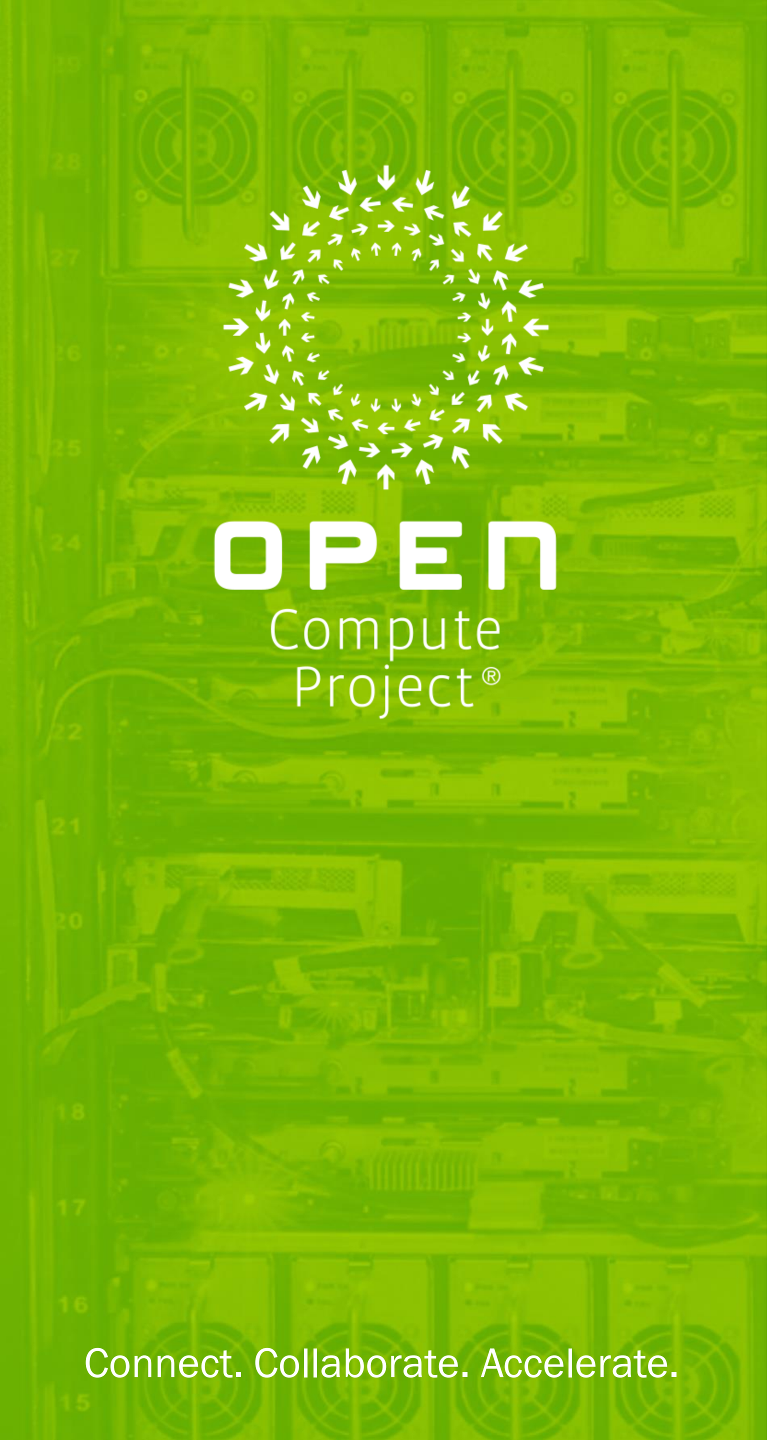


OPEN Compute Project®

Connect. Collaborate. Accelerate.

Time-Sync Beyond Ethernet:

CPU, Wi-Fi, and 5G

Intel

Kevin B. Stanton

Agenda

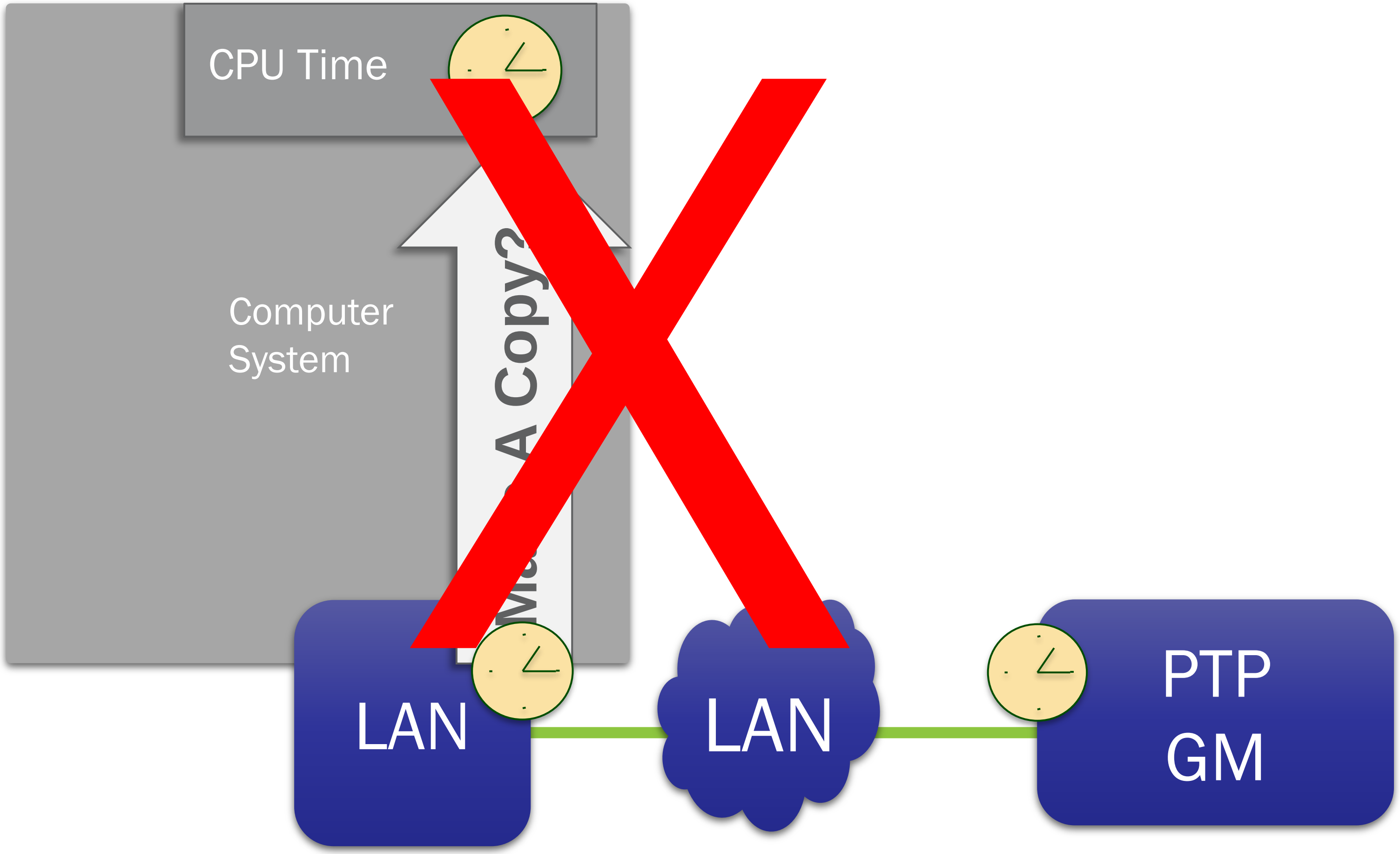
1. Software's Access to "Now"
2. Accurately Transferring Time to the CPU
3. Non-Ethernet Time-Synchronization

Application Software separated from Network Time by a large chasm

Now

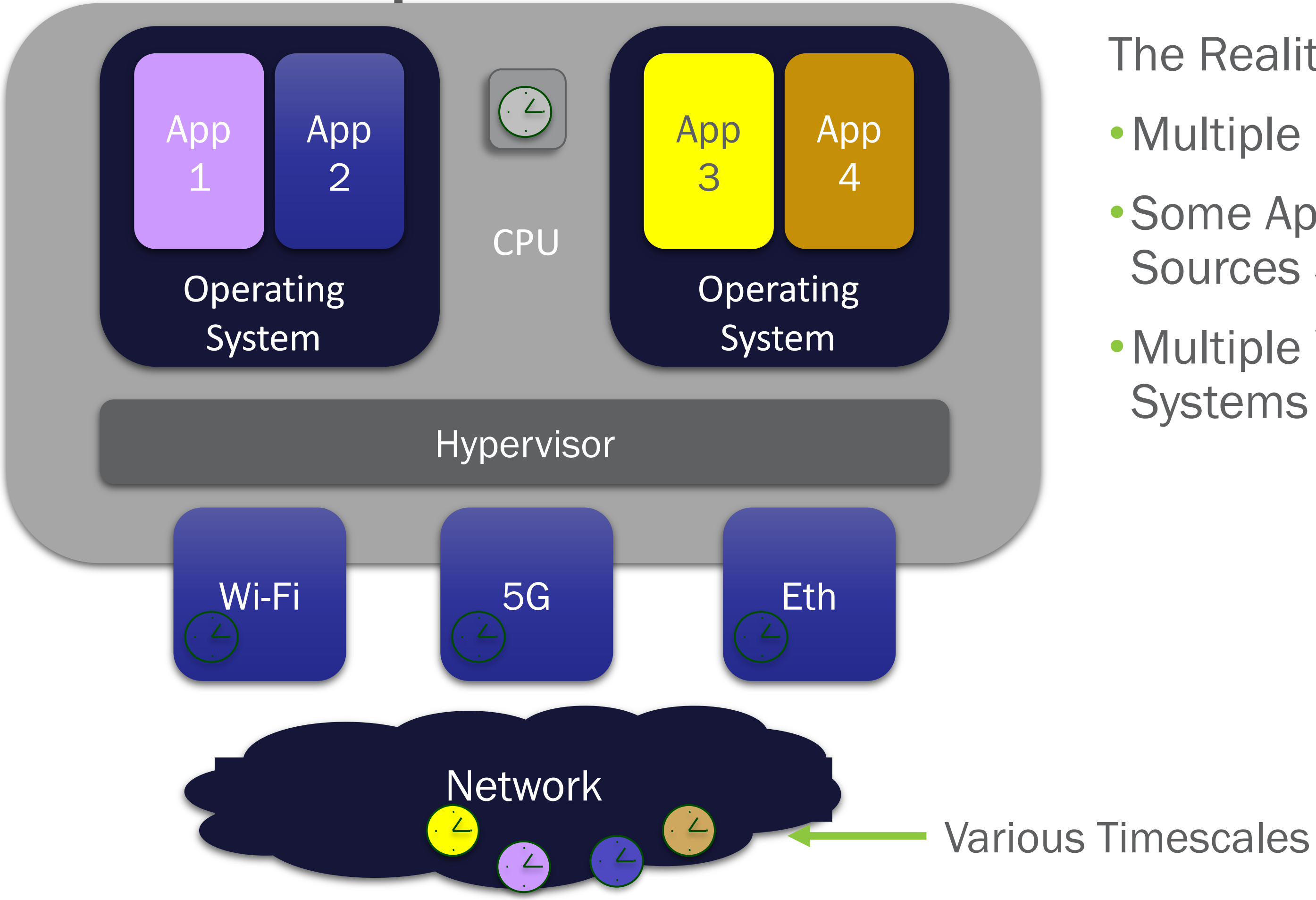
For software, Reading Time from a Network Peripheral can be VERY SLOW

One Approach for Bringing Network Time Near to Software



Replicating Network Time in the CPU Hardware Doesn’t Scale—See next slide

Modern Computer Systems Aren't So Simple

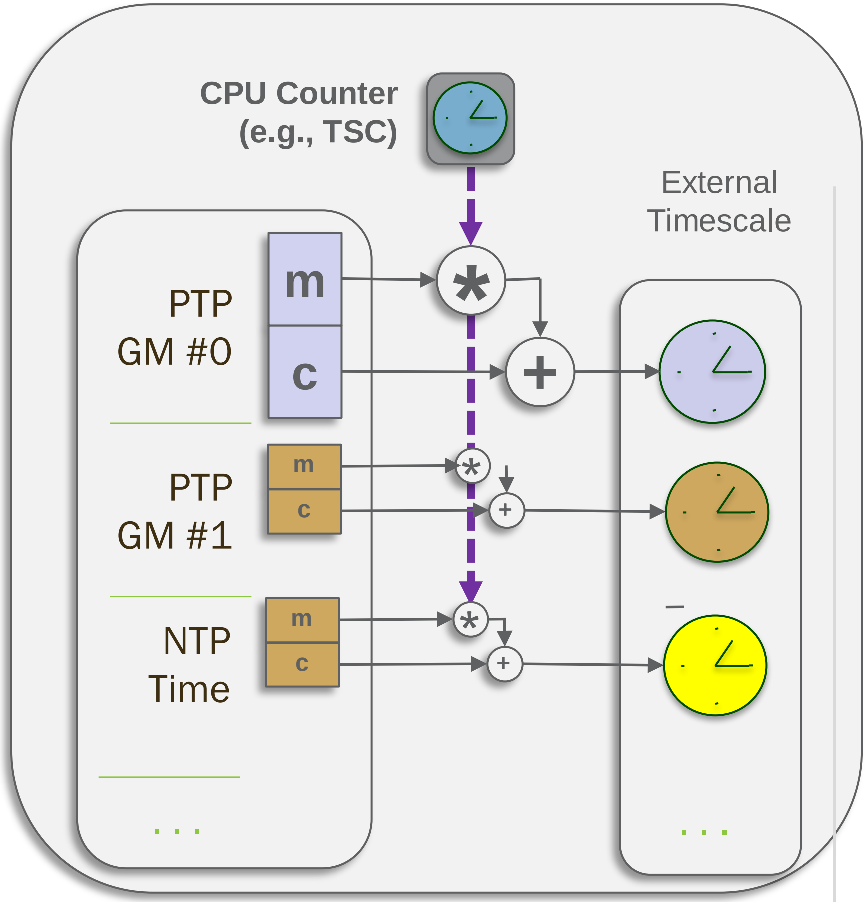


The Reality:

- Multiple Network Time Sources
- Some Applications Track Multiple Time Sources Simultaneously
- Multiple Virtual Machines / Operating Systems

Adding Multiple Hardware Times in the CPU Doesn't Scale

Scalable Timescale Representation



- Here’s what’s needed:

1. A “Stable-Enough” HW Reference
2. Fast * and + Operations
3. Precise estimate of m and c

- ➔ Any Timescale to/from Any Timescale

Timescale Translation Scales Well

Connect. Collaborate. Accelerate.

CPU Counter → Synchronized Time

Time "now" (from a Linux Application)

```
(1)  clock_gettime(CLOCK_MONOTONIC_RAW,  &now);
```

- Returns current TSC value scaled to nominal nanoseconds

```
(2)  clock_gettime(CLOCK_MONOTONIC,      &now);
```

- Returns current TSC value scaled to track TAI, in nanoseconds

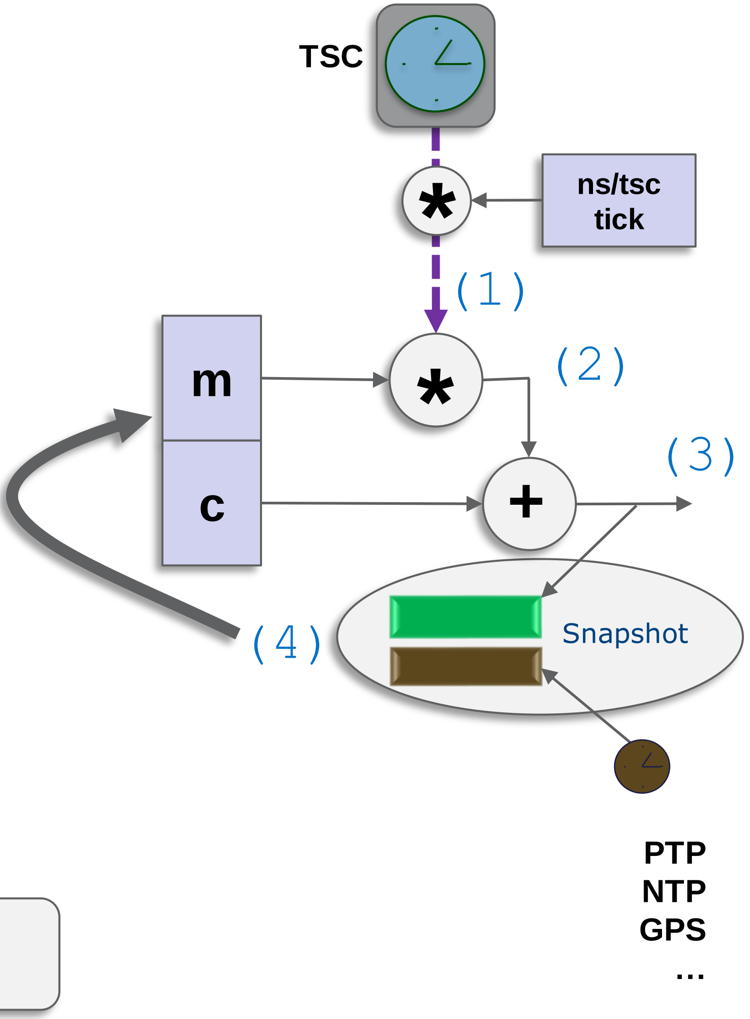
```
(3)  clock_gettime(CLOCK_REALTIME,       &now);
```

- Returns CLOCK_MONOTONIC + (now-1/1/1970) [incl. leap seconds]

Cross-Timestamp Snapshot

```
(4)  ioctl(phc_fd,PTP_SYS_OFFSET[_PRECISE], &offset )
```

- returns the triple:
  - `eth_ptp_time; realtime; monotonic_raw`

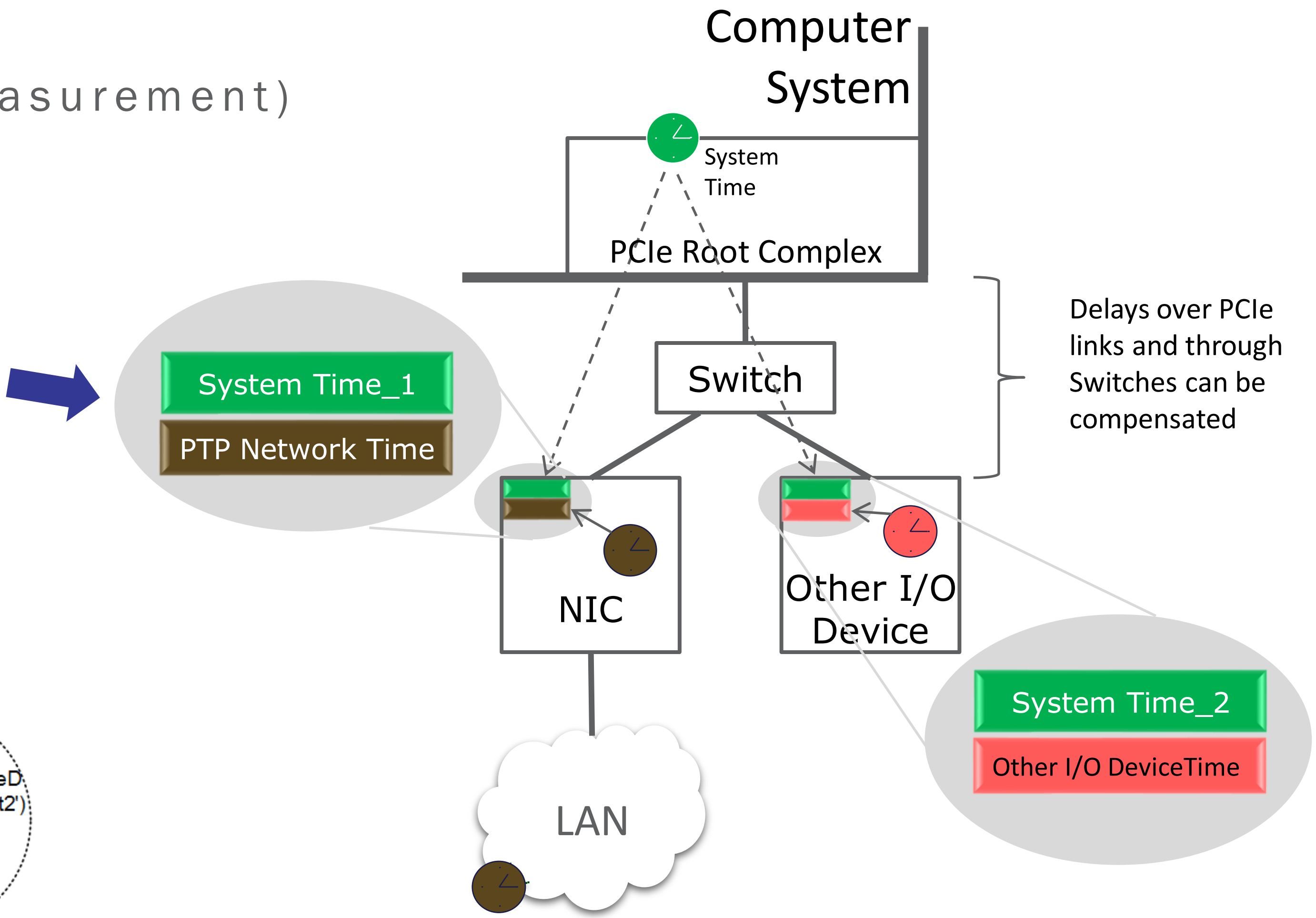


POSIX: Piecewise-Linear Clock Model: y[n]=mx[n]+c
Don't Change the TSC Value

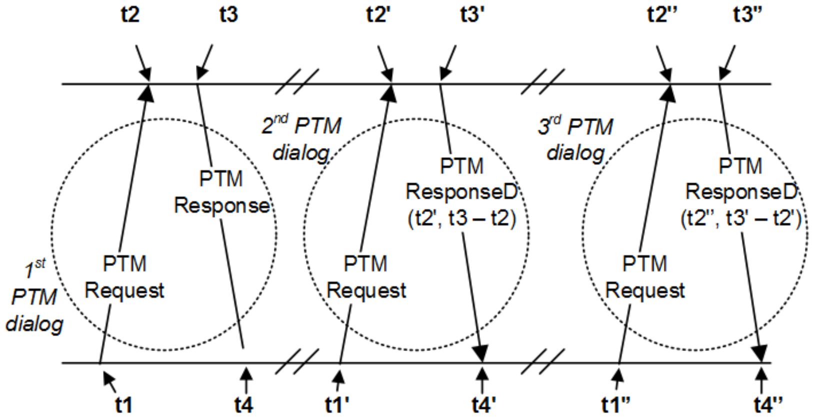
Using PCIe PTM to Cross-Timestamp

(PTM=Precision Time Measurement)

Cross Timestamps, <u>Captured Simultaneously</u>



Delays over PCIe links and through Switches can be compensated



PTM measurements presented by Chris Hall to OCP TAP are here: https://www.youtube.com/watch?v=JgHD1CU4Ycs

Agenda

1. Software's Access to "Now"
2. Accurately Transferring Time to the CPU
3. Non-Ethernet Time-Synchronization

Accurate Time Over Heterogeneous Links

- Ethernet:
  - Many flavors (profiles) of PTP. End-to-End, Peer-to-Peer.
  - Biggest challenge is Switch support for the proliferation of PTP profiles
- PCIe PTM
  - Similar to 1588 Pdelay: Round-Trip 4-timestamps. CPU clock used as shared reference.
  - *Google search: PCIe PTM*
- USB PTM:
  - USB bus clock used as shared reference between Host Controller & Dev (Si support in the latter lags)

As part of Time-Sensitive Networking (TSN), the 802.1AS profile of 1588 is supported with:

- Wi-Fi:
  - 802.11 [Fine] Timing Measurement ([F]TM), similar to PTP PDelay: Simple round-trip 4 timestamps,
  - Immune to retransmission. *Google Search: Avnu WTSN*
- 5G URLLC:
  - Uses 5G system clock as a common reference across infrastructure & UEs, per 3GPP
  - The 5G system appears like a 1588 Transparent Clock. *Google Search: 802.1AS 5G URLLC*

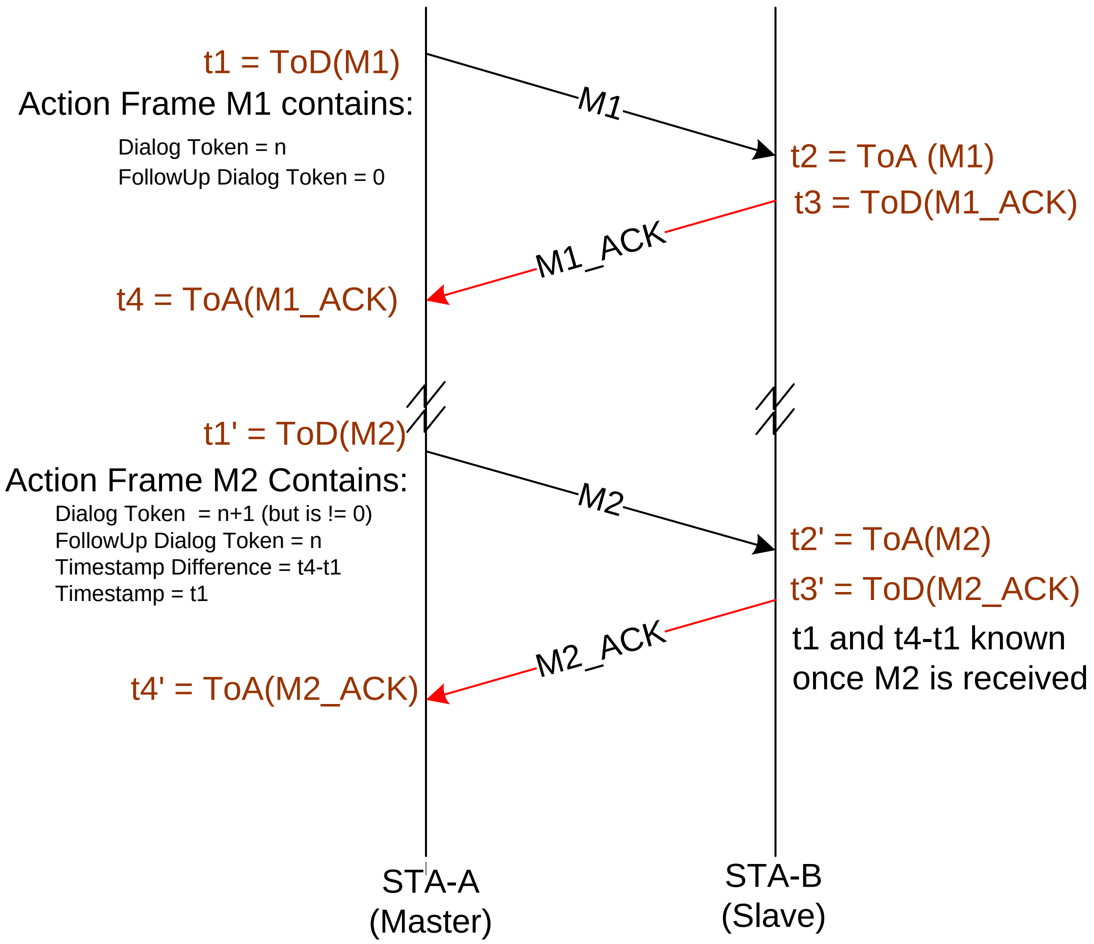
And beyond…

- UWB, I3C, GNSS, WWV, …

Backup

PTP (the 802.1AS Profile) over 802.11 links

Using the 802.11 TimingMeasurement (or FineTimingMeasurement) protocol



t1 = ToD(M1)

Action Frame M1 contains:

Dialog Token = n
FollowUp Dialog Token = 0

t2 = ToA (M1)

t3 = ToD(M1_ACK)

t4 = ToA(M1_ACK)

t1' = ToD(M2)

Action Frame M2 Contains:

Dialog Token = n+1 (but is != 0)
FollowUp Dialog Token = n
Timestamp Difference = t4-t1
Timestamp = t1

t2' = ToA(M2)

t3' = ToD(M2_ACK)

t1 and t4-t1 known
once M2 is received

t4' = ToA(M2_ACK)

NOTE: M1 and M2 have exactly the same format—
they’re TIMINGMSMT Private Action Frames (and Unicast, BTW)

First exchange:

- takes a measurement

Subsequent exchange:

- takes a measurement
- also passes timestamps from prior measurement

Free-running counter used for timestamps

Allows us to compute:

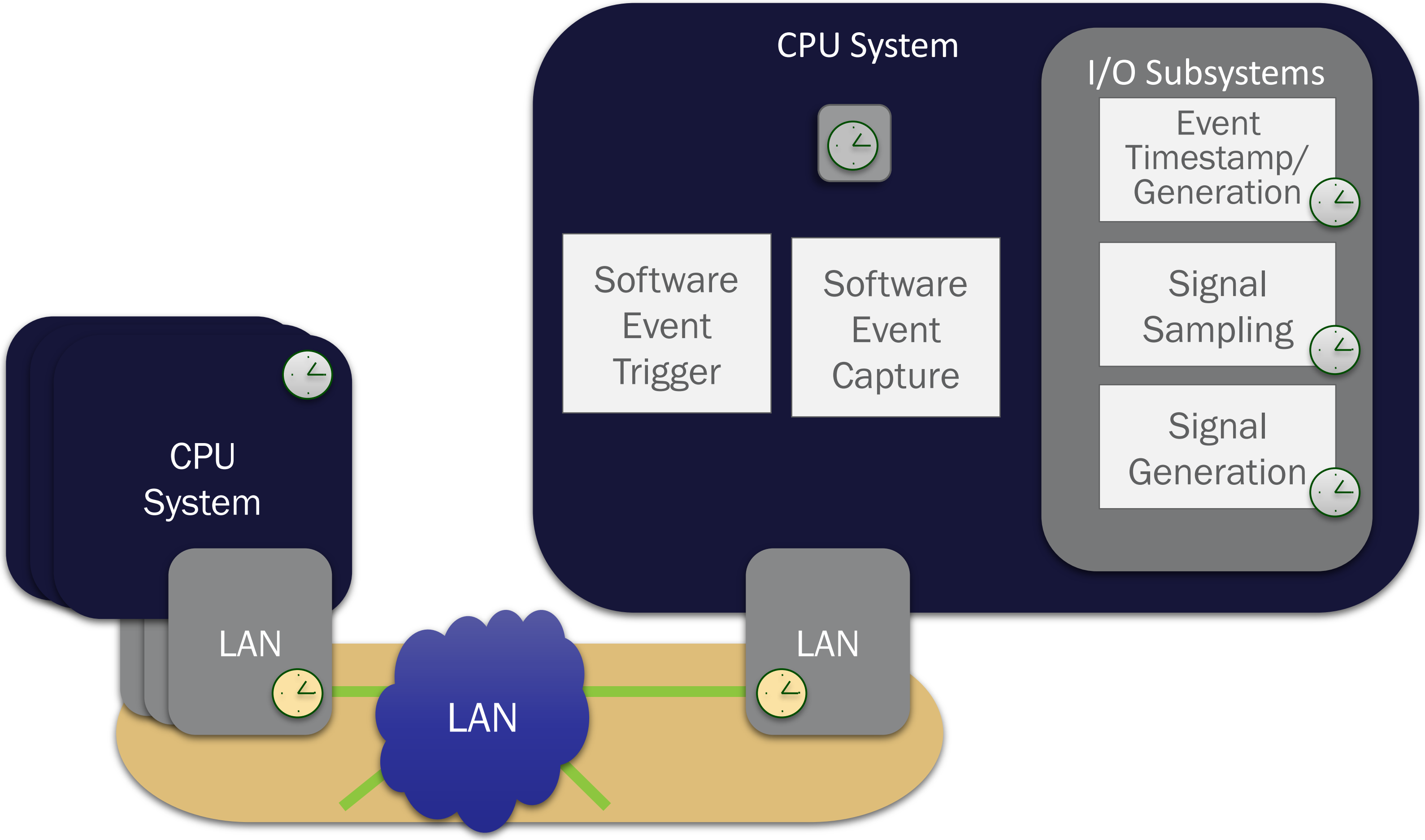
```
neighborRateRatio =
   (t1’-t1)/(t2’-t2)

linkDelay =
        [(t4-t1)-(t3-t2)]/2

timeOffset=
        [(t2-t1)-(t4-t3)]/2
```

[note: rateRatio is also applied]

Derived from ieee802.org/1/files/public/docs2014/as-kbstanton-8021AS-tutorial-0714-v01.pdf

Kevin B. Stanton

Computer Time Architecture



Connect. Collaborate. Accelerate.