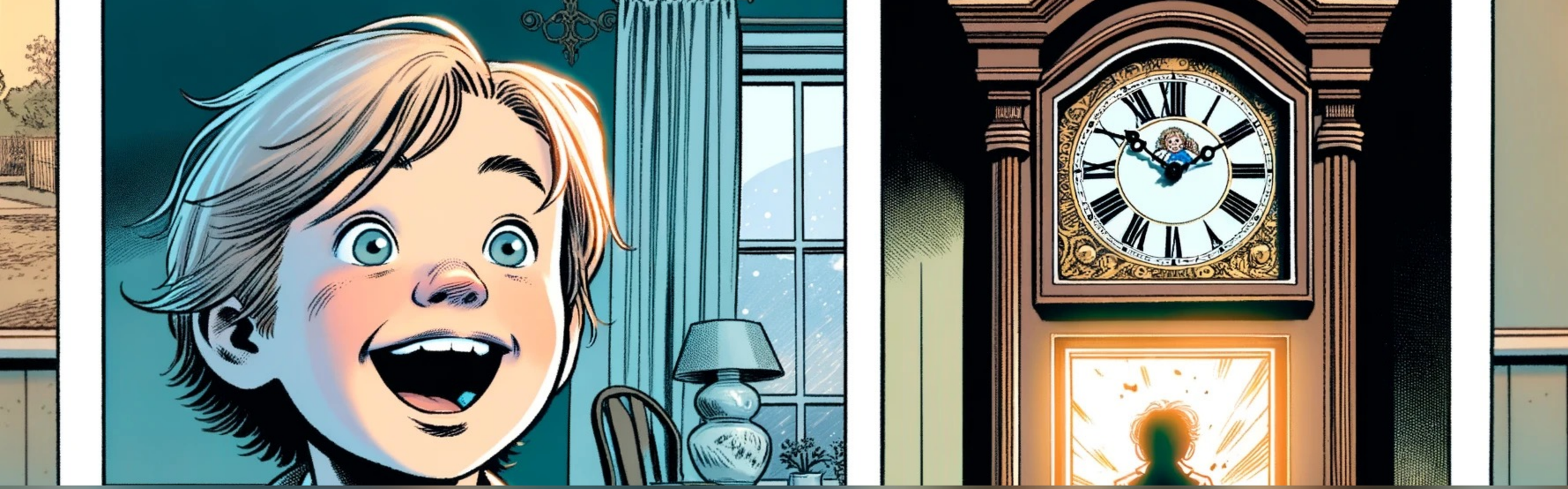
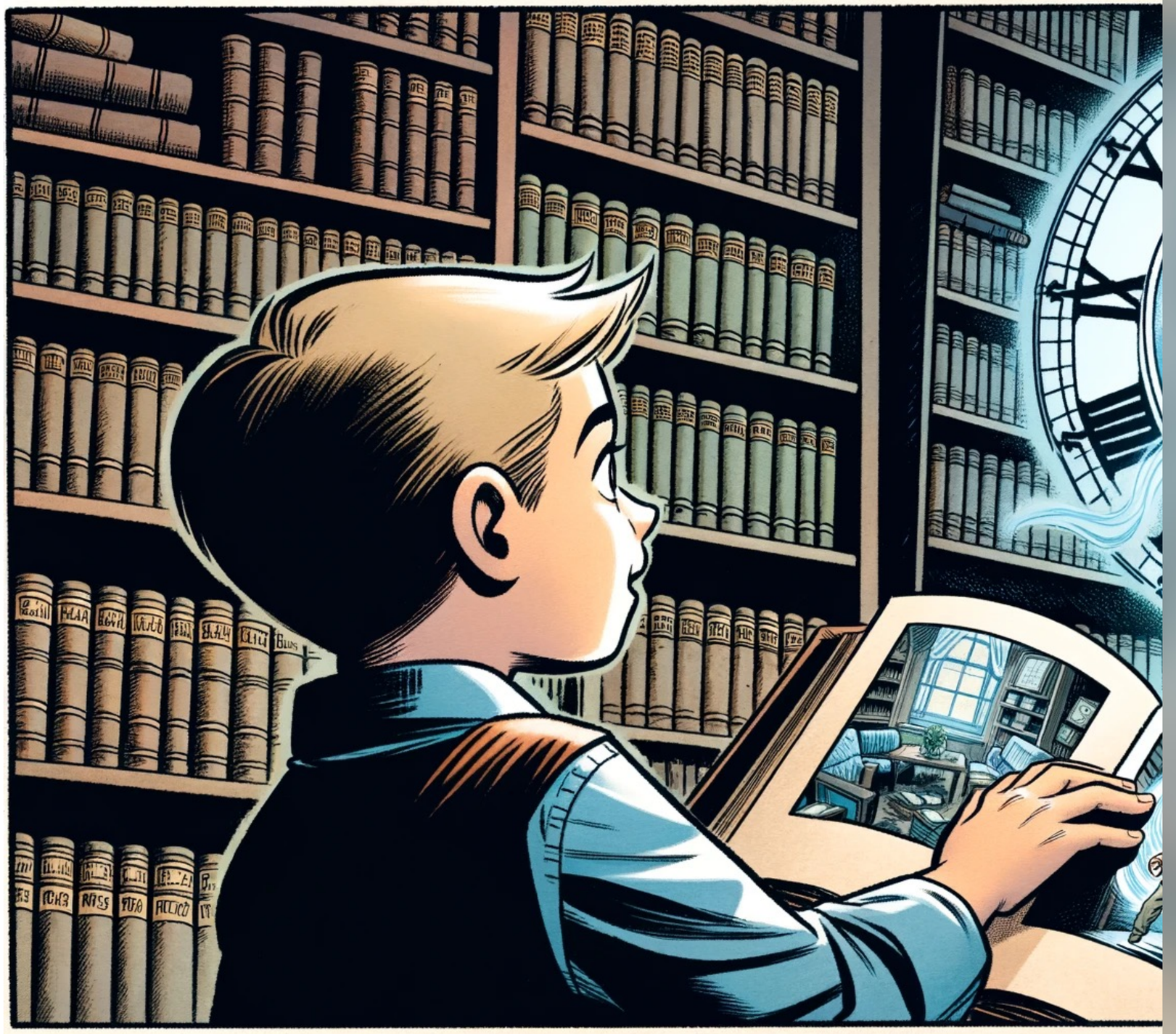


Introduction to time synchronization

Maciek Machnikowski | Netdev 0x17

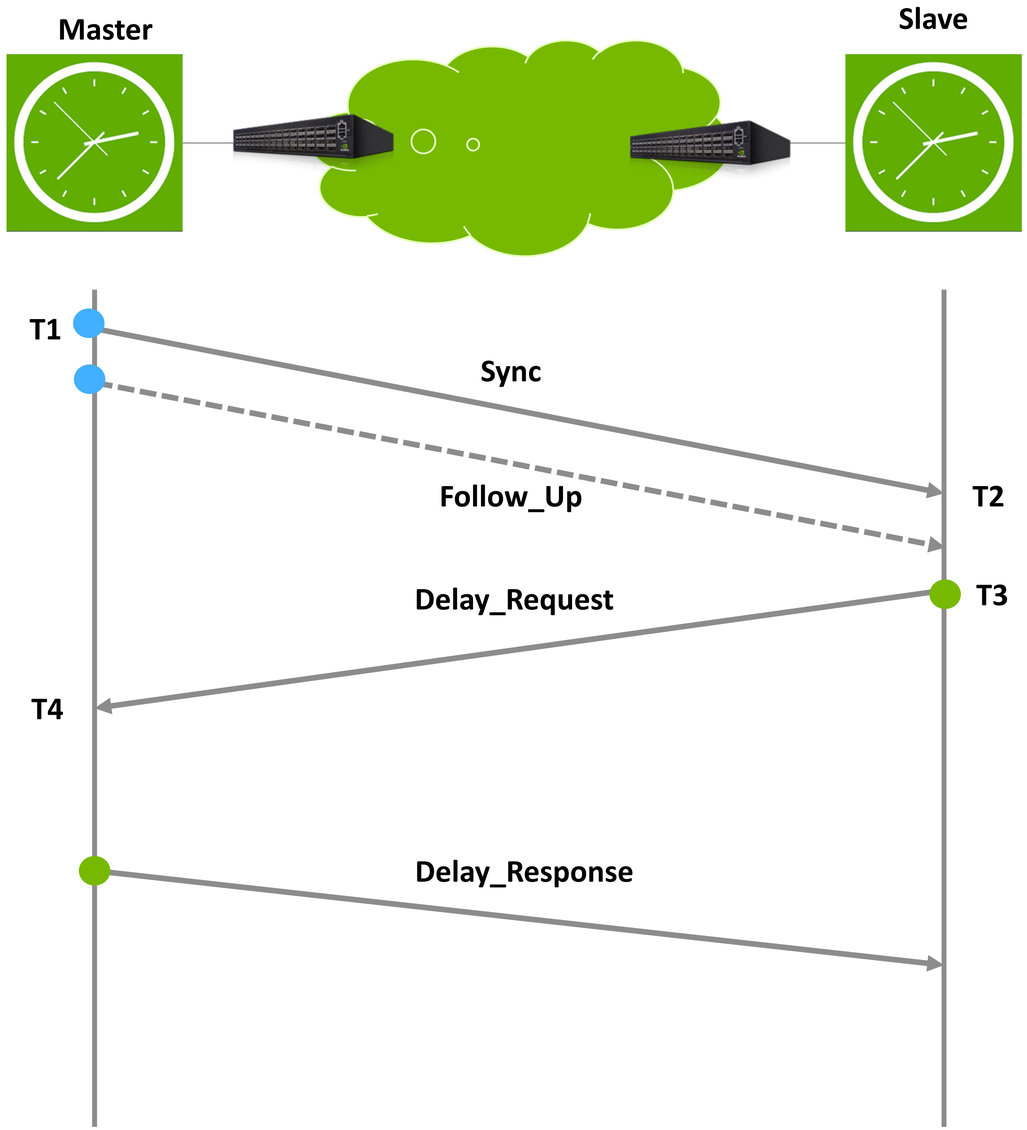
The Plan:

- Rulebook: IEEE 1588 Profiles
- The final frontier: Precise Time Measurement (PTM)
- Monitoring and troubleshooting
- Linuxptp 4.x
- PTP and NTP

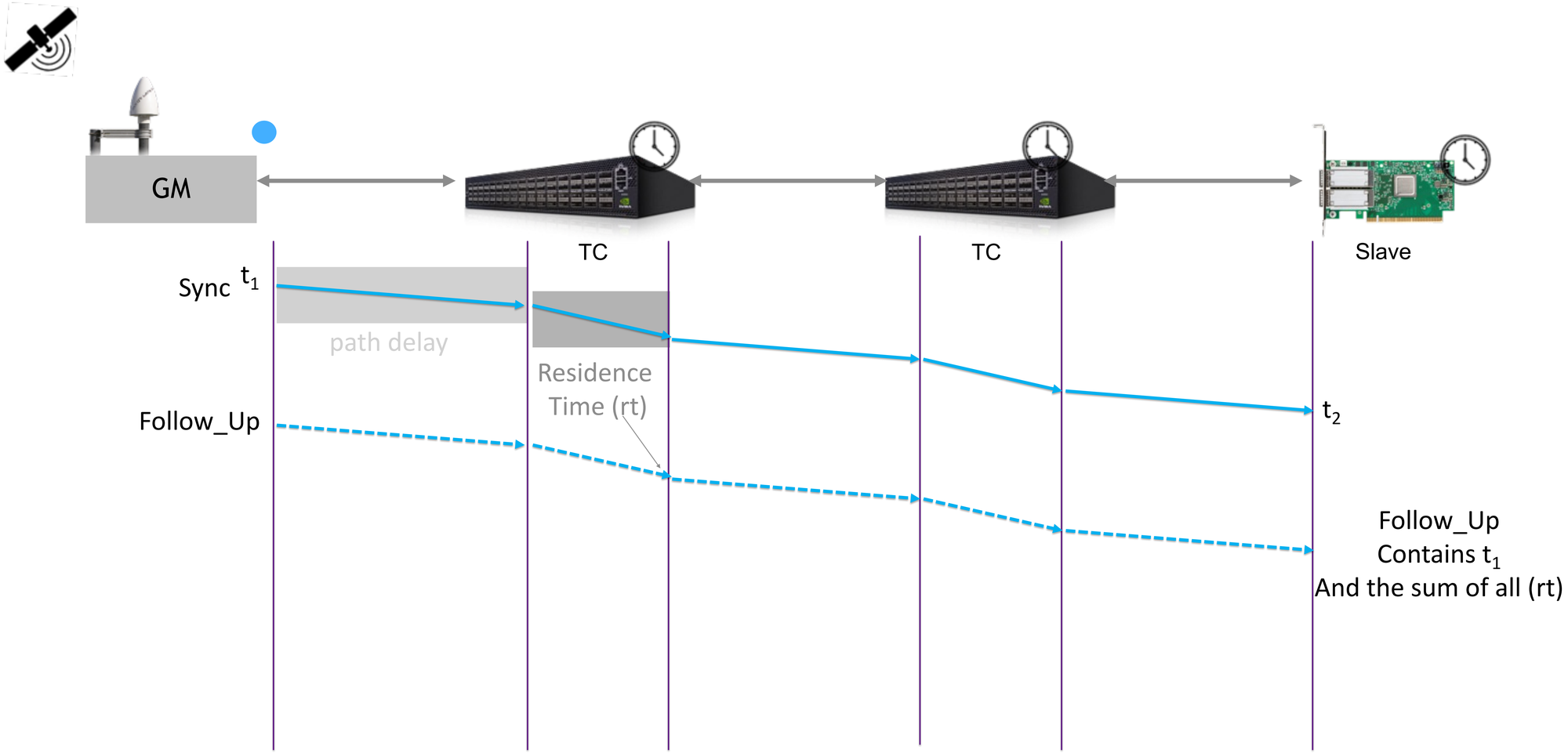


Rulebook:
IEEE 1588 profiles

How PTP (E2E) works?

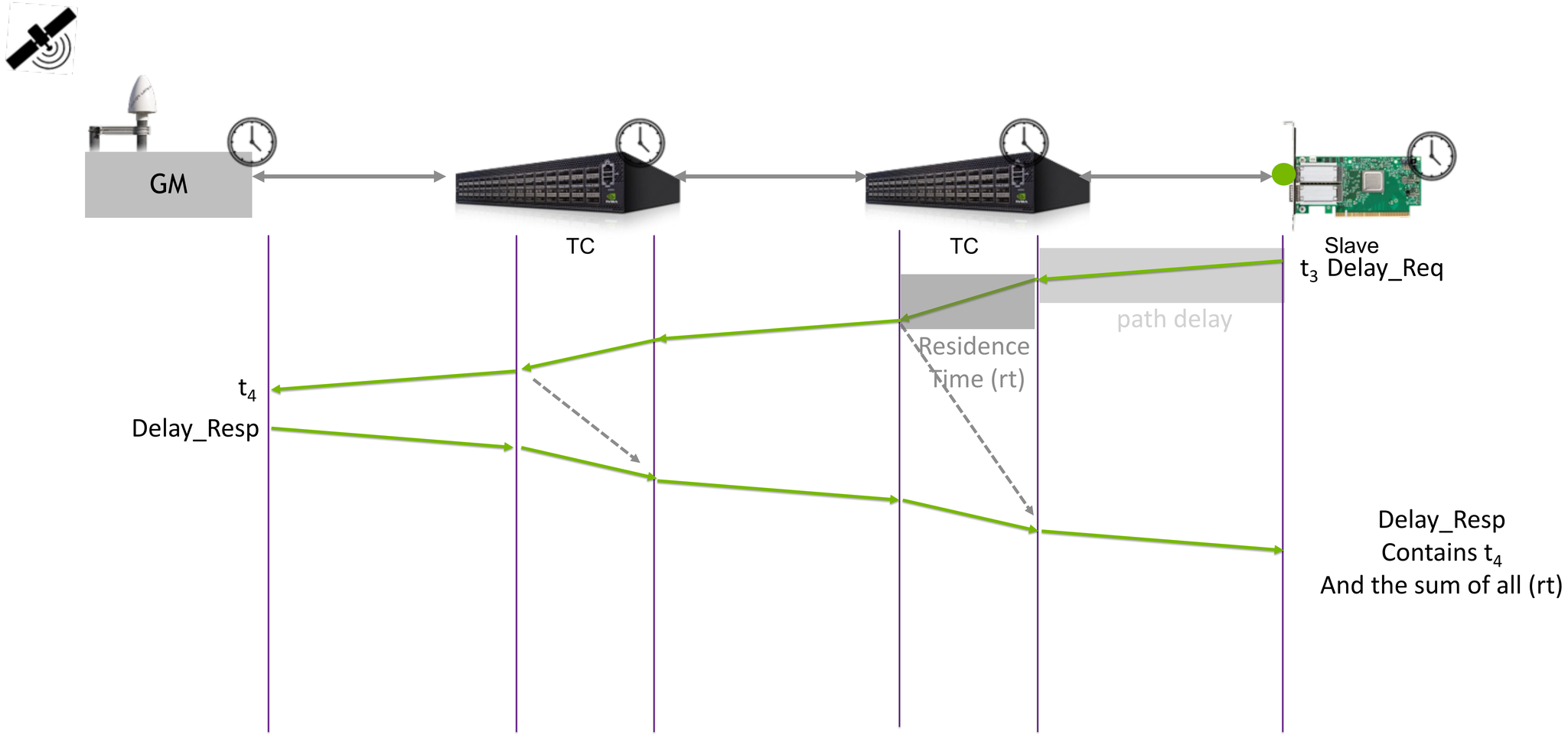


E2E Sync



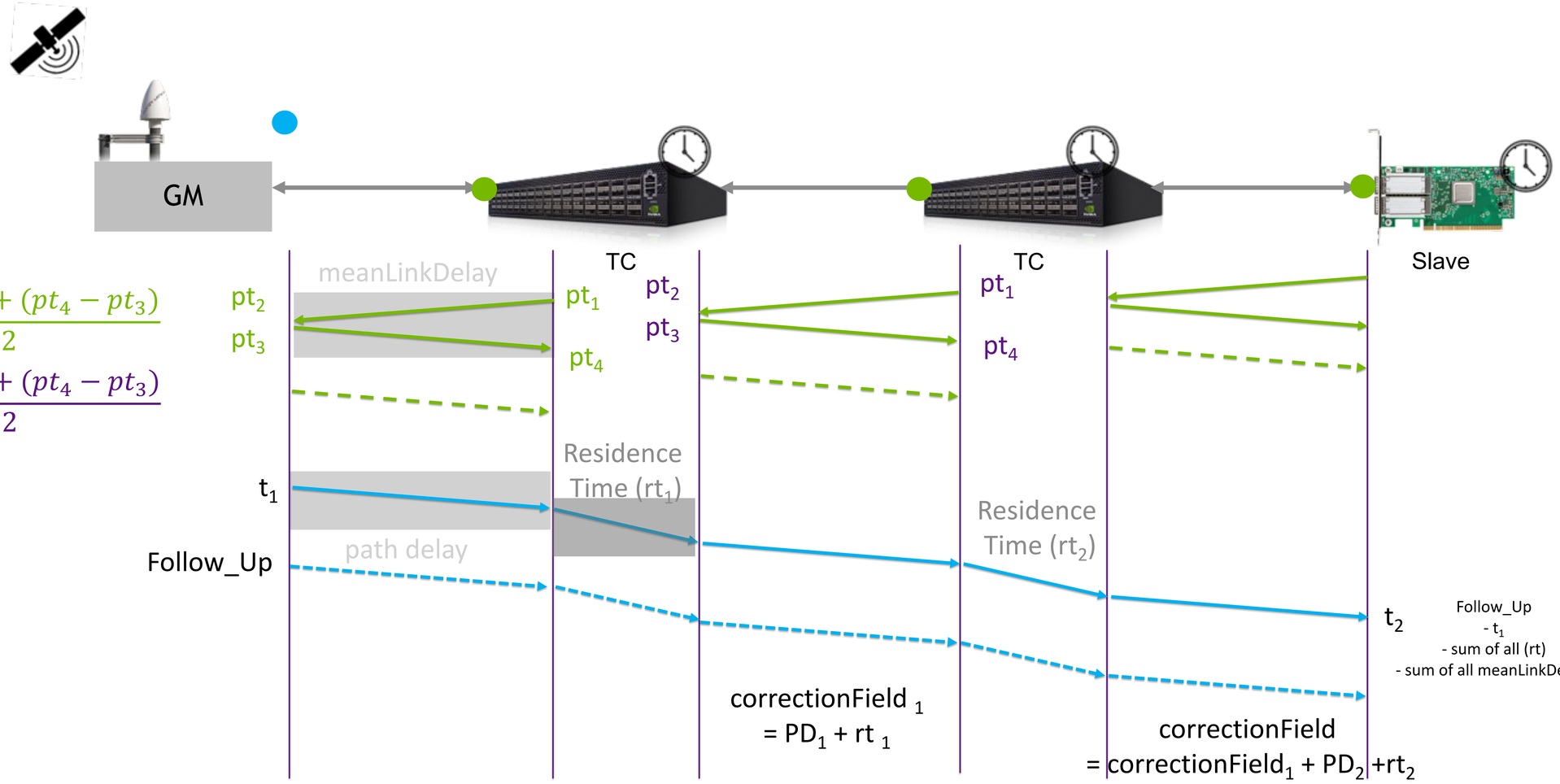
PTP packets : Sync, Announce

E2E Delay



Delay_Req, Delay_Resp

P2P



$$PD_1 = \frac{(pt_2 - pt_1) + (pt_4 - pt_3)}{2}$$

$$PD_2 = \frac{(pt_2 - pt_1) + (pt_4 - pt_3)}{2}$$

- PTP P2P packets (Peer_Delay_req/Res)
- PTP Eth' Multicast Sync and announce packets

TC vs BC

Transparent clock

- Enables fast GM changes
- Not much config is required

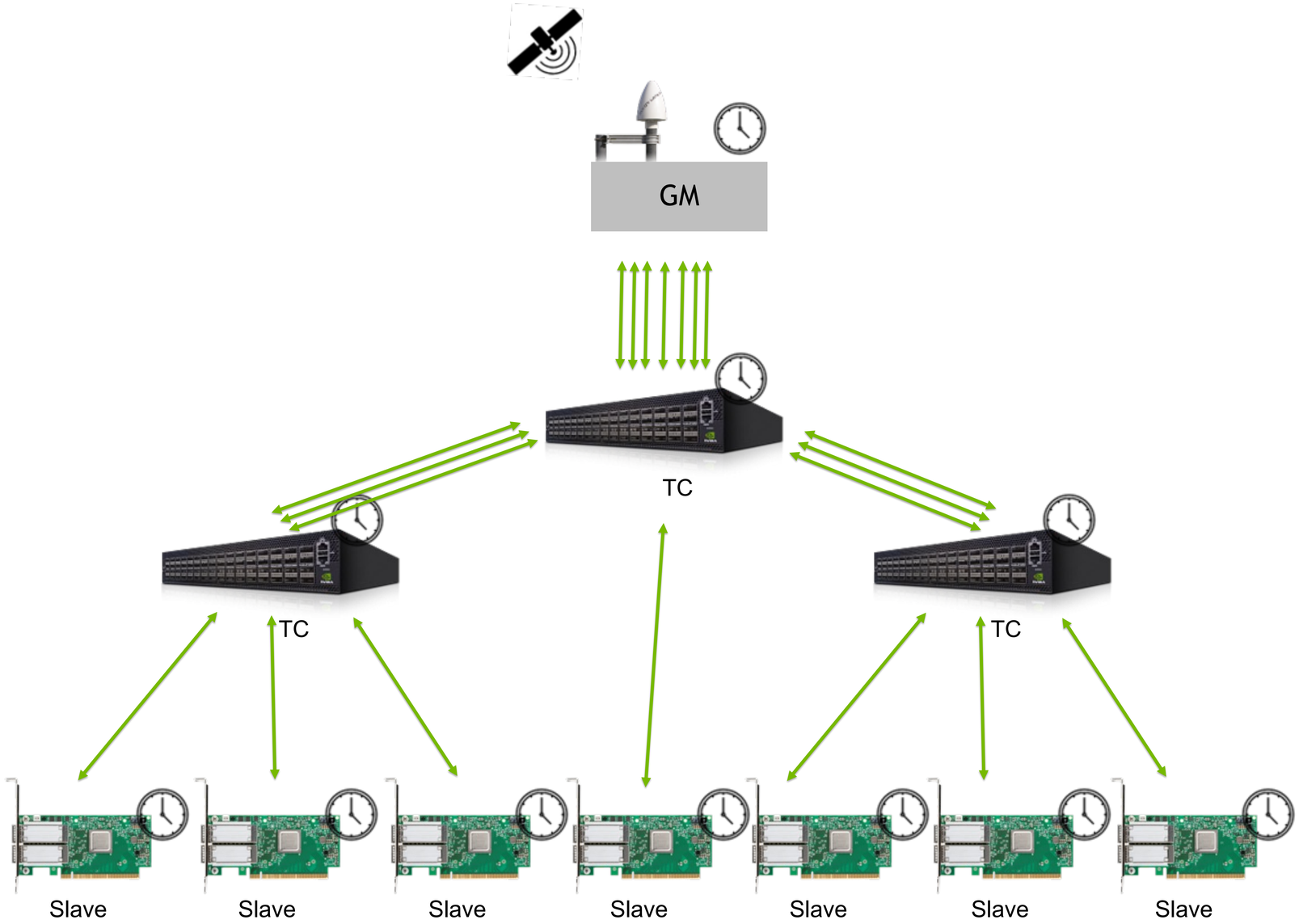
- Scalability issues (all packets traverse to the root node)

Boundary clock

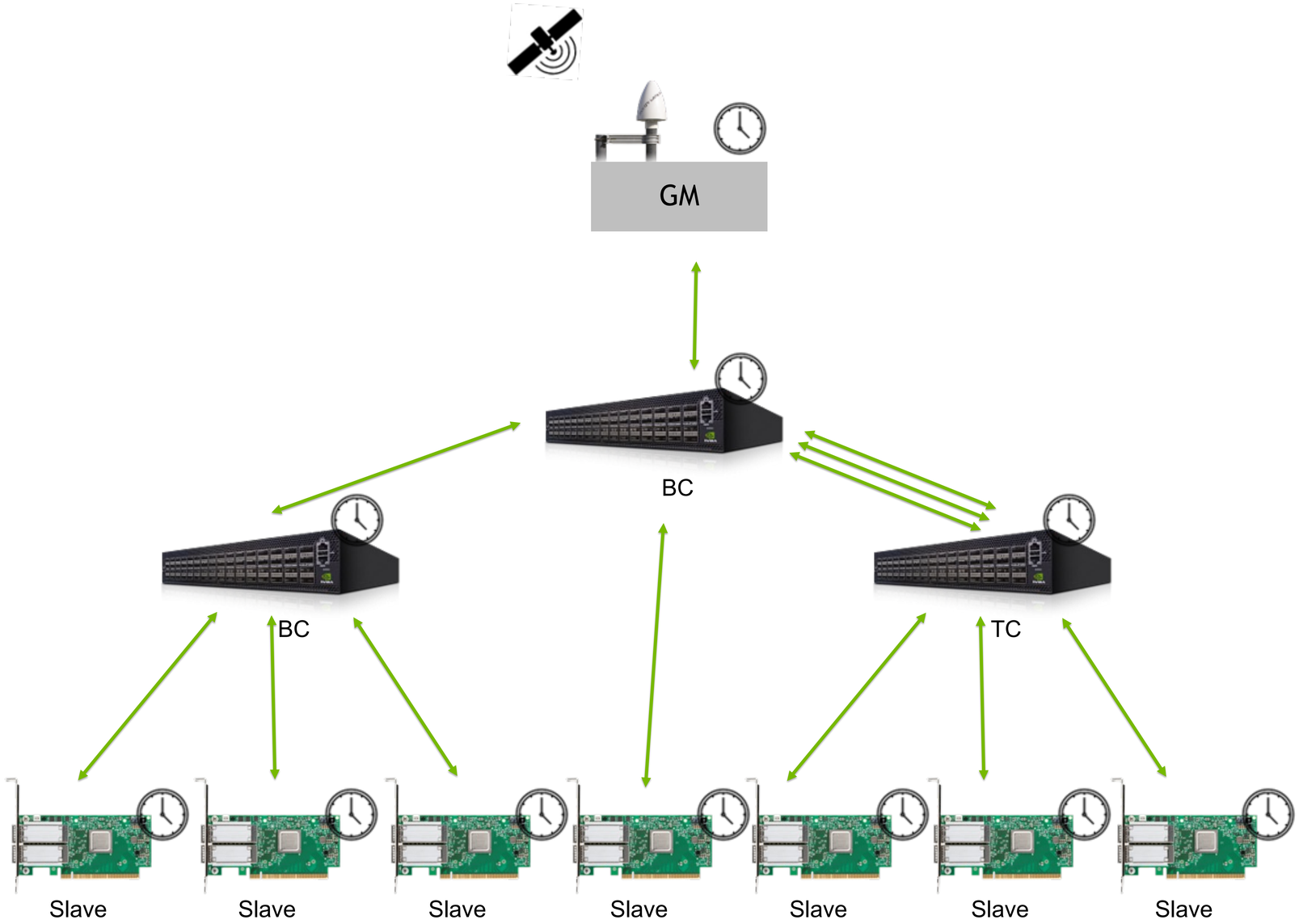
- Runs a time servo
- Breaks up PTP message domains
  - Terminates messages from a GM
- Shields Slaves from transients on hierarchy changes
- Provides backup time source

- Adds wander error
- Cumulates error

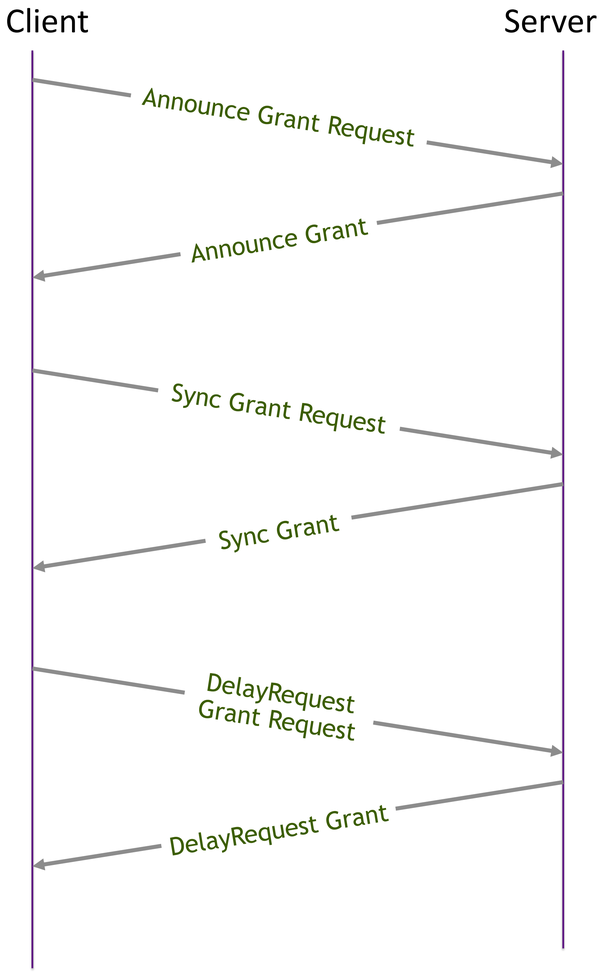
TC vs BC

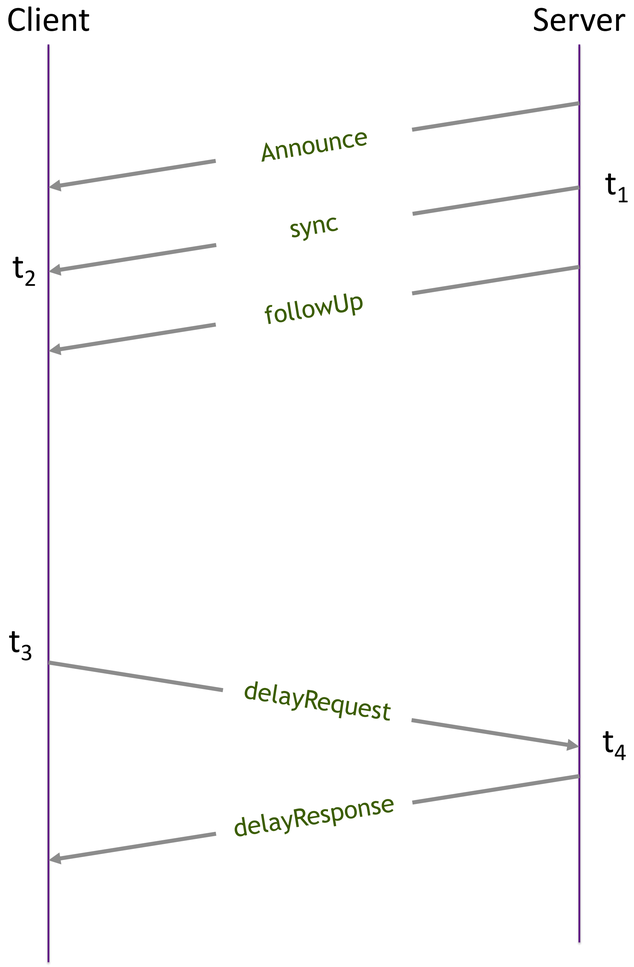


TC vs BC



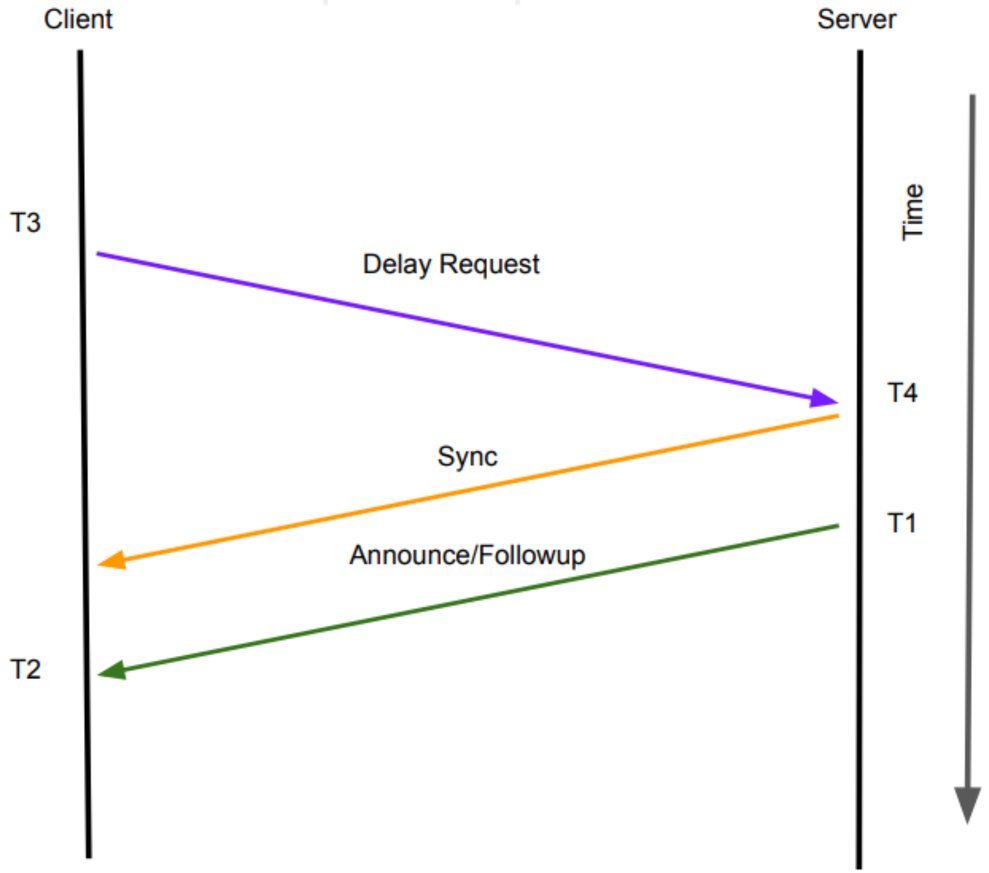
PTP Unicast messaging





SPTP

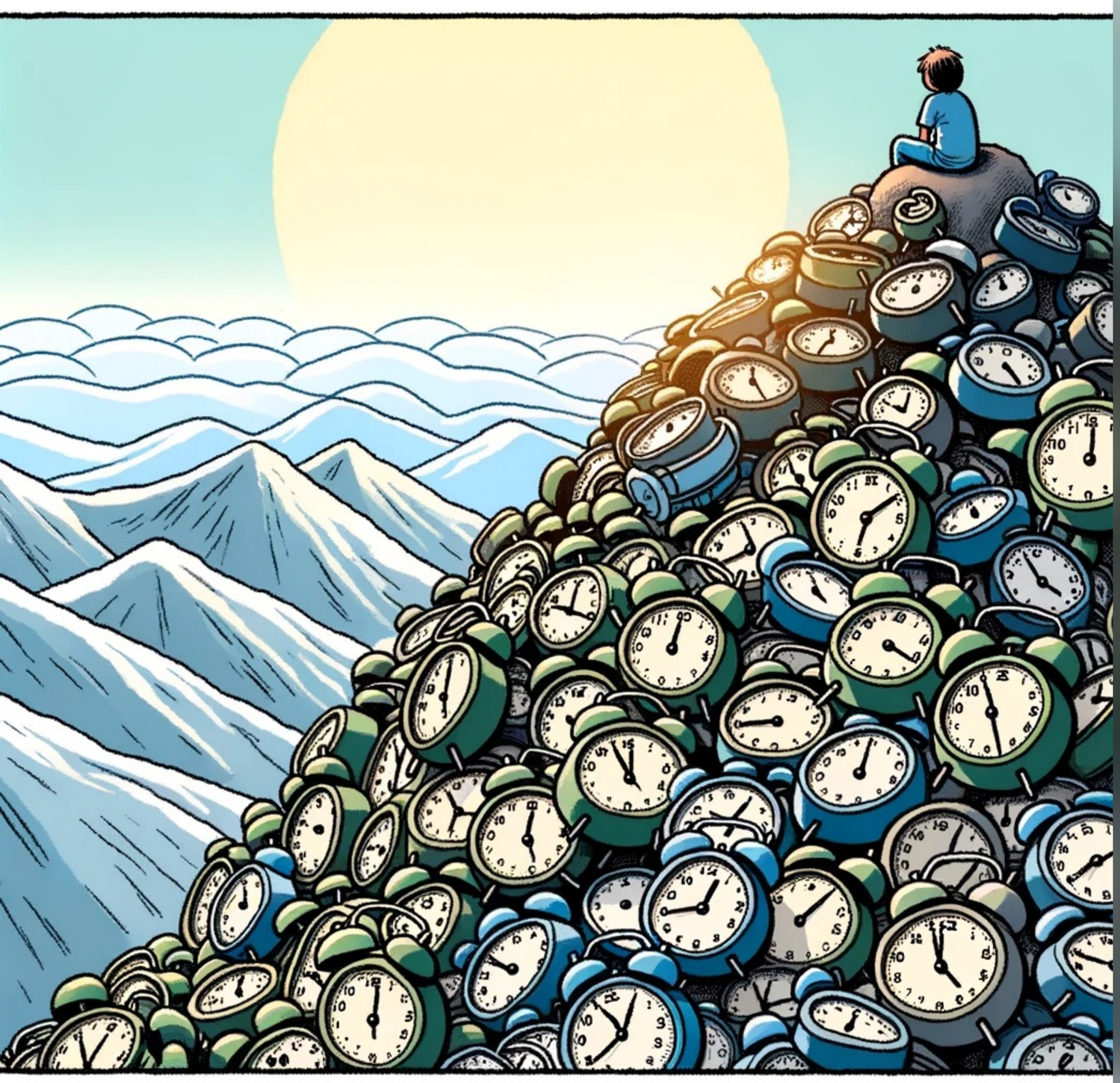
- Client sends a Delay Request
- Server responds with a Sync
- Server sends FollowUp and Announce



github.com/facebook/time/tree/main/ptp/sptp

PTP Profiles Comparison

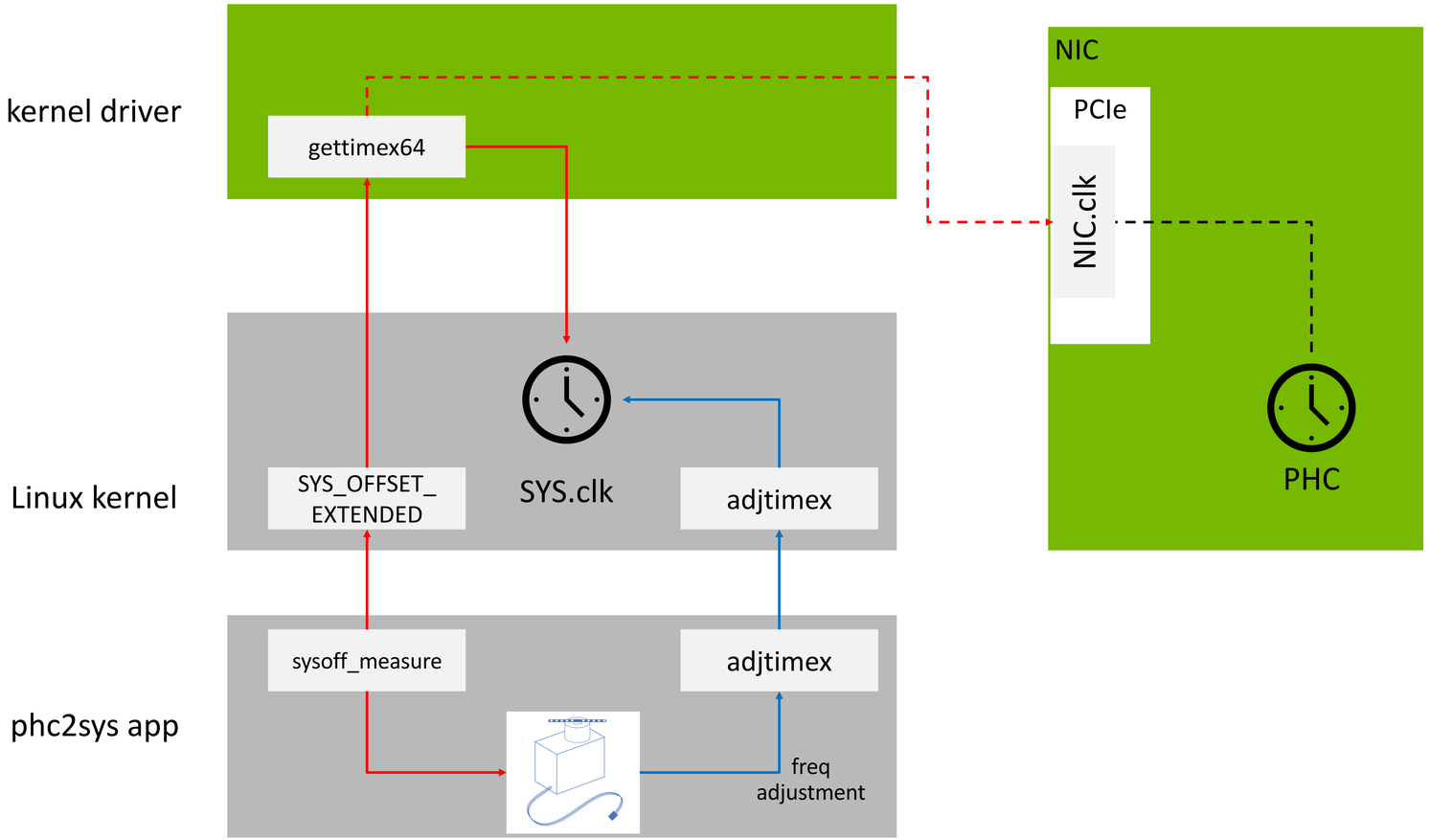
	Default	Broadcast	Enterprise /Financial	Data centers	Telecom			Power	A/V bridged /TSN
Defining body	IEEE	SMPTE	IETF	OCP	ITU-T	ITU-T	ITU-T	IEEE	IEEE
PTP profile	IEEE1588 Default profiles	SMTPE ST- 2059-2	Enterprise profile IETF (draft)	Data Center PTP Profile	G.8265.1	G.8275.2	G.8275.1	C37. 238	802. 1 AS (gPTP)
Time error (vs UTC)	-	Typically sub 1usec	Typically sub 1usec	2.5usec	16ppb	1.1usec	1.1usec	1usec	1usec
Transport mapping	UDP or L2	UDP/IP	UDP/IP	UDP/IP	UDP/IP	UDP/IP	L2	L2	L2
communication mode	Multicast (M) or Unicast (U)	M or U	Hybrid, M is permitted	U	U	U	M (local link)	M	M
Delay measurement	E2E or P2P	E2E	E2E only	E2E only	E2E only	E2E only	E2E only	P2P only	P2P only
PTP full support	Not mandatory	Not mandatory	No or partial	Yes (TC's)	No	No or partial	Yes	Yes	Yes, strictly gPTP only
Physic. level syntonization	Option	Option	Not defined	No	Option	SyncE or GPS	Mandatory (SyncE)	Option	No
Domain's time-scale	PTP/ARB	SMPTE (ARB opt.)	PTP	PTP	PTP	PTP	PTP	PTP	PTP
Best master clock algorithm (BMCA)	Default or Alternate	Default or Alternate	Default, but multiple GM is permitted	Alternative (active standby)	Not used	Alternative	Alternative	Default	Yes**



The final frontier: Precision Time Measurement (PTM)

PHC2SYS: sync NIC -> CPU clk (no PTM)

measurement path
adjustment path
standard interface
internal interface



SYS_OFFSET_EXTENDED/gettimex64:

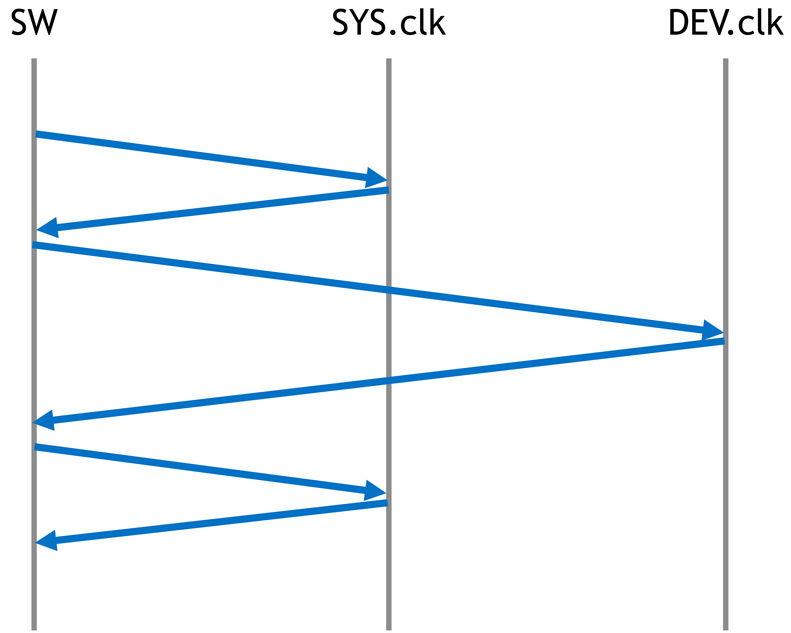
```
for i in 0 … n_samples:
    SYS_pre = ktime_get_real_ts64
    NIC_l = read(NIC.clk.l)
    SYS_post = ktime_get_real_ts64
```

In phc2sys app:

```
// phc2sys:
// use the best sample
// i.e. min SYS_post-SYS_pre
```

device <> CPU time offset measurement

today



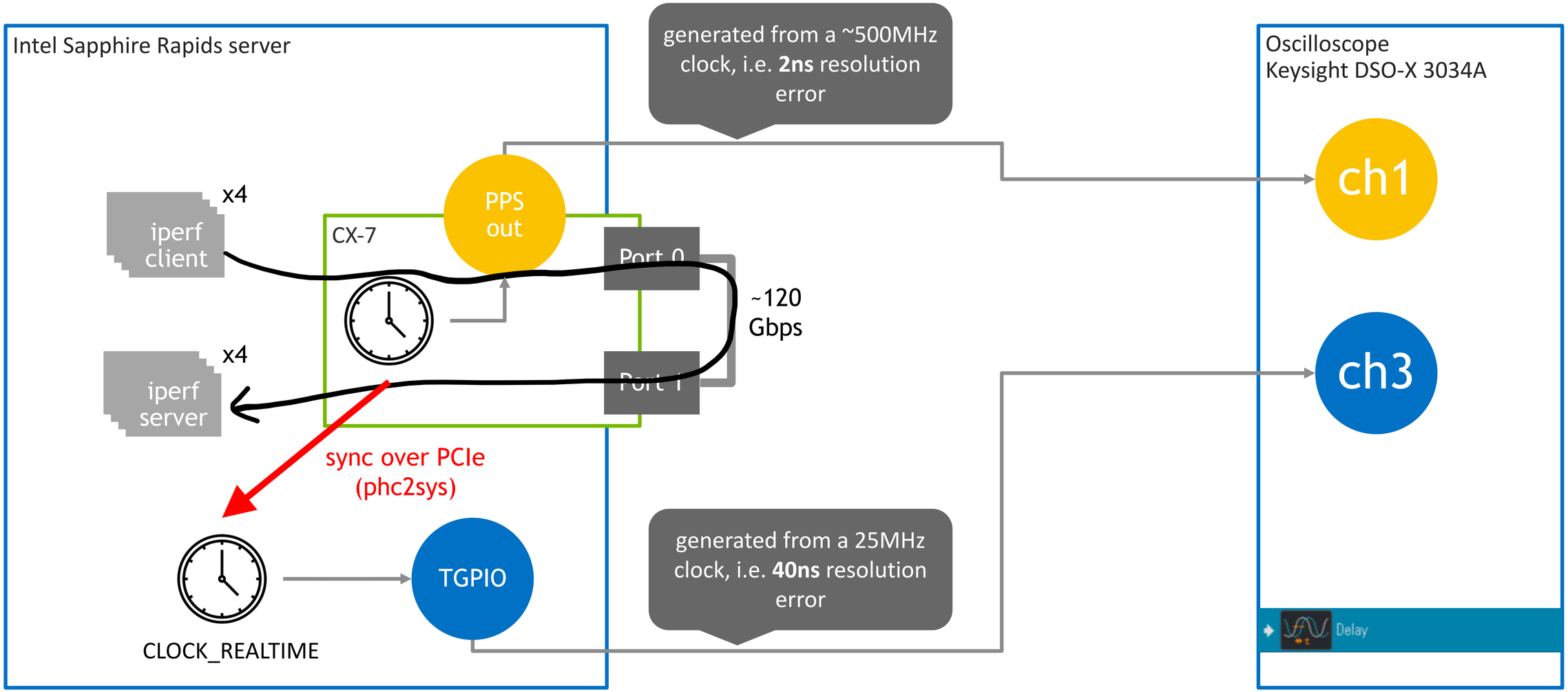

```
SYS_pre  = ktime_get_real_ts64
DEV      = read(DEV.clk)
SYS_post = ktime_get_real_ts64

SYS    = (SYS_pre + SYS_post) / 2
DEV_SYS_offset = (DEV – SYS)
```

More like…

How (bad) are things, actually?

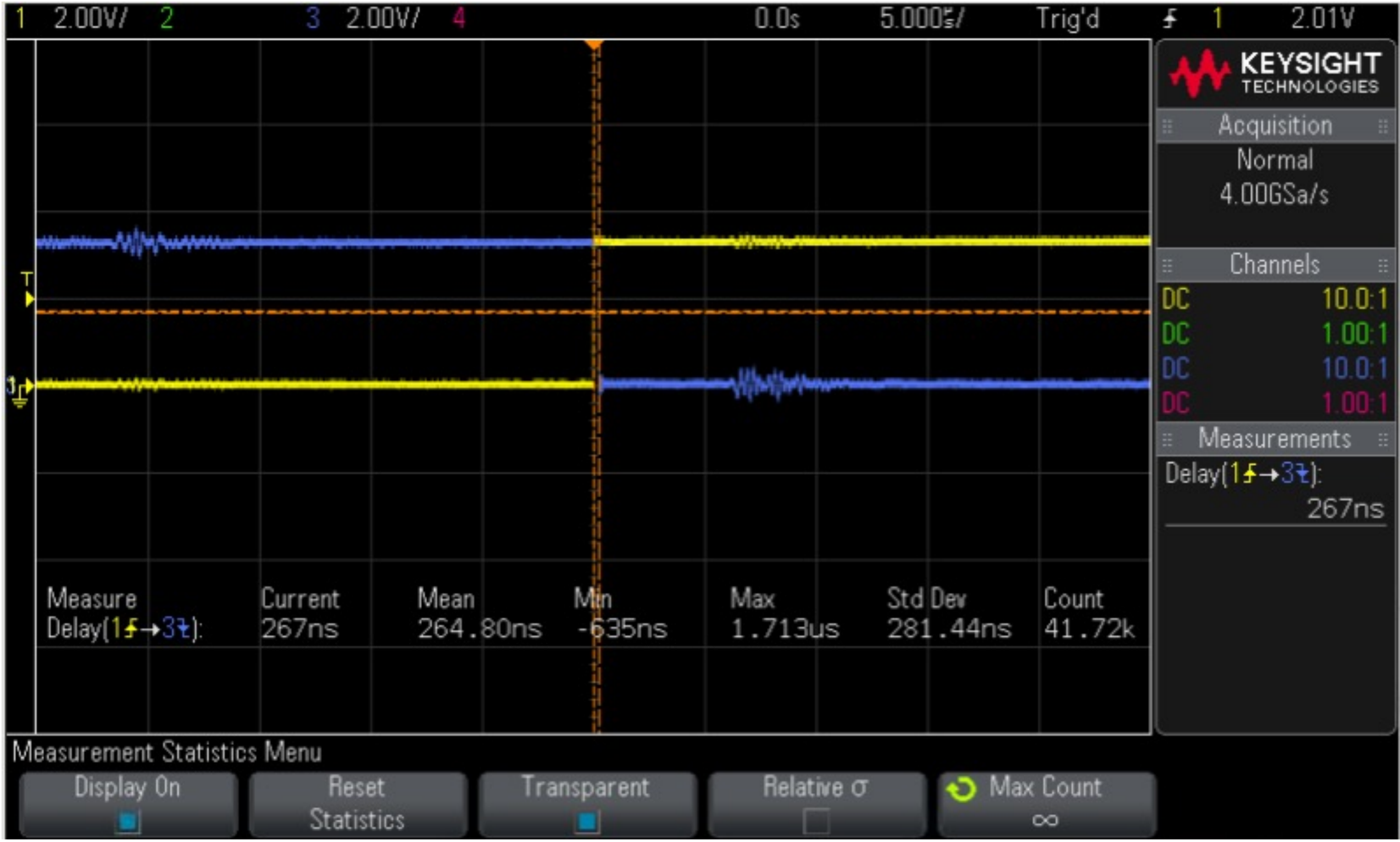
test platform



NIC → CPU sync results

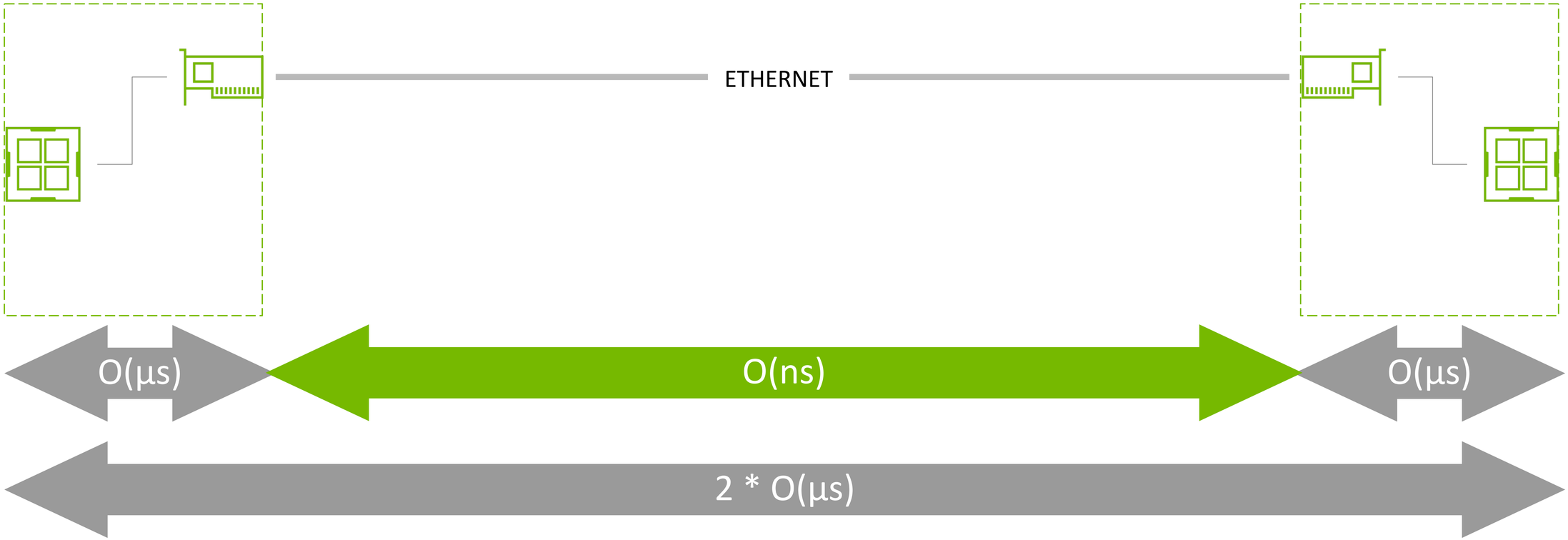
11-hour run, state-of-the-art

- iperf-loaded system CPU/NIC time error:
  - min: -635.0 [ns]
  - max: 1713.0 [ns]
  - pk-pk: 2348.0 [ns]
  - stddev: 281.4 [ns]



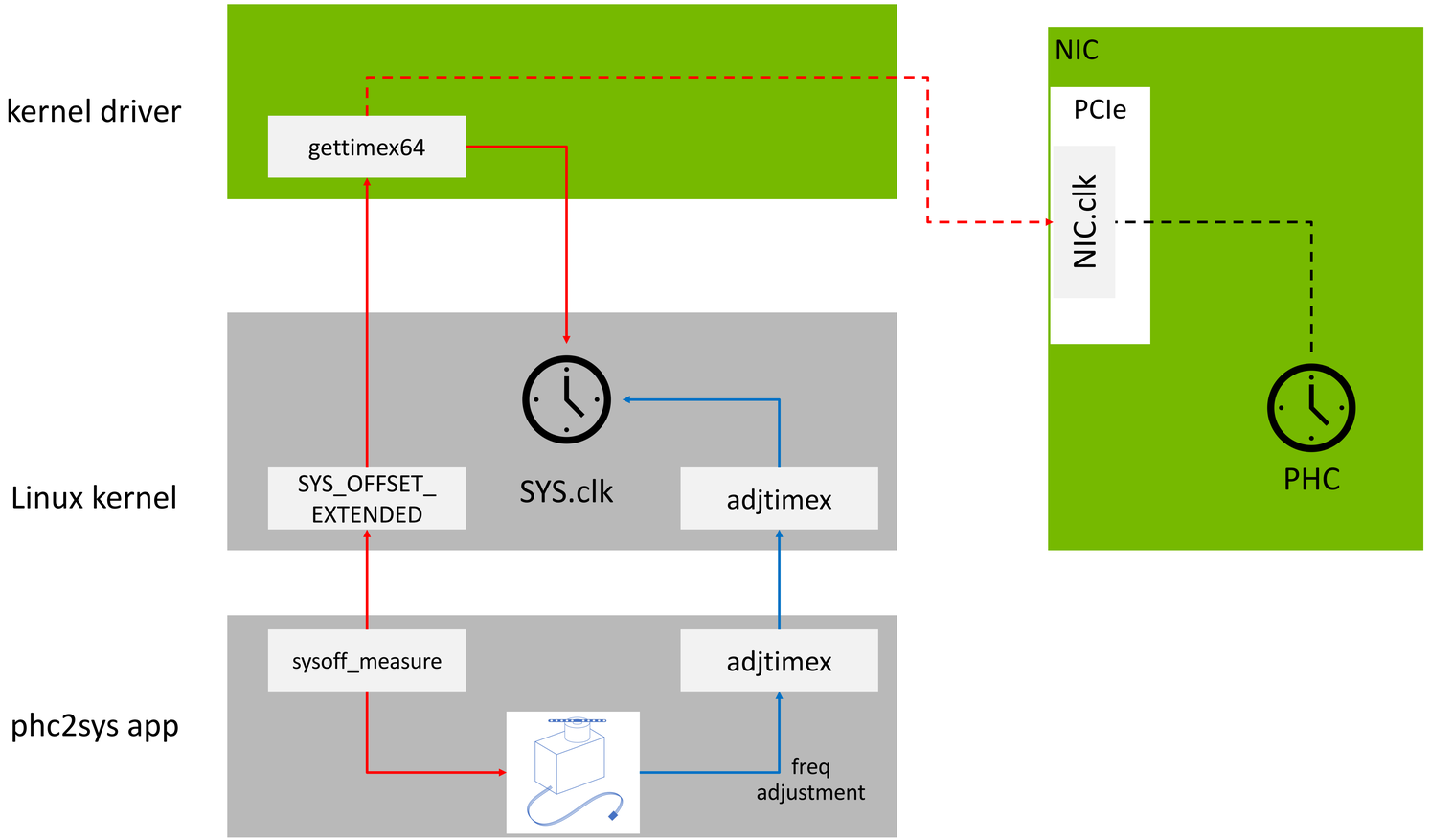
State of the art in timesync

A weird situation...



PHC2SYS: sync NIC -> CPU clk (no PTM)

measurement path
adjustment path
standard interface
internal interface



SYS_OFFSET_EXTENDED/gettimex64:

```
for i in 0 … n_samples:
    SYS_pre = ktime_get_real_ts64
    NIC_l = read(NIC.clk.l)
    SYS_post = ktime_get_real_ts64
```

In phc2sys app:

```
// phc2sys:
// use the best sample
// i.e. min SYS_post-SYS_pre
```

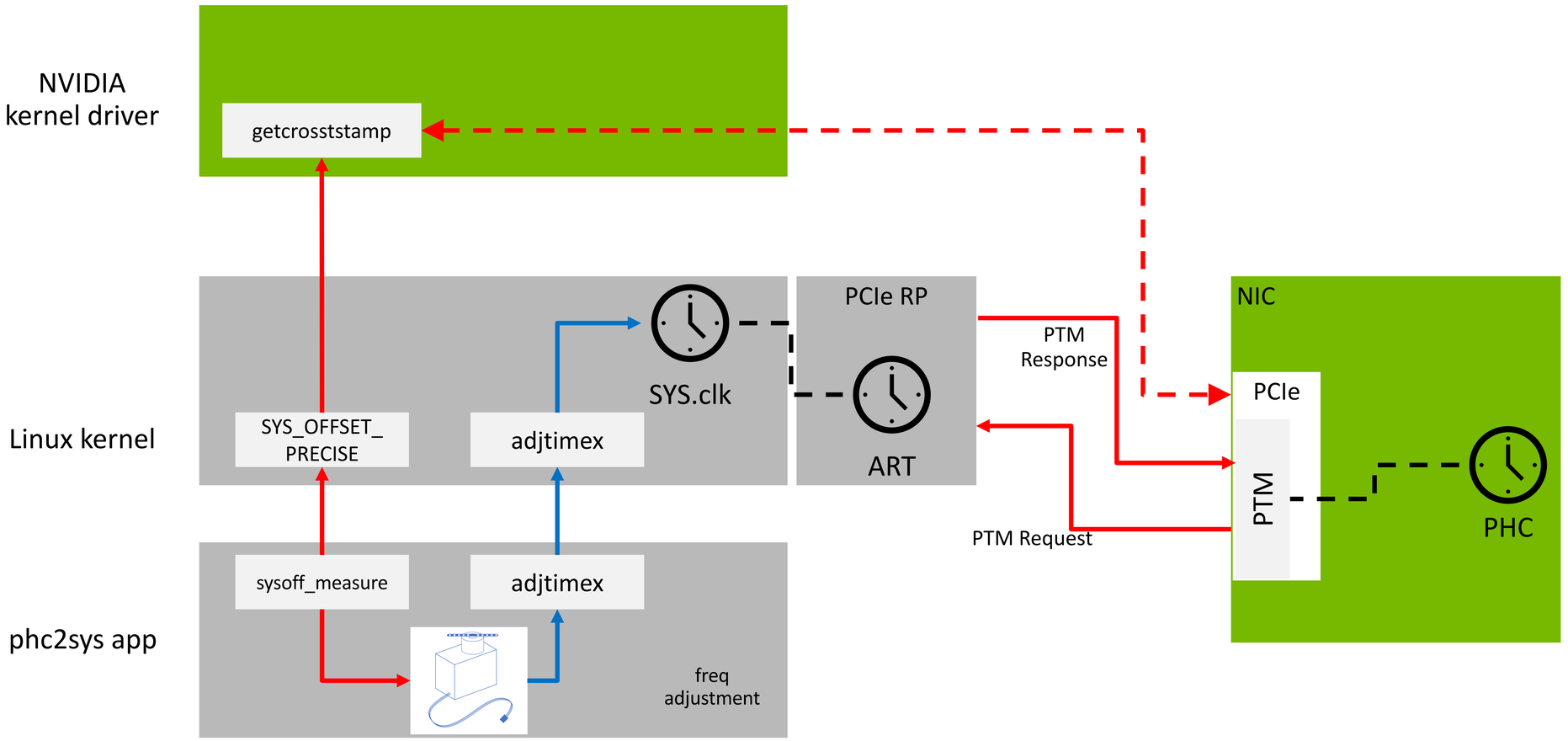
PHC2SYS: sync NIC -> CPU clk (w/ PTM)

measurement path

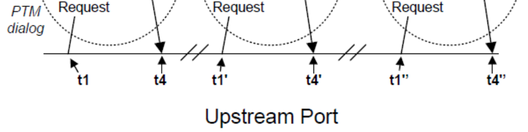
adjustment path

standard interface

internal interface



OCP TAP talk
on PCIe PTM:



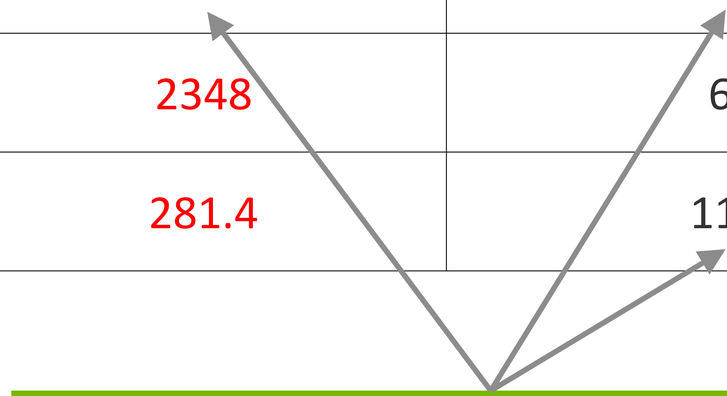
Upstream Port

$$T_{PTM}(t_{1'}) = t_{2'} - \frac{(t_4 - t_1) - (t_3 - t_2)}{2}$$

NVIDIA

Results comparison

Scenario	Metric	legacy method	PCIe PTM
Empty system (~1 hour)	peak-to-peak (max-min) [ns]	50	55
	std dev [ns]	11.2	11.3
Network load (>10 hours)	peak-to-peak (max-min) [ns]	2348	64
	std dev [ns]	281.4	11.9



PPS signal granularity: 42 [ns].

Standard deviation of a uniformly-distributed variable:

$$\frac{42}{\sqrt{12}} \cong 12\ [ns]$$

Monitoring and Troubleshooting

troubleshooting

- Profile
- Domain Value
- Message Rates
- Communication Mode
- Firewall
- Tcpdump/wireshark

- Grandmaster ID

phc_ctl

- Included in the **linuxptp** project

- set [seconds] set PHC time (defaults to time on CLOCK_REALTIME, does not respect UTC-TAI offset)
- get get PHC time
- adj <seconds> adjust PHC time by offset
- freq [ppb] adjust PHC frequency (default returns current offset)
- cmp compare PHC offset to CLOCK_REALTIME
- caps display device capabilities (default if no command given)
- wait <seconds> pause between commands

phc_ctl

- Read the current clock time from the device
  - phc_ctl eth0 get
- Set the PHC clock time to CLOCK_REALTIME
  - phc_ctl eth0 set
- Set PHC clock time to 0 (seconds since Epoch)
  - phc_ctl eth0 set 0.0
- Quickly sanity check frequency slewing by setting slewing frequency by positive 10%, resetting clock to 0.0 time, waiting for 10 seconds, and then reading time. The time read back should be (roughly) 11 seconds, since the clock was slewed 10% faster.
  - phc_ctl eth0 freq 100000000 set 0.0 wait 10.0 get

testptp

- Included with the Linux kernel
- linux/tools/testing/selftests/ptp/
- Bound to the APIs available in a given kernel

- -c query the ptp clock's capabilities
- -g get the ptp clock time
- -k val measure the time offset between system and phc clock
  for 'val' times (Maximum 25)

testptp – clock control

- -n val shift the ptp clock time by 'val' nanoseconds
- -f val adjust the ptp clock frequency by 'val' ppb
- -t val shift the ptp clock time by 'val' seconds

- -s set the ptp clock time from the system time
- -S set the system time from the ptp clock time
- -T val set the ptp clock time to 'val' seconds
- -P val enable or disable (val=1|0) the system clock PPS

testptp – pin options

- -p val enable output with a period of 'val' nanoseconds
- -H val set output phase to 'val' nanoseconds (requires -p)
- -w val set output pulse width to 'val' nanoseconds (requires -p)
- -i val index (channel) for event/trigger
- -L pin,val configure pin index 'pin' with function 'val'

  the channel index is taken from the '-i' option

  'val' specifies the auxiliary function:

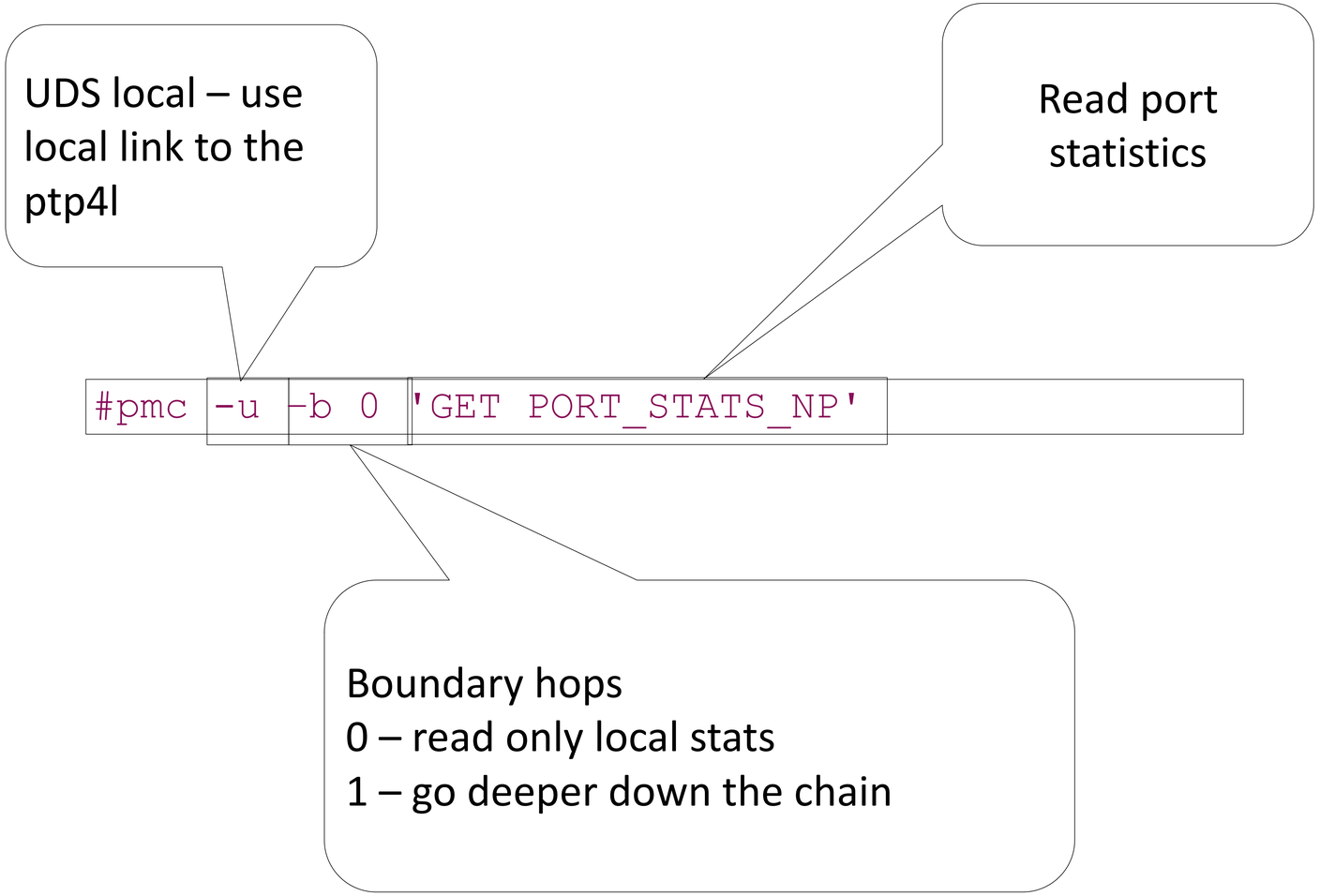
  0 - none

  1 - external time stamp

  2 - periodic output

- -l list the current pin configuration
- -e val read 'val' external time stamp events

pmc – command line



pmc

```
pmc -u -b 0 'GET TIME_STATUS_NP' -s /var/run/timemaster/ptp4l.0.socket
```

```
sending: GET TIME_STATUS_NP
  b49691.fffe.a599e6-0 seq 0 RESPONSE MANAGEMENT TIME_STATUS_NP
          master_offset              -62
          ingress_time               1657722750370354753
          cumulativeScaledRateOffset +0.000000000
          scaledLastGmPhaseChange    0
          gmTimeBaseIndicator        0
          lastGmPhaseChange          0x0000'0000000000000000.0000
          gmPresent                  true
          gmIdentity                 08c0eb.fffe.a6ee7d
```

pmc

```
pmc -u -b 0 'GET CURRENT_DATA_SET' -s /var/run/timemaster/ptp4l.0.socket
```

```
sending: GET CURRENT_DATA_SET
  b49691.fffe.a599e6-0 seq 0 RESPONSE MANAGEMENT CURRENT_DATA_SET
          stepsRemoved     1
          offsetFromMaster 40.0
          meanPathDelay    1597.0
```

Reading all timestamps

- sudo ptp4l -i eth0 -2 -m --tx_timestamp_timeout 100 --free_running 1 --slave_event_monitor /var/run/ts2phc-mon
- sudo pmc -u -i /var/run/ts2phc-mon

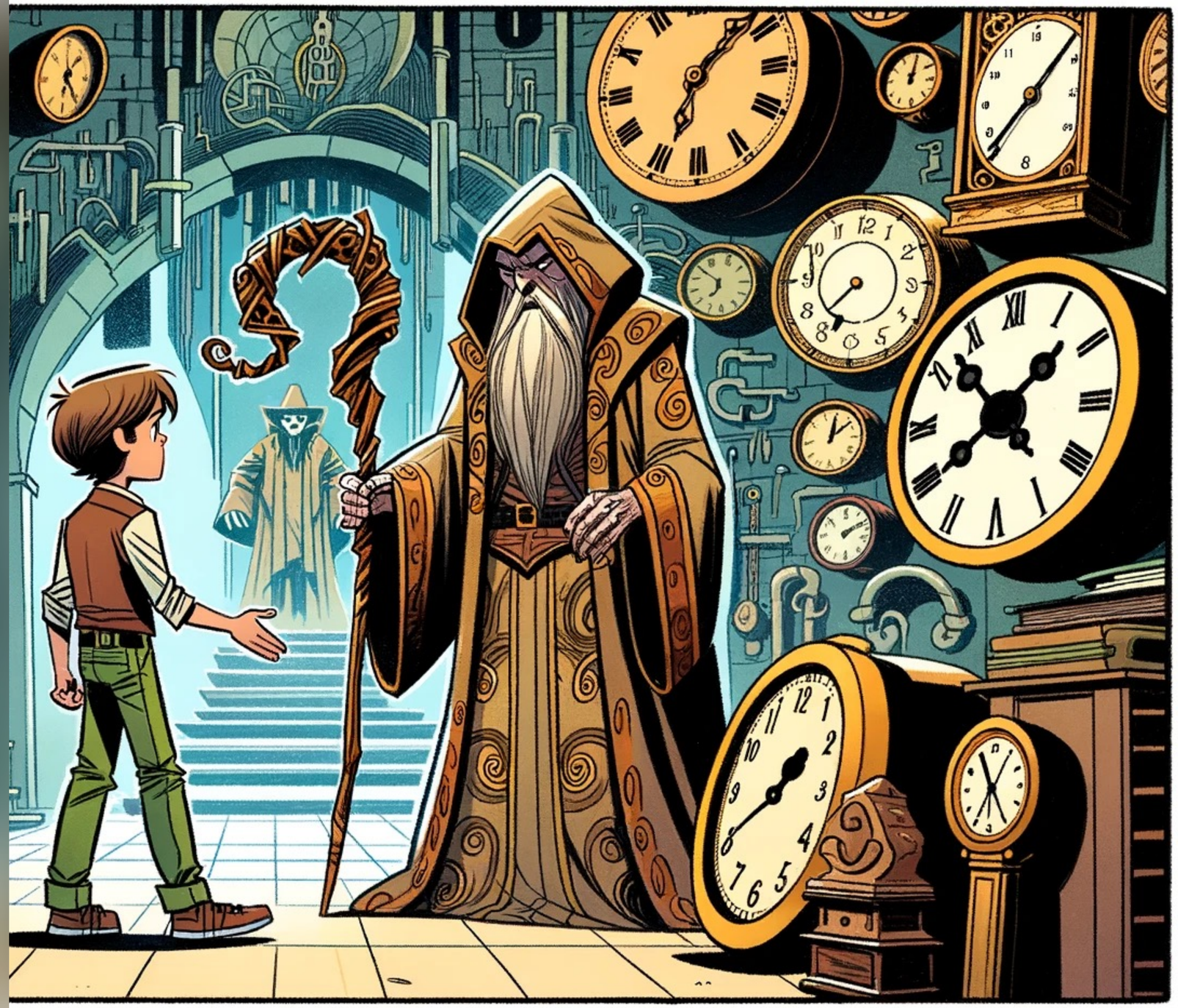
```
e45f01.fffe.c75bc0-0 seq 10 SIGNALING SLAVE_RX_SYNC_TIMING_DATA N 1
        sourcePortIdentity          e45f01.fffe.d83f9d-1
        sequenceId                  21
        syncOriginTimestamp         1697649554.565337688
        totalCorrectionField        0
        scaledCumulativeRateOffset  0
        syncEventIngressTimestamp   1698405472.685634664
e45f01.fffe.c75bc0-0 seq 9 SIGNALING SLAVE_DELAY_TIMING_DATA_NP N 1
        sourcePortIdentity          e45f01.fffe.d83f9d-1
        sequenceId                  9
        delayOriginTimestamp        1698405472.988393480
        totalCorrectionField        0
        delayResponseTimestamp      1697649554.868138256
```

linuxptp 4.x

linuxptp 4.x

- Monitor multiple PHCs with phc2sys
- Free-running mode in phc2sys
- PTP minor version change
  - May break compatibility with old GMs
- Virtual clock support
- Dynamic clock tree reconfiguration in ts2phc
- Add read-only UDS socket for monitoring

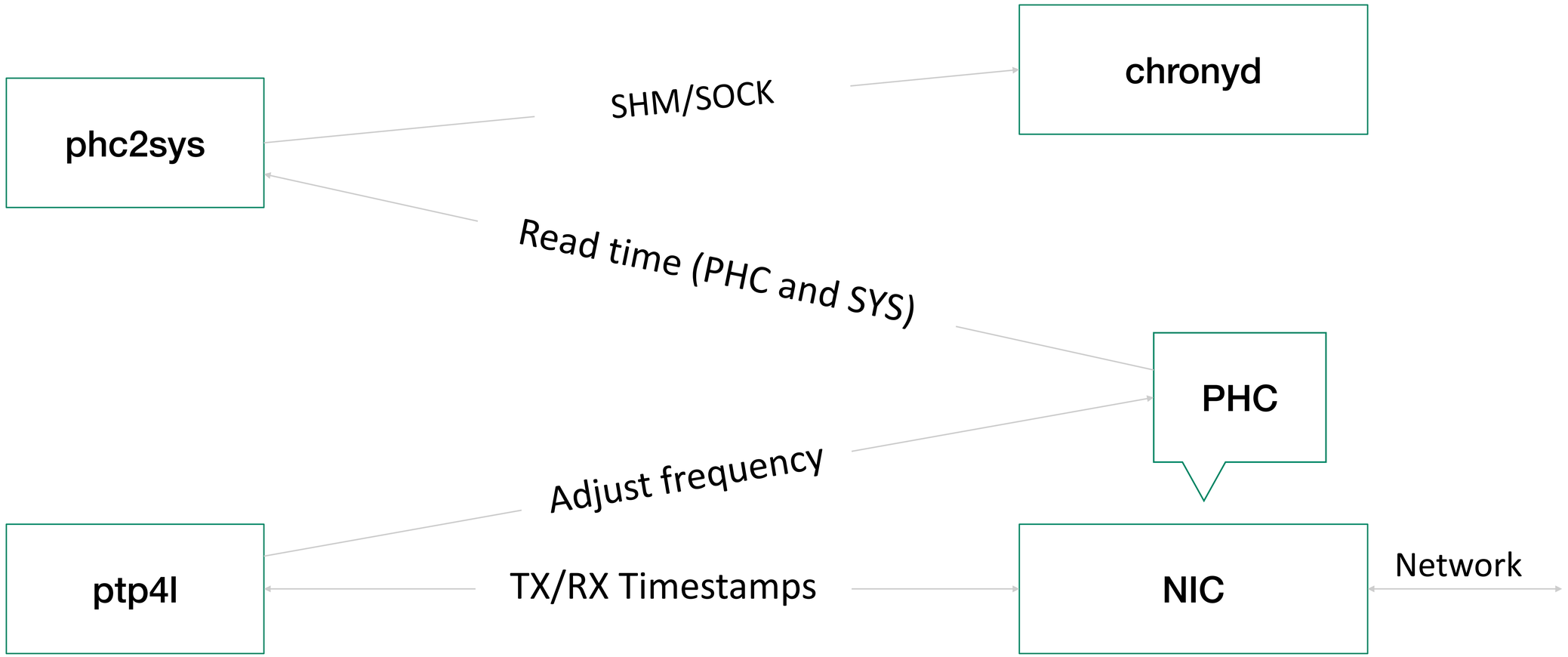
Meeting
the
timemater



timemaster

- Part of the **linuxptp**
- Uses **ptp4l** and **phc2sys**
- Combines with **chronyd** or **ntpd**
- Time from ptp is provided via
  - SOCK reference clock (chronyd)
  - SHM (chronyd/ntpd)
- generates configuration files for **ptp4l, phc2sys** and **chronyd/ntpd**,
- starts processes

timemaster


phc2sys
SHM/SOCK
chronyd
Read time (PHC and SYS)
PHC
Adjust frequency
ptp4l
TX/RX Timestamps
NIC
Network

chrony monitoring

```
chronyc tracking
```

The `tracking` command displays parameters of the system's clock performance. Example output can be:

```
Reference ID    : 50545030 (PTP0)
Stratum         : 1
Ref time (UTC)  : Wed Jul 13 14:06:03 2022
System time     : 0.000000002 seconds slow of NTP time
Last offset     : -0.000000001 seconds
RMS offset      : 0.000000013 seconds
Frequency       : 26.404 ppm slow
Residual freq   : +0.000 ppm
Skew            : 0.006 ppm
Root delay      : 0.000010000 seconds
Root dispersion : 0.000002921 seconds
Update interval : 4.0 seconds
Leap status     : Normal
```

chrony monitoring

```
chronyc sources [-v]
```

```
.-- Source mode  '^' = server, '=' = peer, '#' = local clock.
/ .- Source state '*' = current best, '+' = combined, '-' = not combined,
| /             'x' = may be in error, '~' = too variable, '?' = unusable.
||                                                 .- xxxx [ yyyy ] +/- zzzz
||      Reachability register (octal) -.           |  xxxx = adjusted offset,
||      Log2(Polling interval) --.      |          |  yyyy = measured offset,
||                                \     |          |  zzzz = estimated error.
||                                 |    |           \
MS Name/IP address         Stratum Poll Reach LastRx Last sample
===============================================================================
#* PTP0                           0   2   177     4     -1ns[   -5ns] +/- 5016ns
^? 192.168.1.242                 0   6     0     -     +0ns[   +0ns] +/-    0ns
^? _gateway                      2   6     1    37   +172us[ +172us] +/-  268ms
^? 94-172-186-238.dynamic.c>     2   6     1    38   -380us[ -380us] +/-   36ms
```

chrony monitoring

```
chronyc sourcestats [-v]
```

```
                             .- Number of sample points in measurement set.
                            /    .- Number of residual runs with same sign.
                           |    /    .- Length of measurement set (time).
                           |   |    /      .- Est. clock freq error (ppm).
                           |   |   |      /           .- Est. error in freq.
                           |   |   |     |           /         .- Est. offset.
                           |   |   |     |          |          |   On the -.
                           |   |   |     |          |          |   samples. \
                           |   |   |     |          |          |             |
Name/IP Address            NP  NR  Span  Frequency  Freq Skew  Offset  Std Dev
==============================================================================
PTP0                        6   3    20     +0.000      0.003     +1ns     7ns
192.168.1.242               0   0     0     +0.000   2000.000     +0ns  4000ms
_gateway                   23  11   28m     -0.029      0.029    +49us    17us
94-172-186-238.dynamic.c>  20  12   26m     +0.055      1.288   +182us   653us
```