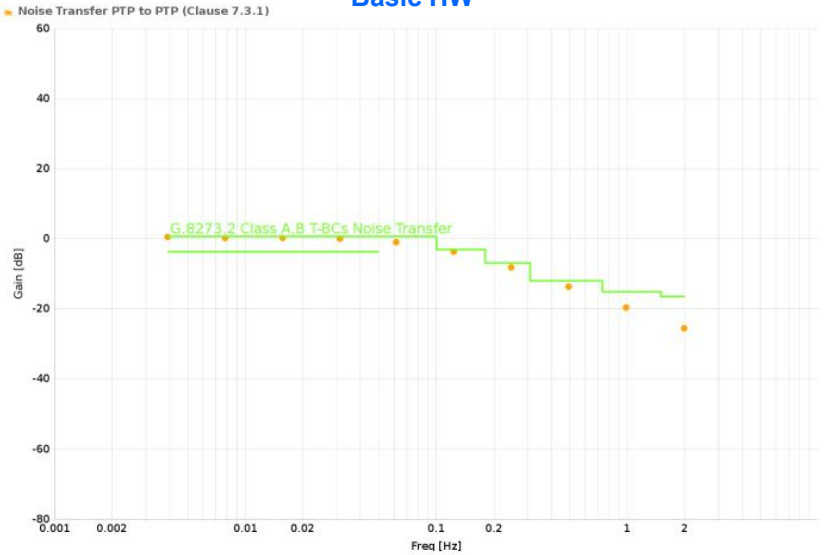
Min	0.142 ns
Max	1.582 ns
Max-Min	1.439 ns

- Both DUTs do pass Class-C for MTIE LF 10ns mask (as well as TDEV mask) with margin
- The performance with SyncE in this test is >2x better then without it in the same configuration

Red Hat

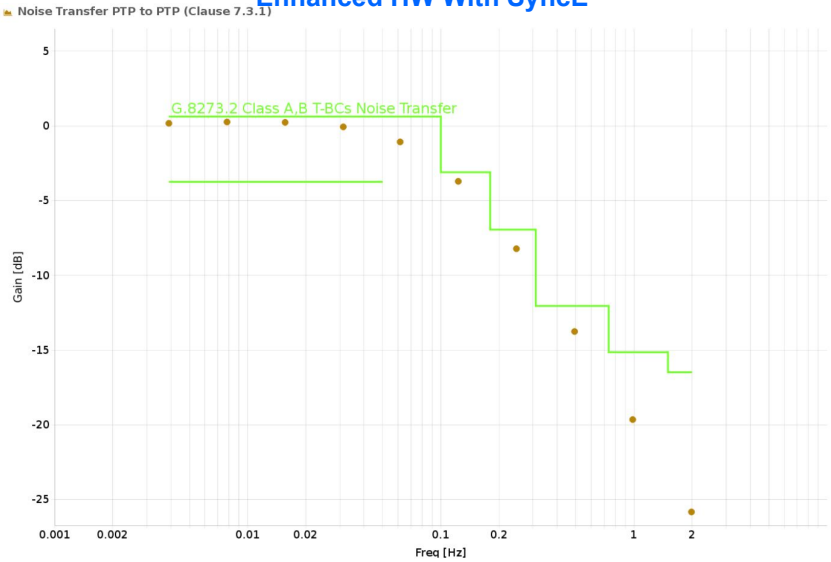
7.3.1 T-BC PTP-PTP Noise Transfer

Basic HW



Point #	Freq (Hz)	Ampl (ns)	Duration (s)	Pk-Pk Output (ns)	Gain (dB)	Upper Pk-Pk Limit (ns)	Lower Pk-Pk Limit (ns)
1	0.00390625	200	1024.00	211.36	0.48	215	130
2	0.0078125	200	896.00	204.57	0.20	215	130
3	0.015625	200	448.00	205.11	0.22	215	130
4	0.03125	200	224.00	198.95	-0.05	215	130
5	0.0615625	200	243.65	178.00	-1.01	215	-
6	0.123125	200	251.78	130.57	-3.70	140	-
7	0.24625	200	251.78	77.73	-8.21	90	-
8	0.4925	200	249.75	41.25	-13.71	50	-
9	0.985	200	249.75	20.84	-19.64	35	-
10	1.985	200	249.87	10.53	-25.58	30	-

Enhanced HW With SyncE

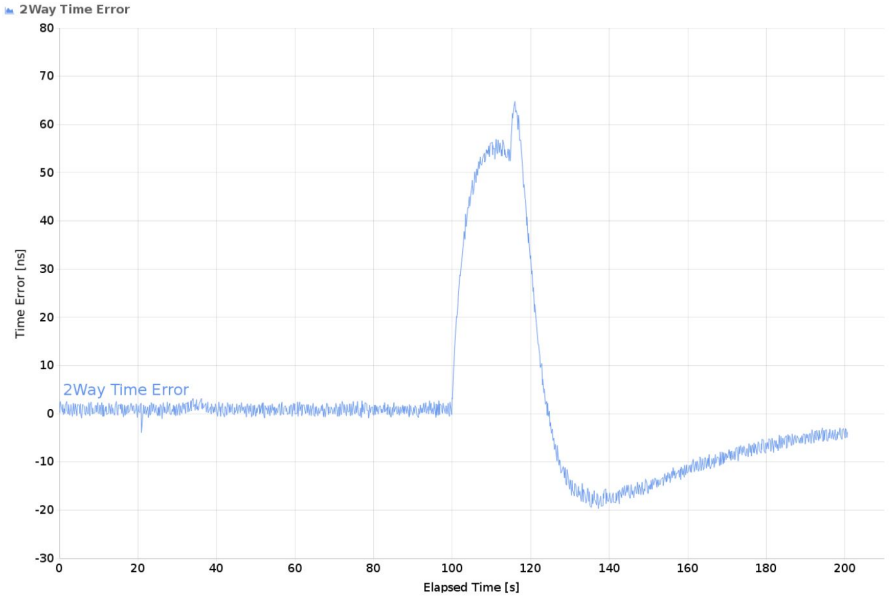


Point #	Freq (Hz)	Ampl (ns)	Duration (s)	Pk-Pk Output (ns)	Gain (dB)	Upper Pk-Pk Limit (ns)	Lower Pk-Pk Limit (ns)
1	0.00390625	200	1024.00	204.09	0.18	215	130
2	0.0078125	200	896.00	206.18	0.26	215	130
3	0.015625	200	448.00	205.57	0.24	215	130
4	0.03125	200	224.00	198.53	-0.06	215	130
5	0.0615625	200	243.65	176.84	-1.07	215	-
6	0.123125	200	251.78	130.53	-3.71	140	-
7	0.24625	200	251.78	77.68	-8.21	90	-
8	0.4925	200	249.75	41.06	-13.75	50	-
9	0.985	200	249.75	20.83	-19.65	35	-
10	1.985	200	249.87	10.23	-25.82	30	-

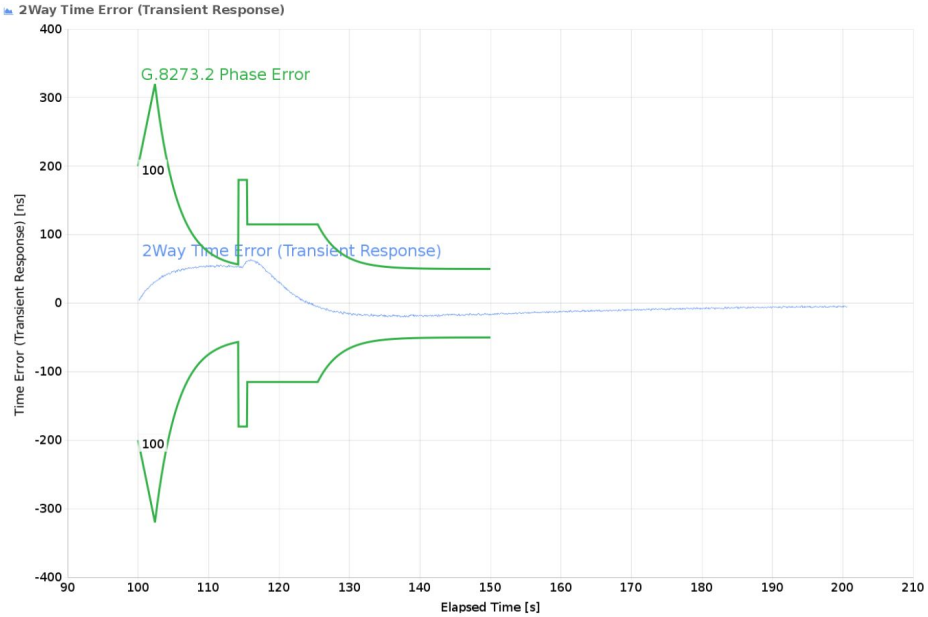
- Both DUTs can pass PTP-PTP Noise Transfer test with appropriate servo and filtering configuration
- Some engineering tradeoffs between initial synchronization time and “slowing down” the servo

Red Hat

7.4.1.3 T-BC Transient Response (SyncE HW only)



Pk-Pk	84.5 ns
Mean	1.455 ns
Min	-19.694 ns
Max	64.806 ns
Max-Min	84.5 ns



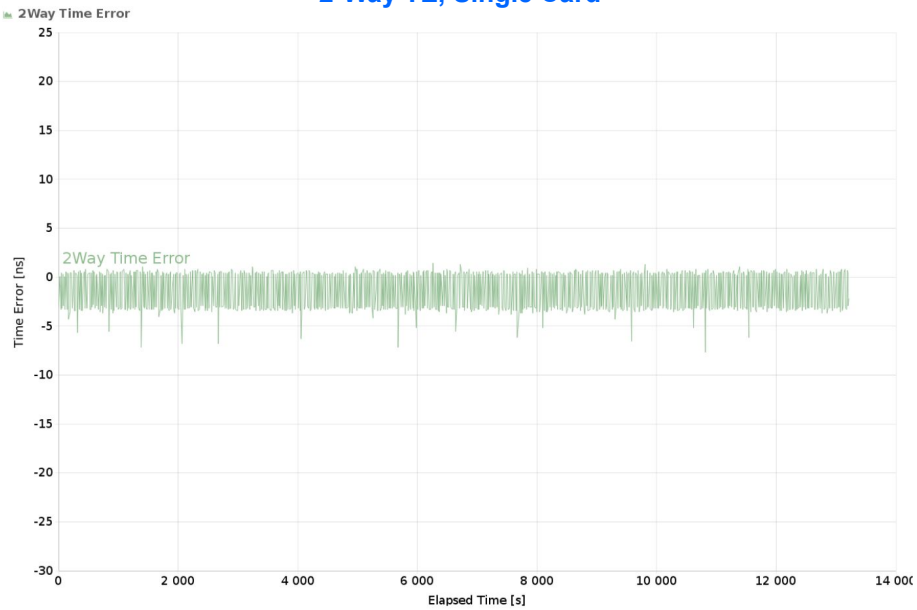
Mean	1.153 ns
Min	-20.571 ns
Max	63.929 ns
Max-Min	84.5 ns

- This is w/ SyncE only, no definitions to PTP only case
- Note that the mask for Class-C/D is FFS; above mask is Annex-B mask for Class A/B

Red Hat

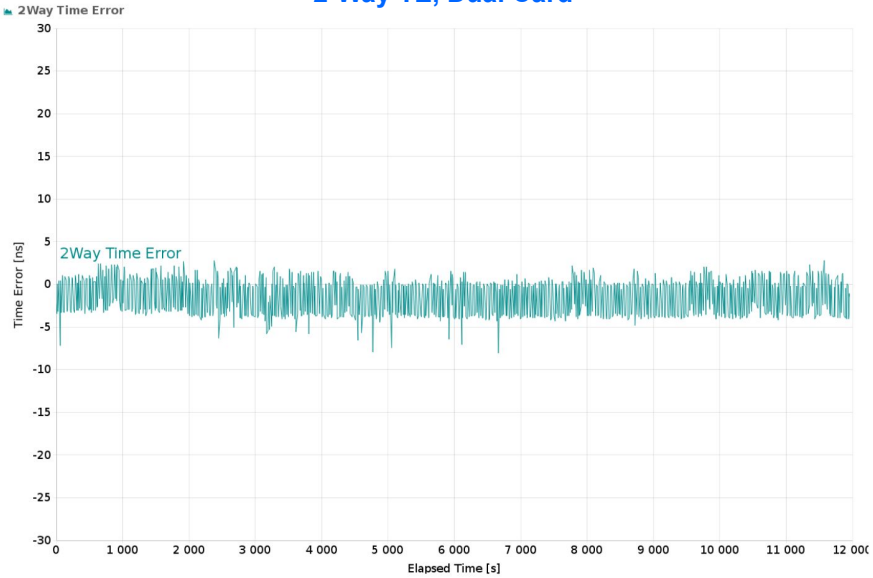
7.1 T-BC Noise Generation - 1 vs. 2-Card with SyncE

2-Way TE, Single Card



Pk-Pk	9.125 ns
Mean	-1.461 ns
Min	-7.68 ns
Max	1.445 ns
Max-Min	9.125 ns

2-Way TE, Dual Card

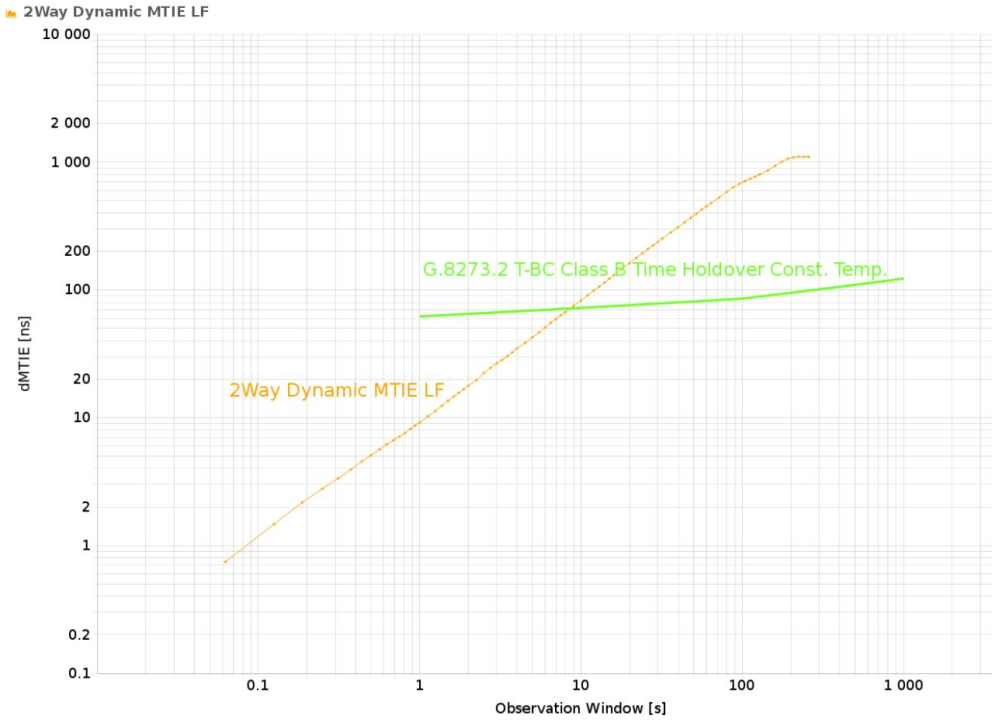


Pk-Pk	10.875 ns
Mean	-1.6 ns
Min	-8.063 ns
Max	2.812 ns
Max-Min	10.875 ns

- In this test, the BC slave port is in 1st card, and master port is in 2nd card
- Combination still passes Class-C for TE with margin (important, multi-card configs are common !)
- This is primarily enabled by physical phase/freq signals to sync the 2nd card from the 1st

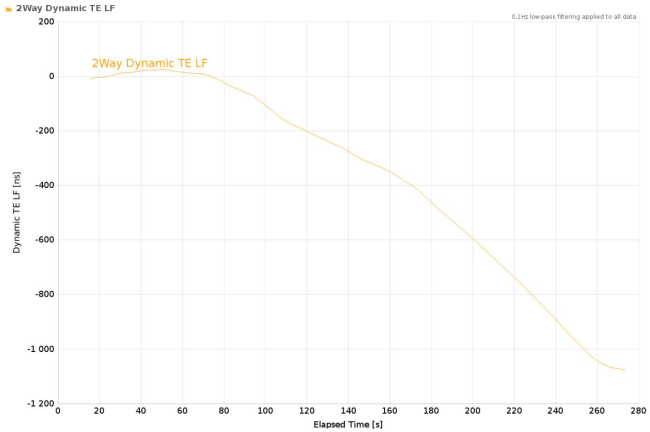
Red Hat

Holdover Performance – Basic Card (no SyncE HW)



Min	0.745 ns
Max	1100.83 ns
Max-Min	1100.086 ns

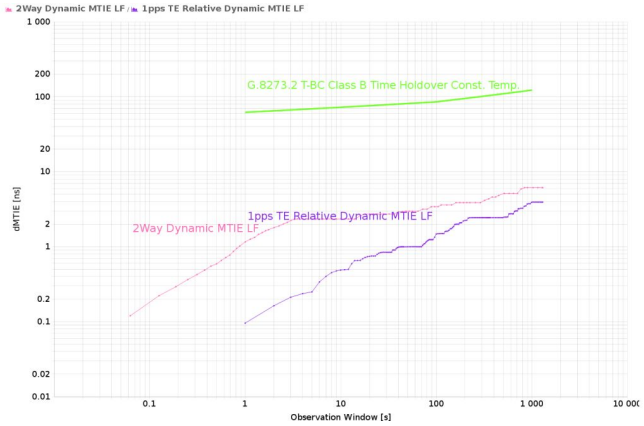
- This is **tested just for reference**
- Does not pass class-B mask
- Because it was not designed to !
- Not all use cases need long Holdover times
- Redundancy is viable option instead in many real-world deployments
- Still able to stay in usable sync state for short transient interruptions
- ~seconds, depending on the target TE



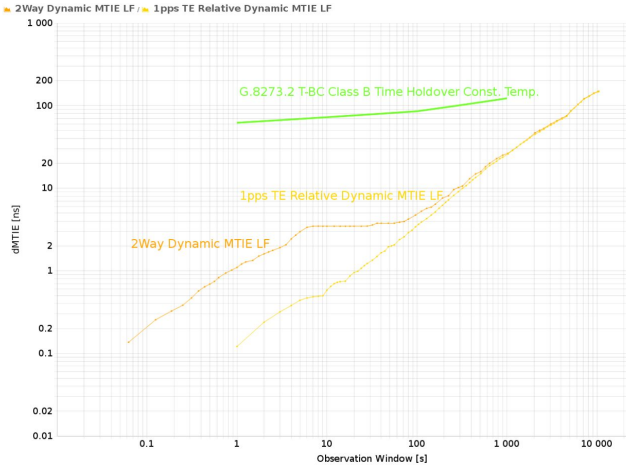
Red Hat

Holdover Performance – SyncE HW (“good” OSC+DPLL)

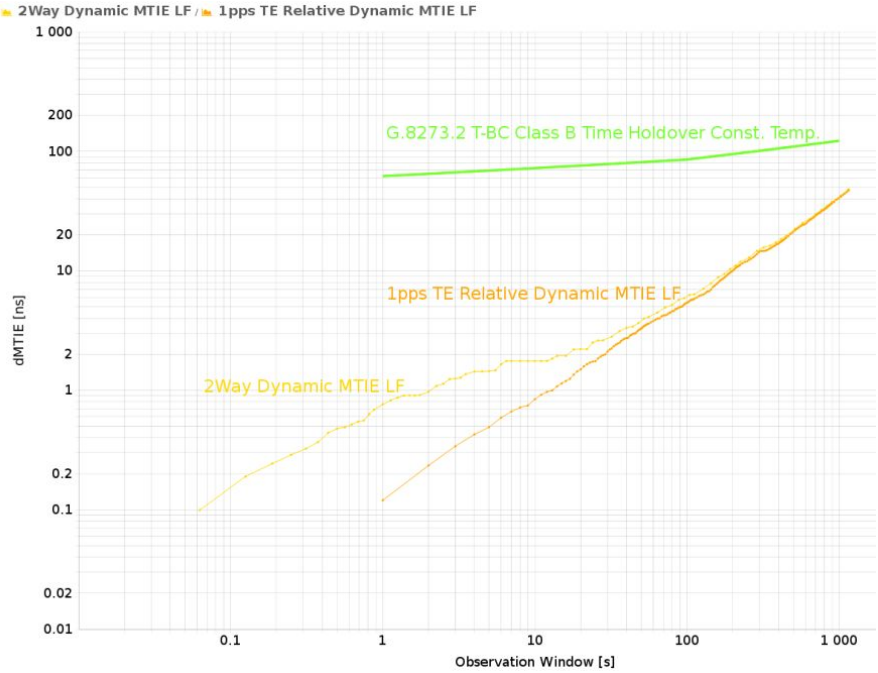
A: Holdover (dMTIE), w/ SyncE



B: Holdover (dMTIE), SyncE lost



C: Holdover (dMTIE), Freq FreeRUN - no SyncE lock/use at all



- Mask (above) is for Class-B, Class C/D is FFS
- ~constant temperature (lab !) tests ONLY done at Red Hat
- Case A: SyncE assisted holdover: ~6ns @1000sec
- Case B: SyncE lost holdover: ~25ns @1000 sec
- Case C: FreeRun, No SyncE: ~47ns @1000 sec

Red Hat

Are we there yet ?

- **Performance looks good, expected to meet the target requirements for RAN cases**

- **Need to finalize, implement and incorporate the new DPLL kernel Interface upstream**
  - Gating dependency for updates on NIC driver and firmware
  - Gating dependency on SyncE - ESMC protocol & control
  - Both drivers and synce4l need updates to implement / utilize the “final” interface
  - Dependency for “proper” upstreaming of both kernel/driver and userspace parts

- **On the meantime, we will continue progressing work with OOT drivers in parallel to progress things**
  - The presented set of baseline tests was conducted with interim OOT device driver

- **Still on our ToDo list:**
  - More testing of multi-card configurations (1..n cards, i.e. beyond 2 cards TE baseline presented)
  - Pass all of the relevant tests on combined (i.e. multi-card) configurations
  - Full 1PPS signals testing (in/out), including the applicable performance/conformance tests
  - Full 10Mhz signals testing (in/out), including the applicable performance/conformance tests
  - DPLL parameter tuning
  - Automation of all required state transitions
  - Complete set of failure / exception scenario handling & tests
  - Complete performance characterization of GNSS/T-GM with and without SyncE enabled

- **Longer term, we would prefer community “approved” solution for SyncE**

Get Involved & Get In Sync

Upstream projects - SW & open HW

- Chrony (primarily NTP)
- LinuxPTP project
- Linux kernel - common sync if/s
- Linux HW dev. Drivers - sync features
- OpenCompute TAP
- OpenCompute Networking
- OpenCompute Telco
- OpenCompute Telco Edge
- TIP OOPT
- TIP RAN projects

Key Standards / Spec. Organizations

- O-RAN WG 4,5,6,7,8 and 9...
- ITU SG15 Q13/15
- IEEE P802 / 802.3cx (TS accuracy)
- IEEE P1588
- 3gpp

How to contact us @Red Hat

Timo Jokiaho: tjokiaho@redhat.com

Pasi Vaananen: pvaanane@redhat.com

Special Thanks to

- LinuxPTP community, Linux Kernel Community, our NIC Si/card & FPGA Si/card HW partners, vRAN SW partners, Calnex & Spirent & Keysight

Red Hat

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.


linkedin.com/company/red-hat


youtube.com/user/RedHatVideos


facebook.com/redhatinc


twitter.com/RedHat

Red Hat