Tutorial on the Time-Synchronization Standard from the AVB suite:
IEEE Std 802.1AS™-2011

Kevin B. Stanton, Ph.D.   July 2014

Sr. Principal Engineer

Intel Corporation

kevin.b.stanton at intel.com

Abstract

This presentation provides an overview of clock and time synchronization across heterogeneous networks using the published standards:

IEEE Std 802.1AS™-2011 and

IEEE Std 802.11™-2012 Timing Measurement

and with references to the in-progress revision:

P802.1ASbt

Kevin B. Stanton

Agenda

- **802.1AS goals / applications**
- **Overview**
- **Grand Master selection**
- **Time propagation**
  - Media-independent
  - Media-dependent
- **Time *inside* the system**

802.1AS goals

Distribute a single, accurate, time reference that is optimized for audio/video synchronization

- **Accurate**
    - 802.1AS time in participating LAN-attached stations is accurate to within 1us
        - If one station thinks it's 09:57:55 AM PST, the others agree within specified # of network hops
        - Assuming +/-100PPM crystal, 1PPM / second maximum frequency drift
        - Accuracy has been physically demonstrated in even near-worst-case LAN environments
- **One time reference for the entire LAN/Subnet**
    - LAN-agnostic architecture
        - Can be applied to any 802-compatible LAN
            - Is explicitly defined for 802.3 (Ethernet), 802.11 (Wi-Fi), MoCA, ITU-T G.hn, and EPON.
        - Contains a Profile of IEEE Std.1588™ -2008 (PTP, or Precision Time Protocol)
            - 1588-2008 reserves a special value for the 802.1AS profile (see Transport Specific in IEEE Std. 1588™-2008)
    - LAN-specific measurements
        - 802.3 measurements follow generic 1588
        - 802.11 measurements leverages IEEE Std. 802.11™ -2012 "Timing Measurement"
            - Formerly known as IEEE Std. 802.11v™ -2011
        - Coordinated Shared Network (CSN) measurements defined also
- **Plug and play**
    - Grand Master (GM) clock is selected automatically
        - Time stabilizes in a fraction of a second
    - Clock tree reconfigures automatically if GM is lost

These are **NOT** explicit 802.1AS goals/characteristics

NO!: Improve the latency of packets/frames (but useful—see TSN)

NO!: Improve the delivery jitter of packets/frames (see TSN)

NO!: Constrain delivery jitter of 802.1AS packets

NO!: Distribute a media clock

> Media clocks are typically described by referencing 802.1AS-based presentation times

NO!: Work over Ethernet only (or 802.11 only)

NO!: Limited to 7 network hops

How 802.1AS is used with audio/video

- **First, 802.1AS time-sync is <u>NOT</u> needed where:**
  - *Stored* A/V content is streamed to *one* renderer
    - Renderer can pull content from storage as it is needed
  - *Self contained* A/V system (network isn't in the path)
- **Shared time <u>IS</u> needed:**
  - When there are multiple renderers or sources
  - Where the renderer cannot tell the source to slow down or speed up, yet latency must be very small

- **The "Presentation Time" uses 802.1AS time to port the media rate**
  - "Render audio sample #896 when 802.1AS-time is 8:11:35.023712"
  - Every "presentation time" provides render time of specific sample
  - Every pair of "presentation times" communicates the actual "rate"
    - Nominal media rate (e.g., 44.1KHz audio, or 165MHz for HDMI) is assumed known
    - But actual media rate always differs by a non-zero PPM, and changes over time due to thermal and other factors

How 802.1AS is used in industrial/IOT

- **First, 802.1AS time-sync is <u>NOT</u> needed where:**
  - Coordination BETWEEN devices is not necessary
  - *All the data comes from and is acted upon by a single device*
- **Shared time <u>IS</u> needed:**
  - In cyber-physical systems (CPS)
    - For coordination of distributed inputs, outputs, computation, and communication
  - In sensor networks where big-data analytics are applied to the aggregate
    - …without losing temporal information
    - Especially for
      - Low-power, infrequently-communicating devices
      - Cooperative Diversity in wireless communications
      - Devices responsible cooperatively for beamforming or array processing

Kevin B. Stanton

Agenda

- **802.1AS goals / use cases**
- **Overview**
- **Grand Master selection**
- **Time propagation**
  - Media-independent
  - Media-dependent
- **Time *inside* the system**

High level architecture

- **Grand Master selection**
  - GM-capable stations advertise themselves via ANNOUNCE messages
  - If a station hears from station with "better" clock, it does not send ANNOUNCE
    - Configurable "Priority" field can override clock quality
    - MAC address is tie breaker
  - Time relays drop all inferior ANNOUNCE messages
    - Forward only the best
  - Last one standing is Grand Master for the domain
    - GM is the root of the 802.1AS timing tree
    - GM periodically sends the current time

- **Propagation of time**
  - Non-leaf devices in the tree propagate time toward the leaves
   - Taking queuing delay into account (aka "Residence Time")

802.1AS is not alone

- ## 802.1AS is one of three core 802.1 AVB standards
  - The others define a reservation protocol and traffic shaping algorithm

- ## 802.1AS over Ethernet (802.3) qualifies as a Profile of IEEE 1588-2008
  - 802.1AS simplifies 1588, adds other media (e.g. 802.11)
  - Also other optimizations
    - Better stability over large networks (no cascaded PLLs)
    - Faster GM failover

Agenda

- **802.1AS goals / use cases**
- **Overview**
- **Grand Master selection**
- **Time propagation**
  - Media-independent
  - Media-dependent
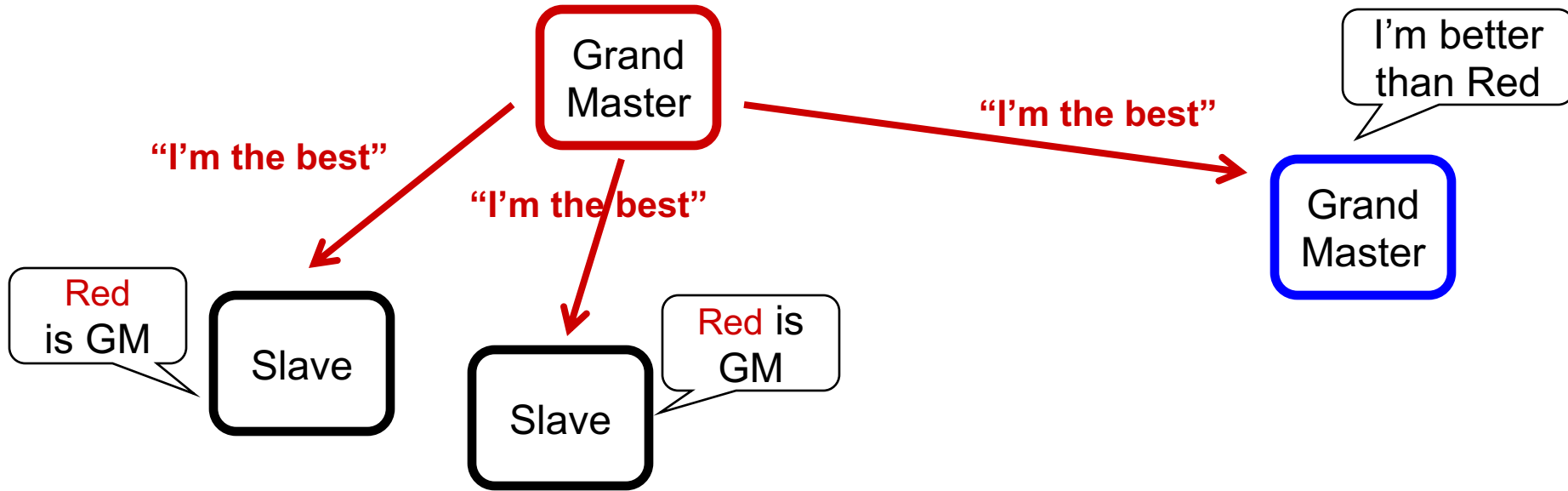- **Time *inside* the system**

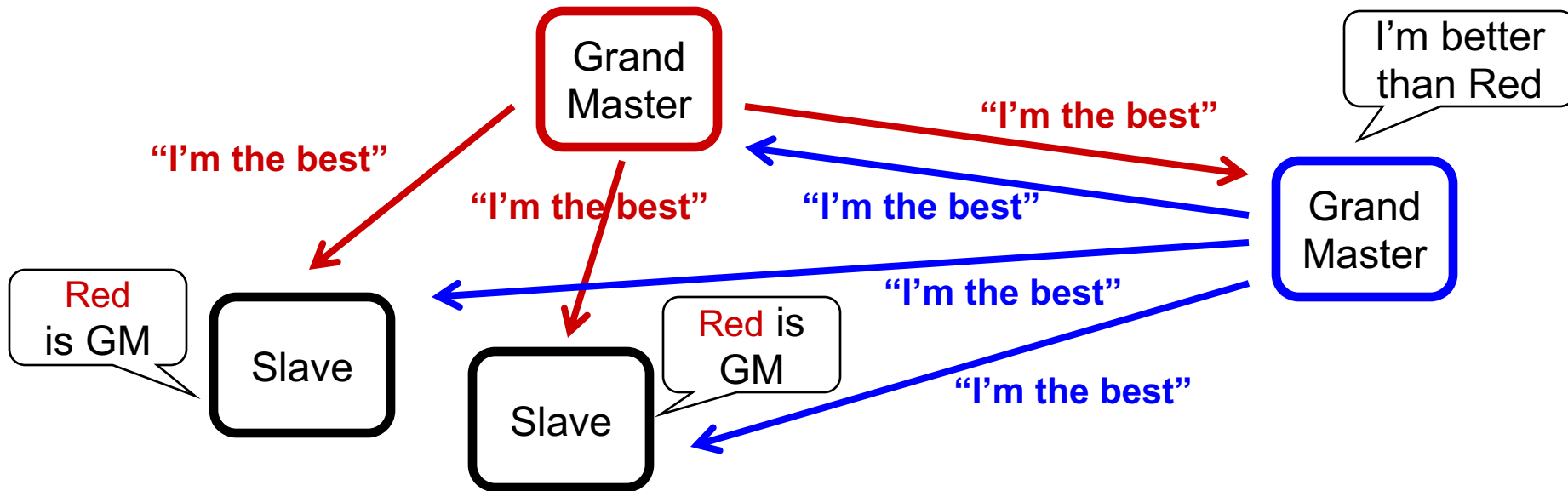# Grand Master selection – Steady state
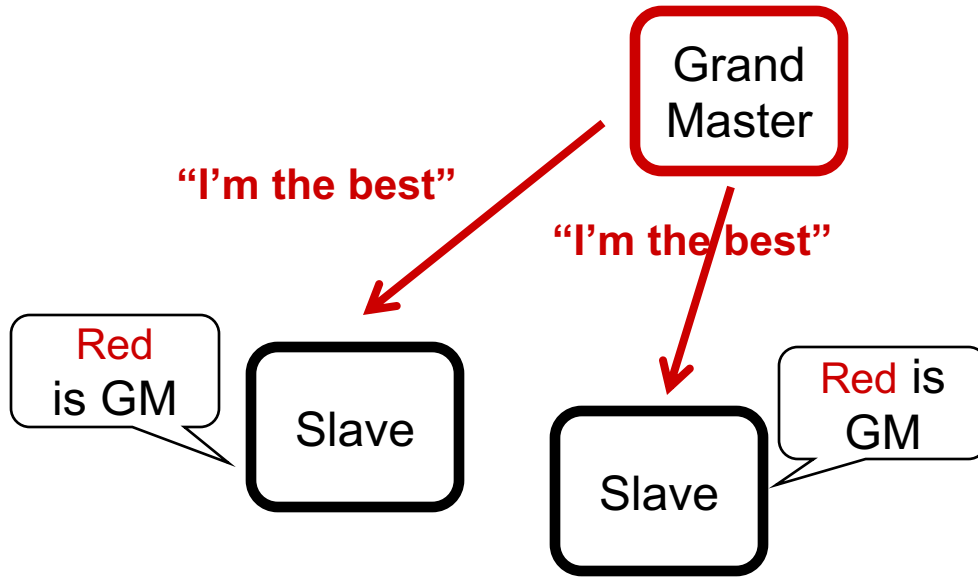
Steady state:

- **The GM sends ANNOUNCE**
- **Slaves acknowledge the GM is best (including self)**
  - …using simple bitwise compare
  - And do not send ANNOUNCE messages
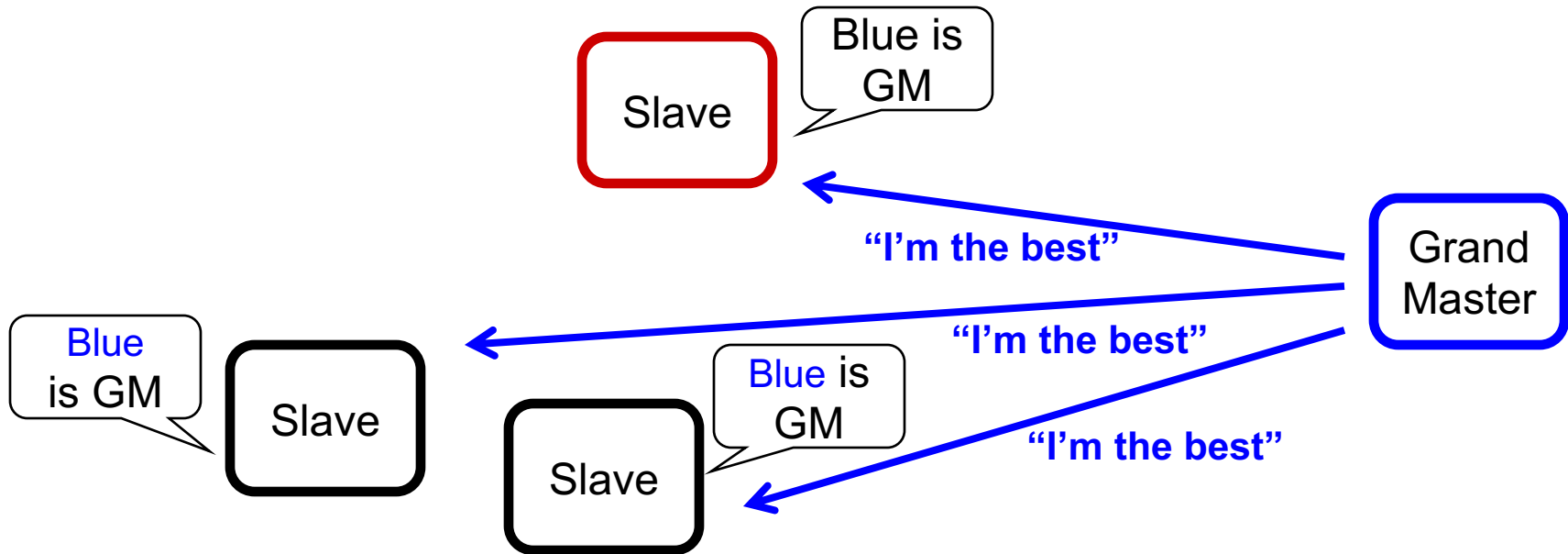- **Everyone knows their role**
- **Life is good**

- **Blue station with better clock appears**

Grand Master selection – New, better GM (2)


- **Blue station with better clock appears**

- **Blue sends ANNOUNCE**

- **Stations all realize blue is superior**
  - Including Red

Grand Master selection – New, better GM (3)

- **Blue station with better clock appears**

- **Blue sends ANNOUNCE**

- **Stations all realize blue is superior**

- **Red stops sending ANNOUNCE**

- **Blue is quickly the undisputed GM**

Grand Master selection – Lost GM

Grand Master

"I'm the best"

"I'm the best"

Grand Master

Red is GM

Slave

Slave

Red is GM
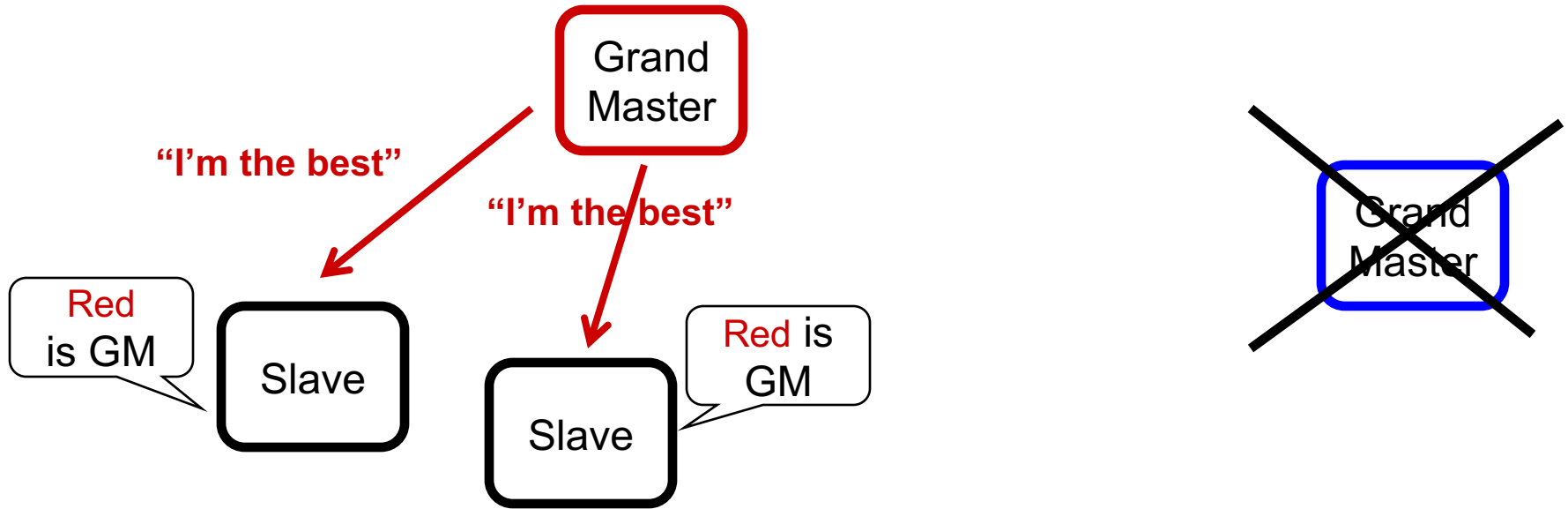
- **If Blue disappears, all GM-capable stations send ANNOUNCE**
- **Eventually, only Red sends ANNOUNCE**

ANNOUNCE comparison

- **The credentials passed in ANNOUNCE messages are compared (in order of decreasing importance):**

  - Priority (settable by management)
  - Multiple **"quality of my clock"** fields
  - MAC address

Kevin B. Stanton

Agenda

- **802.1AS goals / use cases**
- **Overview**
- **Grand Master selection**
- **Time propagation**
  - Media-independent
  - Media-dependent
- **Time *inside* the system**

Time domain is propagated to the "subnet"

- **Time-aware Relays**
  - The best ANNOUNCE
  - The GM's time
- **Relays can be a**
  - Bridge
  - Router
  - Linux computer with 2 Ethernet, a wireless and MoCA NIC(s)
- **Links can be**
  - 802.3 Ethernet
  - 802.11 WiFi
  - Almost anything compatible with IEEE 802

- **Let's look at one relay**
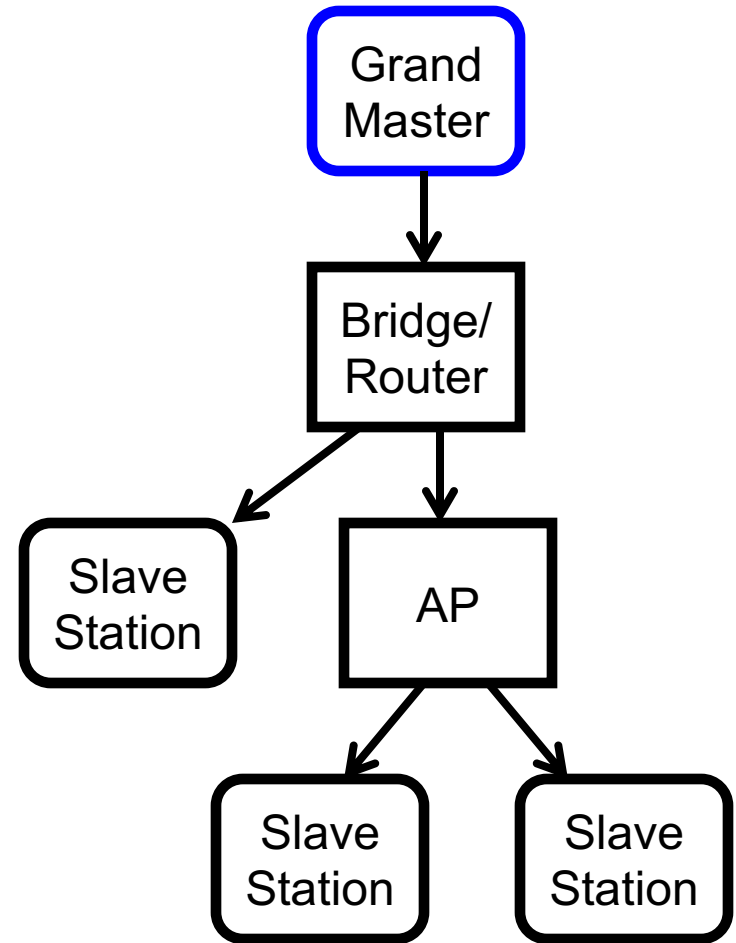
Grand Master selection – Relays help decide

- **Relays filter inferior ANNOUNCE**
  - Relays forward best ANNOUNCE messages
  - Relays drop inferior ANNOUNCE messages
  - Benefits:
    - Reduction in network traffic
    - Faster GM selection

- **Announce messages establish the Clock Tree**

Kevin B. Stanton

Grand Master responsibilities
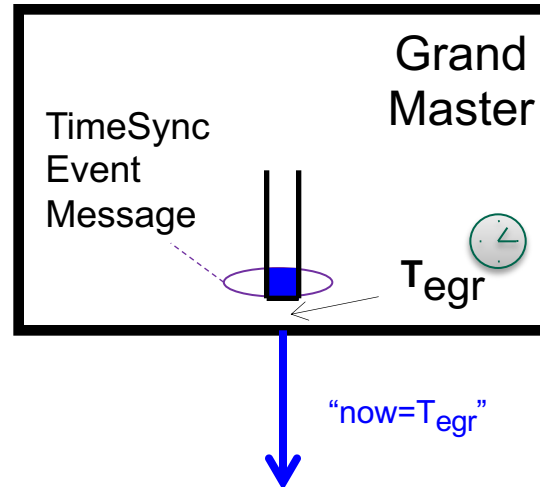
- **Transmit an Event message each Sync Interval**

- **Hardware captures the egress, or Tx time (T_{egr})**

- **Pass T_{egr} downstream**
  - Typically in a Follow-Up message

- **GM time need not be synchronized to atomic time**
  - …for many applications, but UTC offset can also be provided

Transferring time through a network device

T_{link}

"now=37"

T_{in}

Frames are not "forwarded"
Rather, they are processed
and new frames generated

"now=37+(T_{egr3}-T_{in})+T_{link}"

T_{egr3}

T_{egr2}

"now=37+(T_{egr2}-T_{in})+T_{link}"
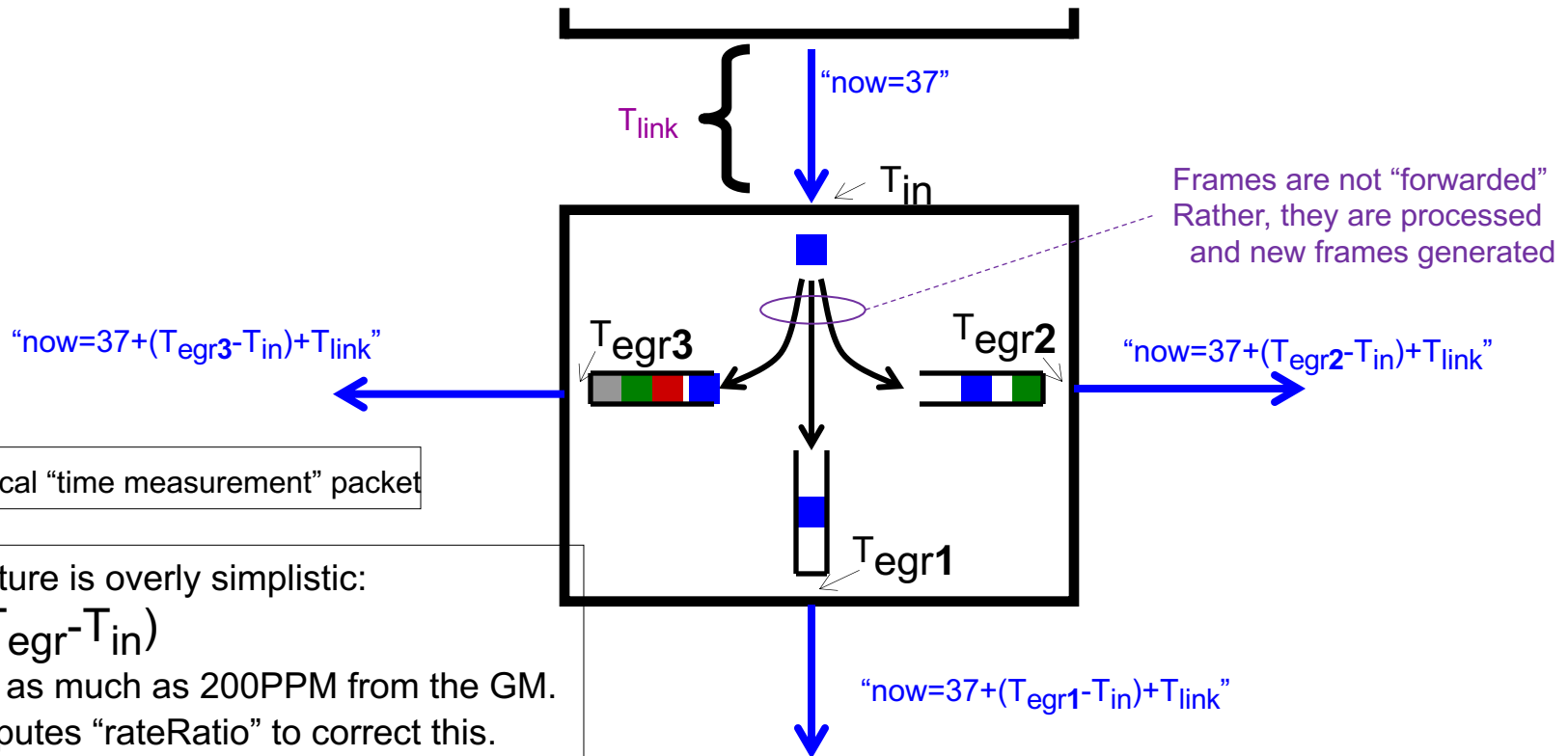
■ is a hypothetical "time measurement" packet

T_{egr1}

Note: This picture is overly simplistic:

$$(T_{egr}-T_{in})$$

can be off by as much as 200PPM from the GM.
802.1AS computes "rateRatio" to correct this.

"now=37+(T_{egr1}-T_{in})+T_{link}"

- **Time is sent from master over the link with frames that are LAN/media-specific**
  - Illustrated abstractly here by a Blue frame
- **Accurately measure how the long Blue frame is delayed in processing/queueing**
  - Called "Residence Time"
- **Also compensate for link delay (T_{link}) and rateRatio**

Example linkDelay measurement

Device 1   Network link   Device 2

t1=0

25

t2=5000

5

t2'=5005

t1'=25

time
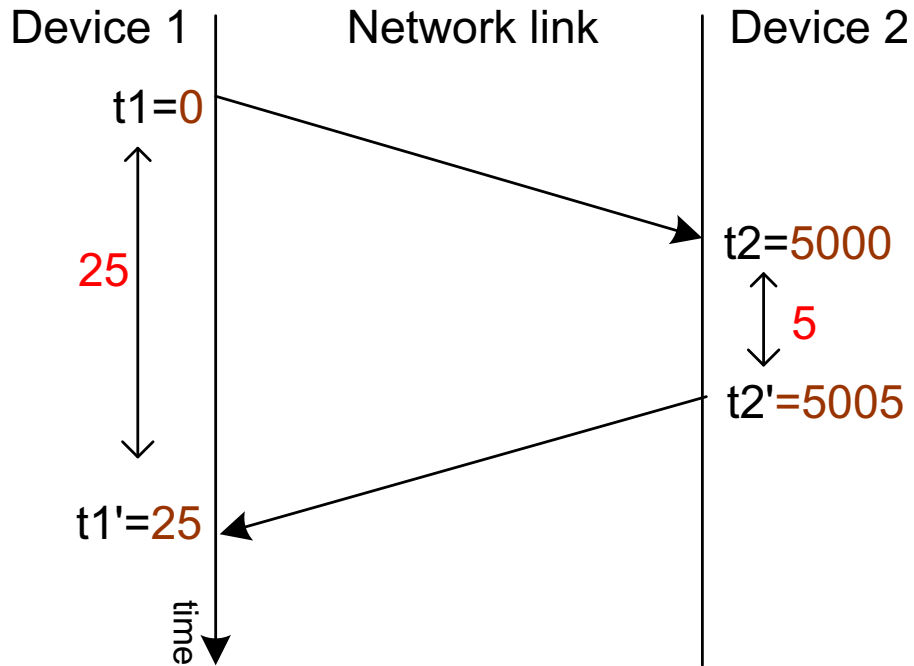
$$linkDelay = \left(\frac{(t1'-t1) - (t2'-t2)}{2} \right)$$

$$= \left(\frac{(25-0) - (5005-5000)}{2} \right)$$

$$= 10$$

If link delay is symmetrical, link delay is 10

Example neighborRateRatio measurement
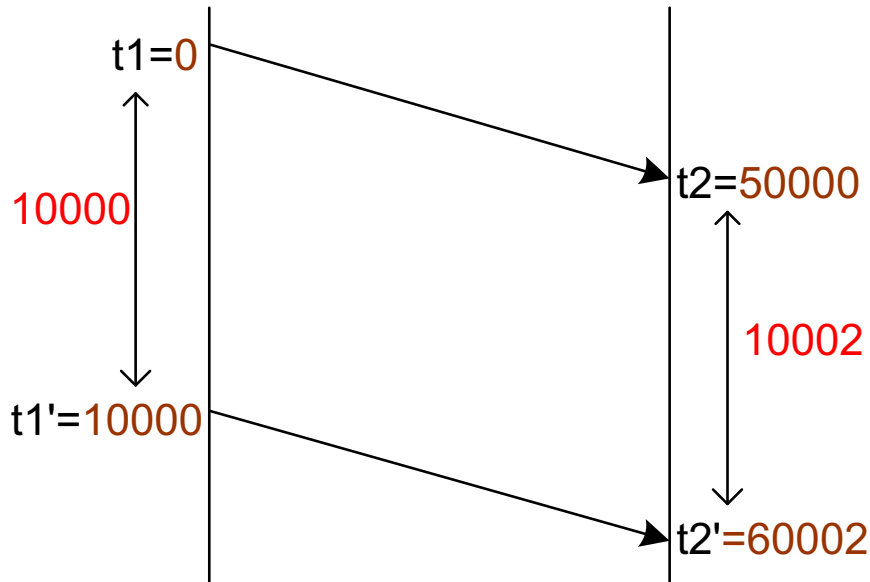
$$\text{neighborRateRatio} = \frac{(t1'-t1)}{(t2'-t2)}$$

$$= \frac{(10000-0)}{(60002-50000)}$$

$$\approx 0.998$$

$\rightarrow$ neighbor is
slow by 2000 PPM

If link delay is constant, station on left is running 2000 PPM (0.2%) slower than station on the right

Note: one *could* measure neighborRateRatio and linkDelay using the same packets (802.11 does, 802.3 does not)

Link Delay and Neighbor-Rate-Ratio

All ports measure

- – linkDelay to neighbor
- – neighborRateRatio

Example values are shown

Station

linkDelay=5
neighborRateRatio= 1+60PPM

↕

linkDelay=5
neighborRateRatio= 1-60PPM

Time Relay (e.g. Router)

linkDelay=20
neighborRateRatio=1+10PPM

↕

linkDelay=20
neighborRateRatio= 1-10PPM

Time Relay (e.g. AP)

linkDelay=15
neighborRateRatio= 1-60PPM

↕

linkDelay=15
neighborRateRatio= 1+60PPM

Station
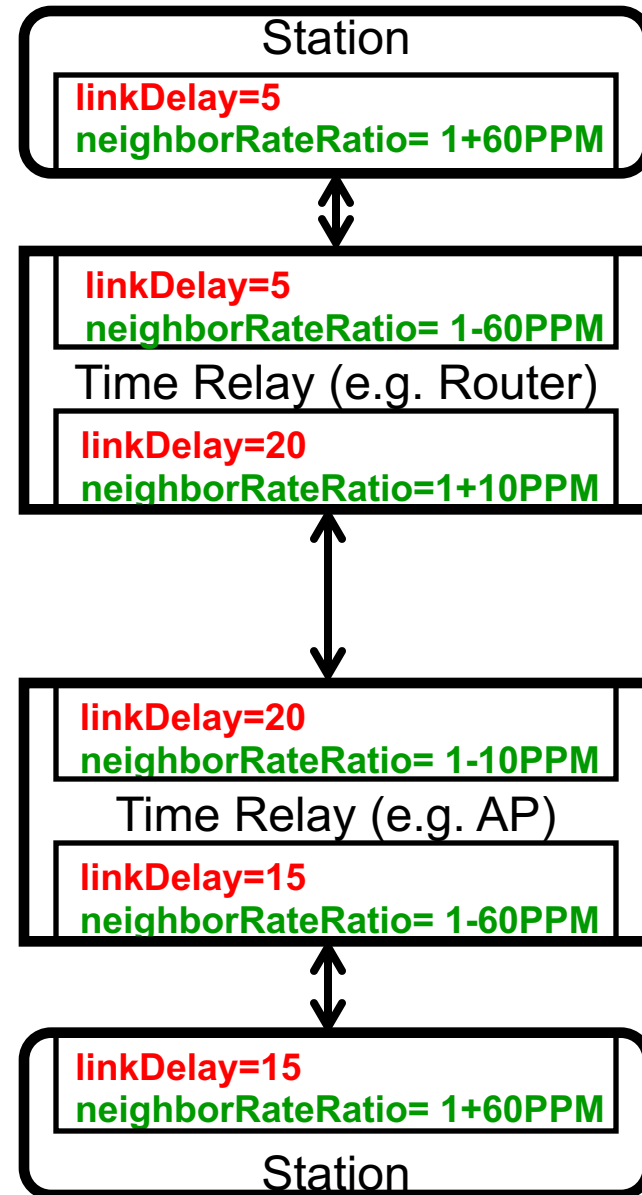
Note: No GrandMaster is needed here (yet)

End-to-end Rate Ratio

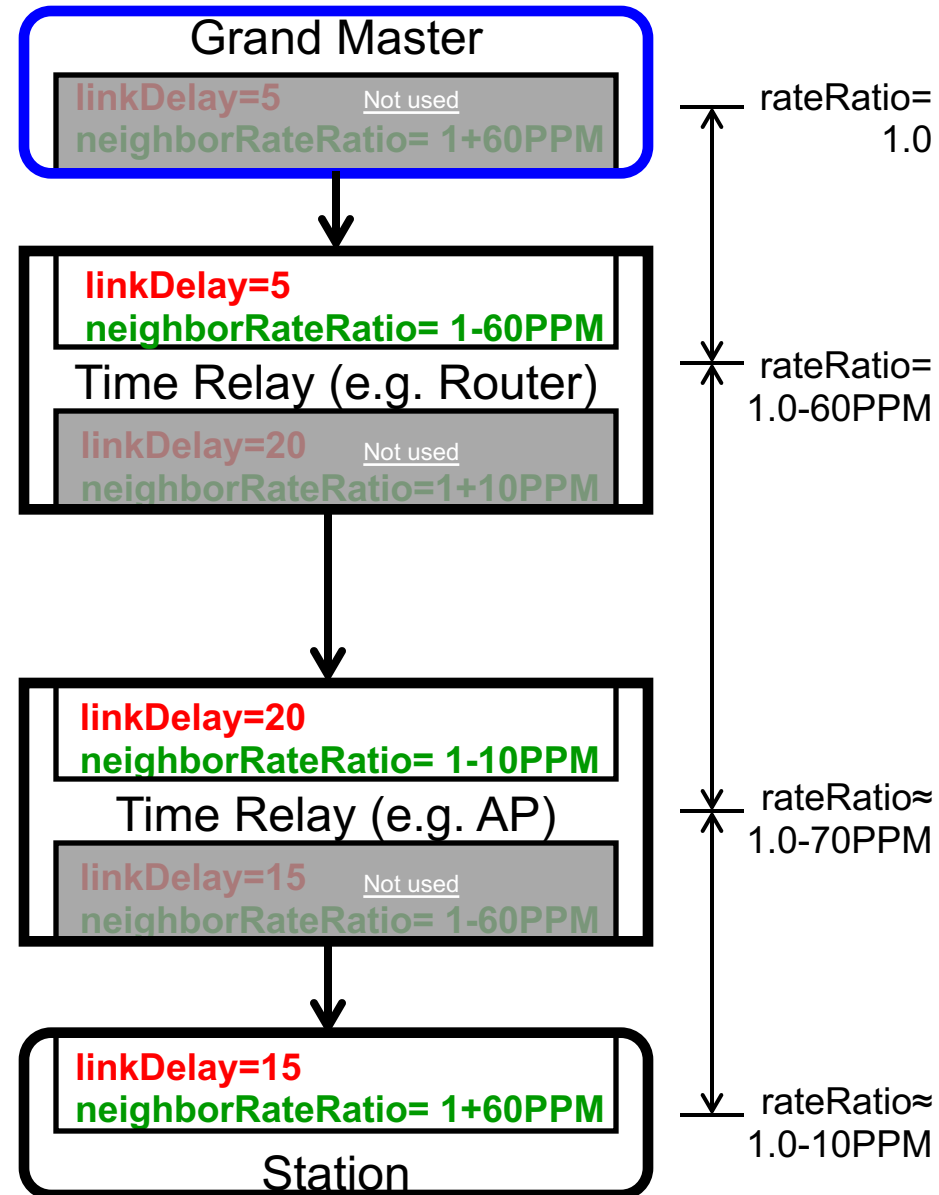**As time is propagated downward, neighborRateRatio is accumulated at each hop**

…using the approximation:

```
rateRatio +=
        (1-neighborRateRatio)
```

[Initial rate ratio is 1.0 at the GM]

Alternative is to do syntonization per relay, but cascaded PLLs are bad

Added benefit: When changing GM, endpoints stabilize quickly because neighbor parameters are already measured

Grand Master
linkDelay=5   Not used
neighborRateRatio= 1+60PPM

rateRatio= 1.0

linkDelay=5
neighborRateRatio= 1-60PPM
Time Relay (e.g. Router)
linkDelay=20   Not used
neighborRateRatio=1+10PPM

rateRatio= 1.0-60PPM

linkDelay=20
neighborRateRatio= 1-10PPM
Time Relay (e.g. AP)
linkDelay=15   Not used
neighborRateRatio= 1-60PPM

rateRatio≈ 1.0-70PPM

linkDelay=15
neighborRateRatio= 1+60PPM
Station

rateRatio≈ 1.0-10PPM

End-to-end time synchronization across the LAN
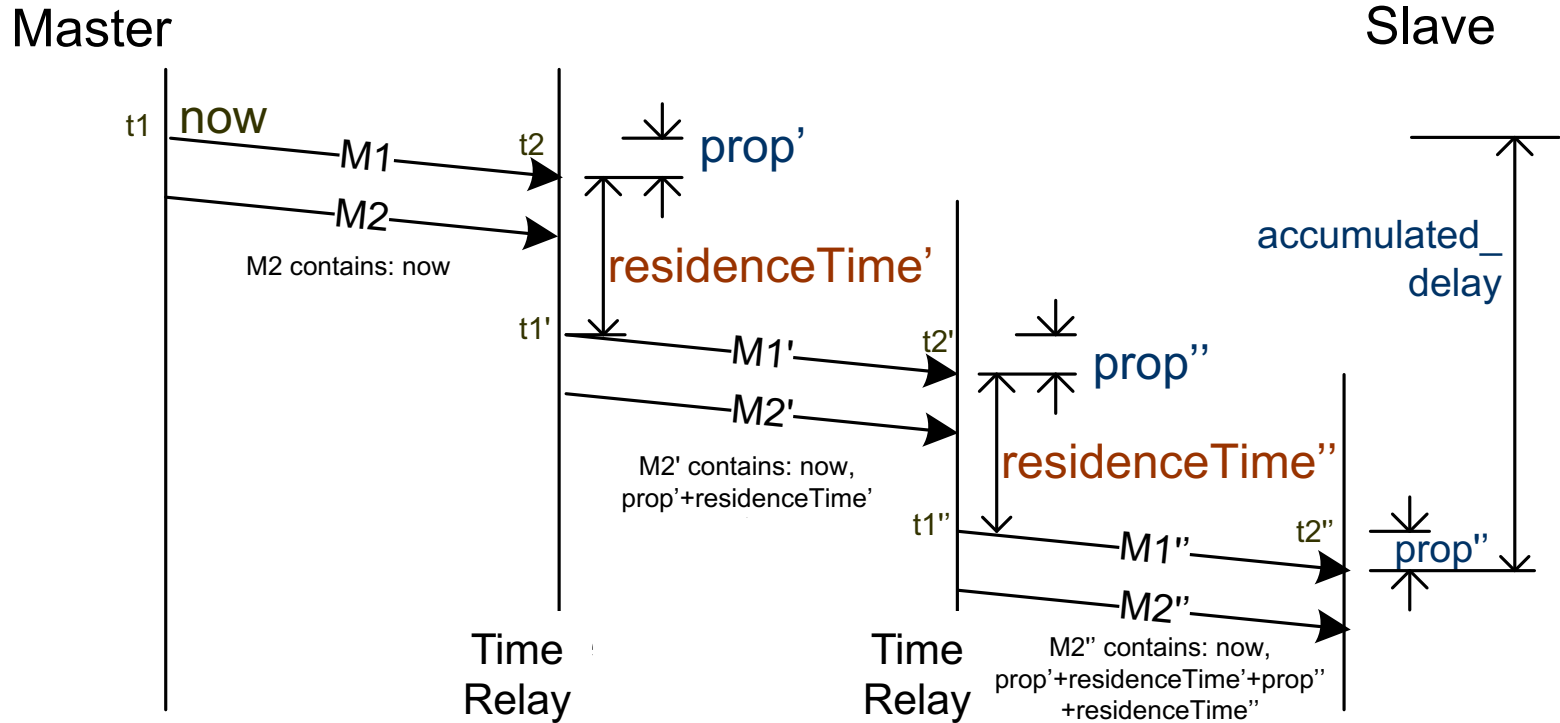
Grand Master initiates timing message, shown as M1, every Sync Interval

At each hop, propagates *now* and measures the actual delay of M1:

delay = prop + residenceTime * rateRatio

And carries the *accumulated_delay* in another message, M2

– Using two messages eliminates real-time processing requirements

Slaves compute: *currentTime at t2'' = now* plus the *accumulated_delay*

Note: Message interval on each link may be different

Measurements summarized

Accurate local egress and ingress timestamps captured in the MAC/PHY

Timestamps from special time-measurement frame used to compute:

- **Link delay (per slave port)**
- **Next-neighbor rate ratio (PPM offset to link partner)**

When the GM sends "now", the following are accumulated down the tree:

- **Residence time (per transmitted time measurement frame)**
- **rateRatio (PPM relative to the GM)**

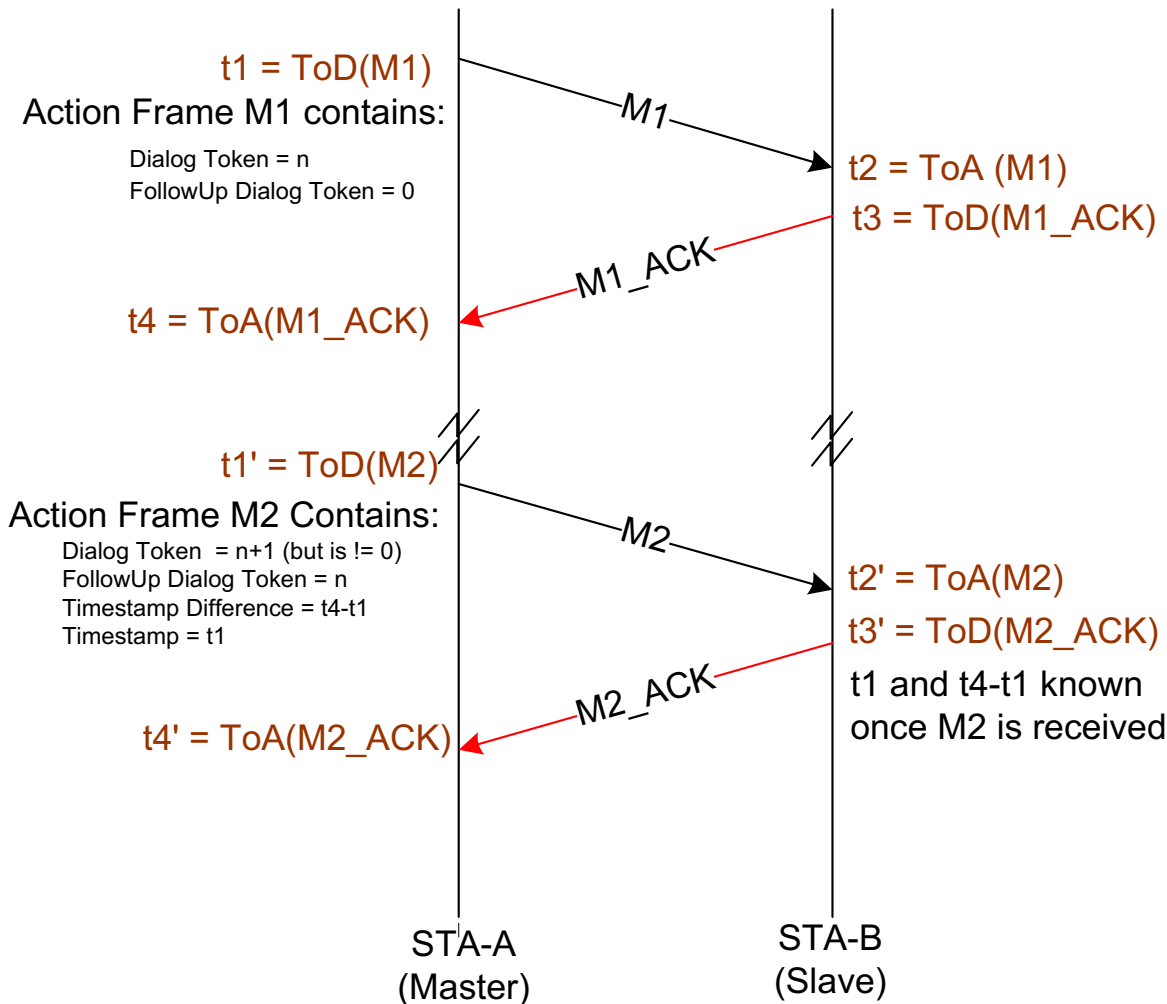**These measurements described in 802.1AS:**

– Clause 11: IEEE 802.3 (Ethernet)

– Clause 12: IEEE 802.11 (Wi-Fi)

– Annex E: MoCA, ITU-T G.hn

Agenda

- **802.1AS goals / use cases**

- **Overview**

- **Grand Master selection**

- **Time propagation**

  - Media-independent

  - Media-dependent

    - 802.11 links

    - 802.3 links

    - Other CSN links

- **Time *inside* the system**

Kevin B. Stanton

802.1AS over 802.11 links (Using the 802.11 TIMINGMSMT protocol)

t1 = ToD(M1)

Action Frame M1 contains:

Dialog Token = n
FollowUp Dialog Token = 0

M1

t2 = ToA (M1)
t3 = ToD(M1_ACK)

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

M2

t2' = ToA(M2)
t3' = ToD(M2_ACK)

t1 and t4-t1 known
once M2 is received

M2_ACK

t4' = ToA(M2_ACK)

STA-A
(Master)

STA-B
(Slave)

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:
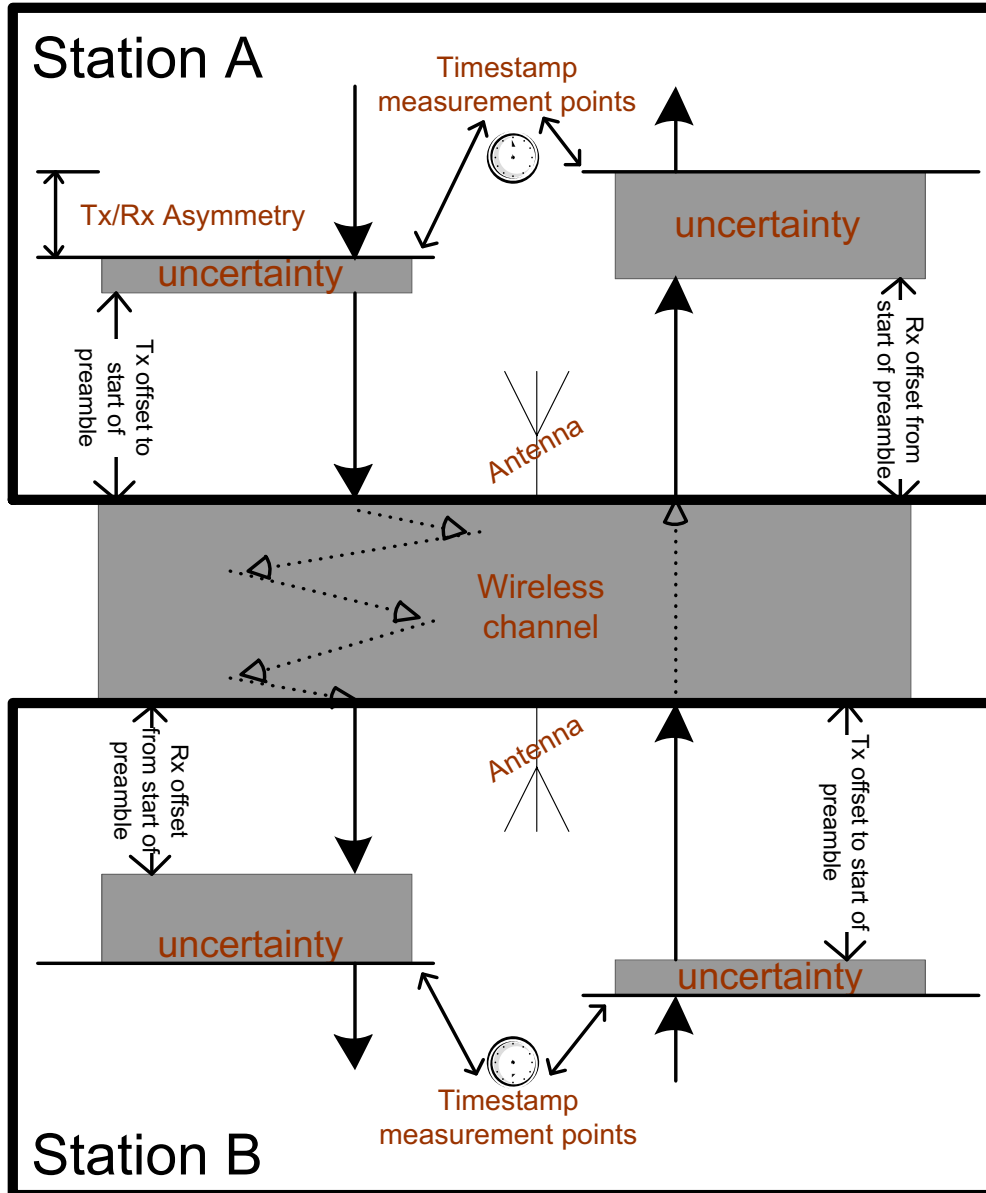
```
neighborRateRatio =
    (t1'-t1)/(t2'-t2)

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

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

[note: rateRatio is also applied]

Accounting for round-trip 802.11 path asymmetry

Station A

Timestamp measurement points

Tx/Rx Asymmetry

uncertainty

uncertainty

Tx offset to start of preamble

Rx offset from start of preamble

Antenna

Wireless channel

Antenna

Rx offset from start of preamble

Tx offset to start of preamble

uncertainty

uncertainty

Timestamp measurement points

Station B

The Antenna is the *reference* for Rx and Tx timestamp measurements

Each system knows the delay between the antenna and where the actual Rx and Tx timestamps are captured
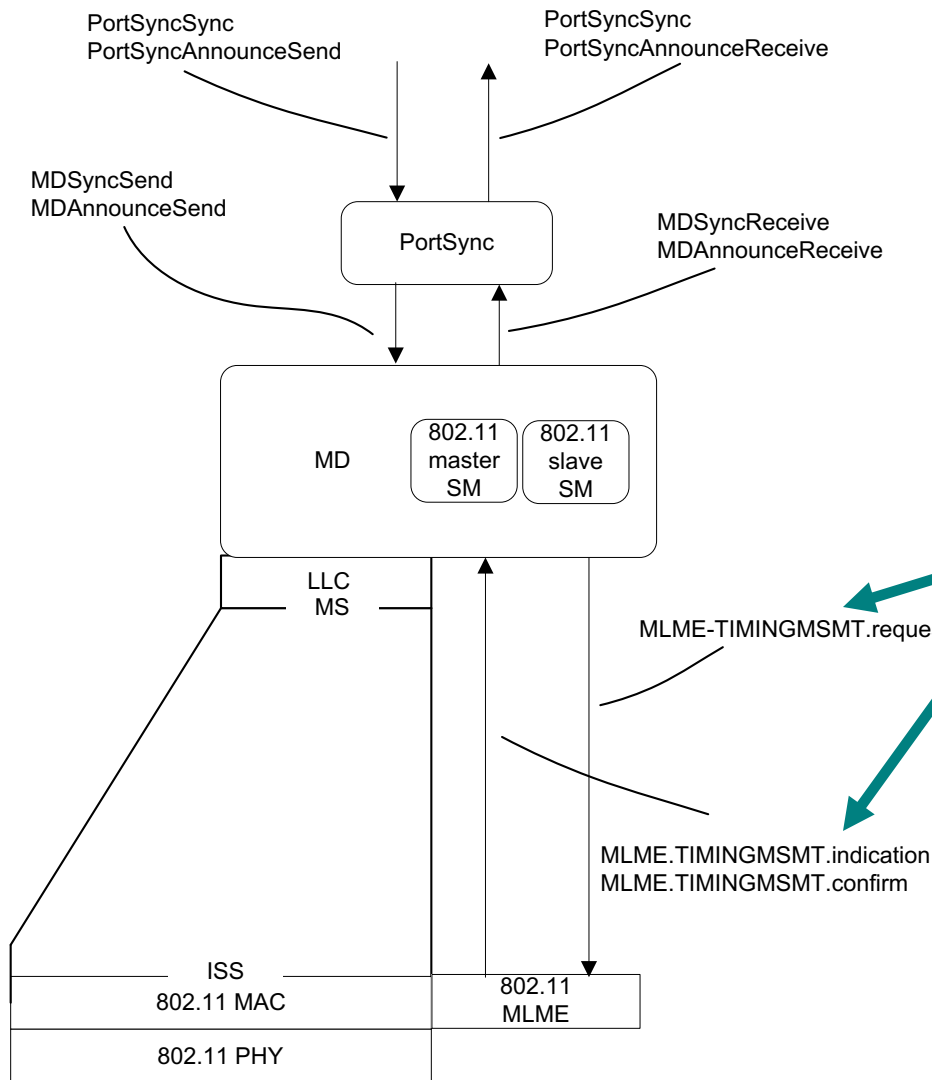
Each timestamp includes some uncertainty

The channel also introduces path asymmetry (and additional uncertainty)

In practice, the difference between the Rx offset and the Tx offset in each system is sufficient information

The rest appears as fixed channel delay and channel uncertainty

802.11 links in the 802.1AS architecture

PortSyncSync
PortSyncAnnounceSend

PortSyncSync
PortSyncAnnounceReceive

MDSyncSend
MDAnnounceSend

PortSync

MDSyncReceive
MDAnnounceReceive

MD

802.11 master SM

802.11 slave SM

LLC MS

MLME-TIMINGMSMT.request

MLME.TIMINGMSMT.indication
MLME.TIMINGMSMT.confirm

ISS
802.11 MAC

802.11 MLME

802.11 PHY

PortSync is per-port, media-independent

MD is per-port media dependent

802.11-2012 defines these MLME.TIMINGMSMT primitives

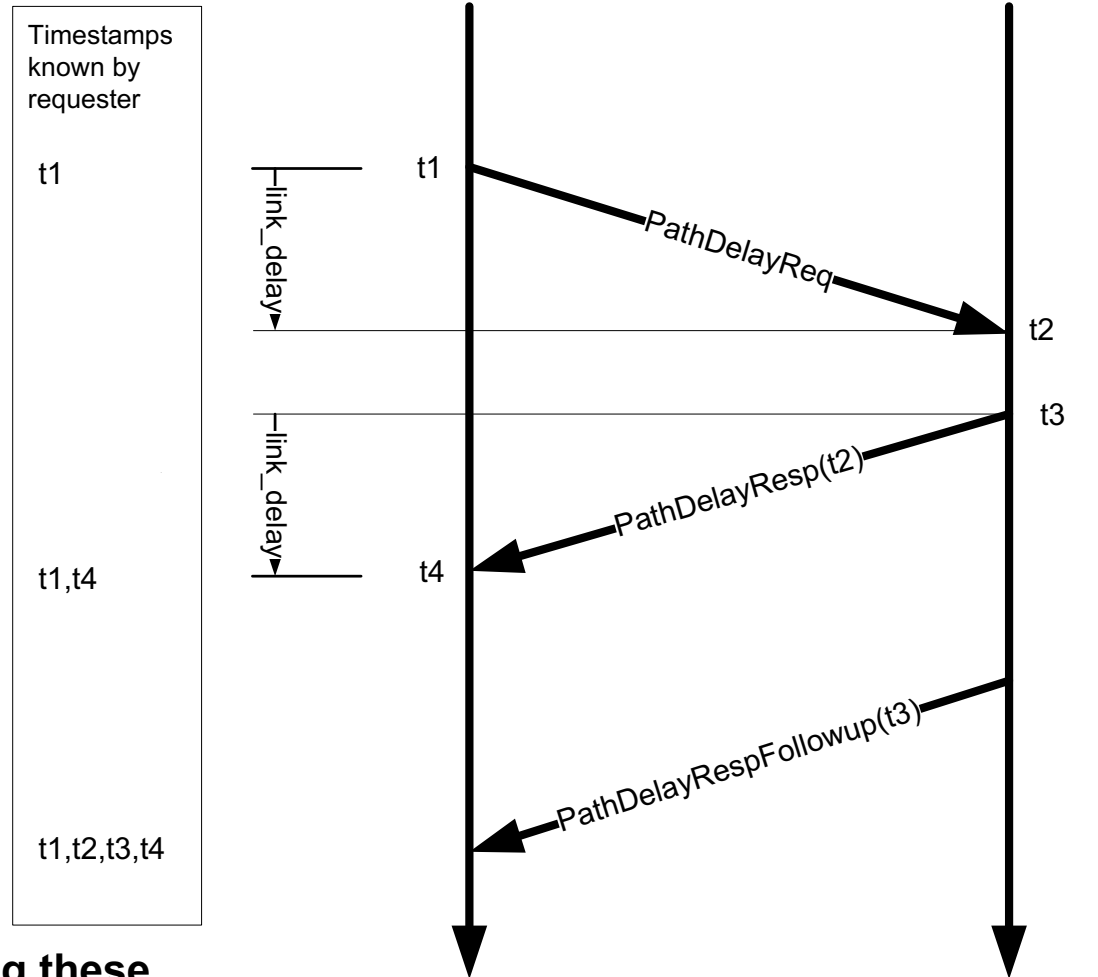**Measure link delay:**

1. **Requester schedules PathDelayReq for transmission**
2. **As it passes out the PHY, t1 is captured**
   Using the master's free-running clock
3. **Time t2 captured as passes from PHY to MAC**
   Using the slave's free-running clock
4. **Responder schedules PathDelayResp for transmission, sends t2**
5. **Timestamps t3 and t4 captured**
   Using local free-running clocks
6. **PathDelayRespFollowup carries t3 to requester**

If link delay is fixed & symmetric:
 link_delay = [(t4-t1) – (t3-t2)] / 2

NeighborRateRatio is computed using these and previous timestamps

Requester   Responder

Timestamps known by requester

t1

link_delay

t1,t4

t1,t2,t3,t4

t1

PathDelayReq

t2

t3

PathDelayResp(t2)

link_delay

t4

PathDelayRespFollowup(t3)

802.3 protocol (step 2 of 2)

Synchronize clocks

1. **Master schedules Sync for Tx**
2. **As it passes out the PHY, t1 captured**
   Using master's free-running clock
3. **Time t2 captured when it arrives**
   Using the slave clock
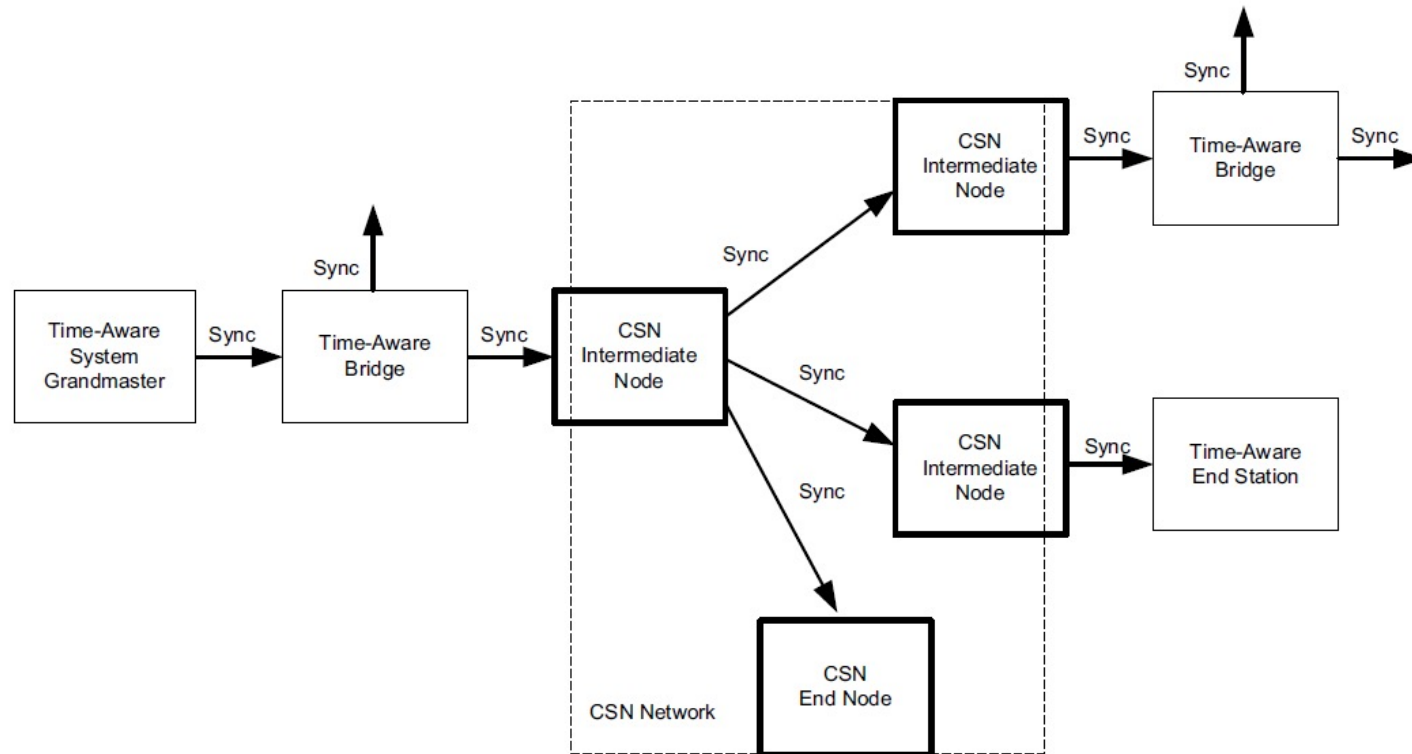4. **FOLLOWUP carries t1 to slave**

If link delay is fixed & symmetric:

Slave's clock offset
 = t2 – t1 – link_delay

Note: Network infrastructure devices do this too, and communicate the 'residence time' per Sync in the FOLLOWUP frame

Master   Slave

Timestamps known by slave

t1
link_delay
Sync
t2 — t2
Followup(t1)
t1
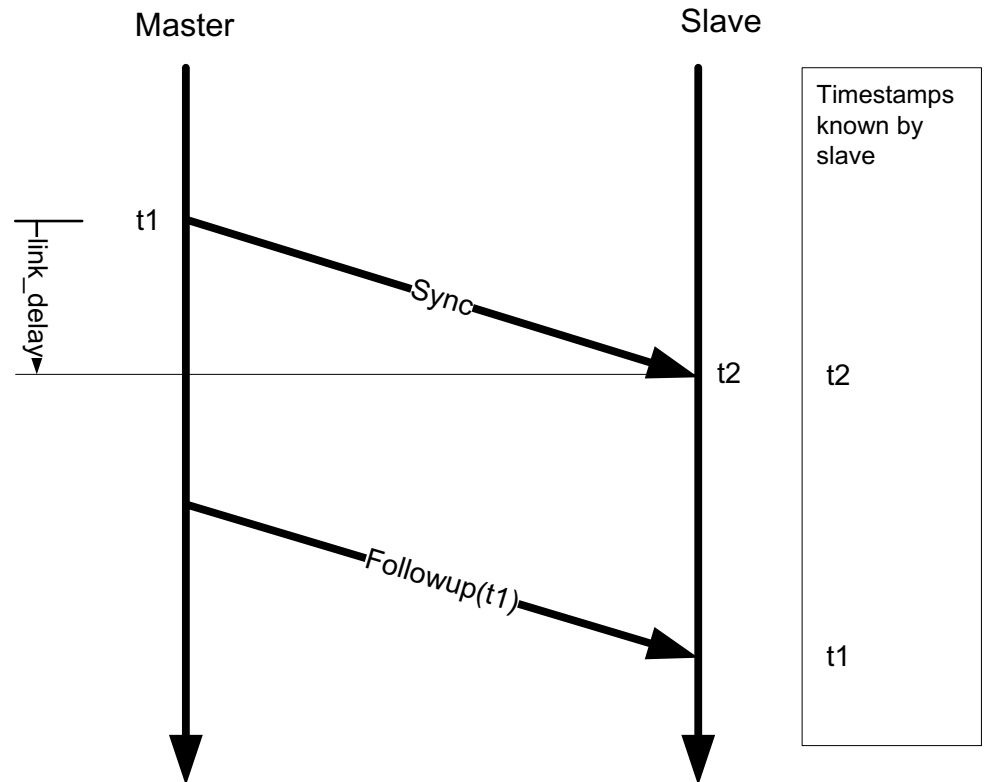
Coordinated Shared Network (CSN) protocol

- **CSNs use the same protocol as does 802.3 (Ethernet)**
- **Examples of CSNs:**
  - MoCA : Coax networking specification
  - ITU-T G.hn: Coax and power-line networking standard
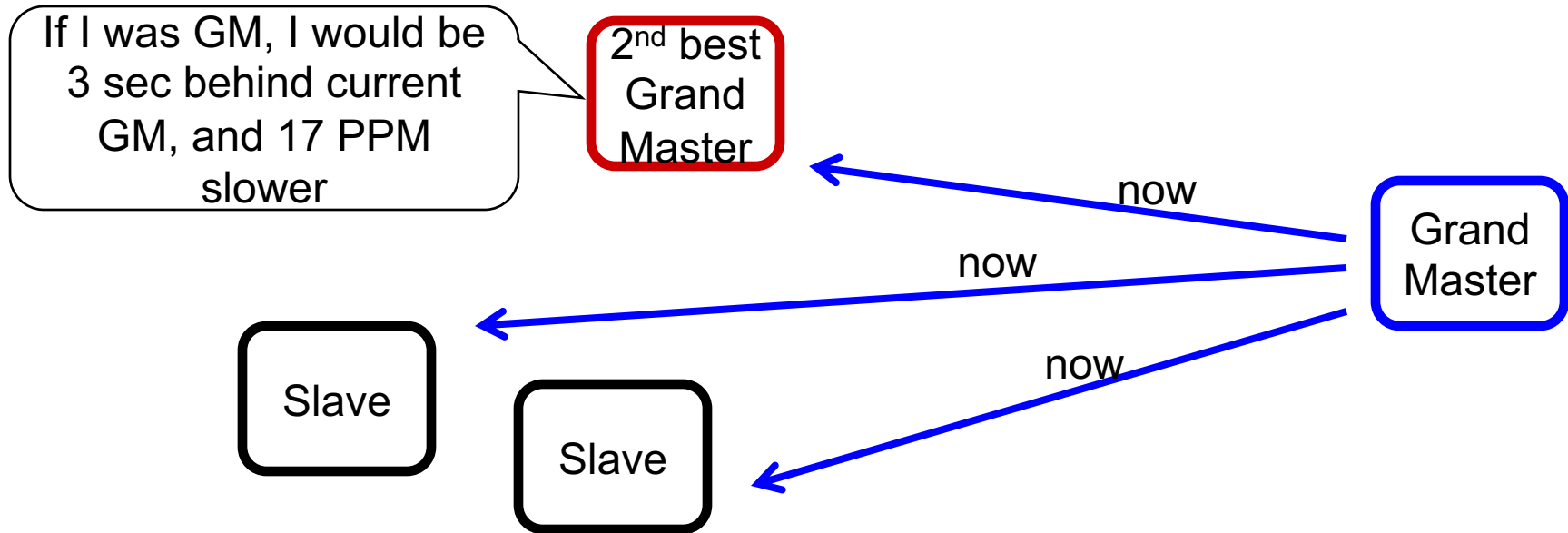- **This is described more in Annex E of 802.1AS**

Miscellaneous Details

Default Message Intervals defined by the standard

- **Announce sent once per second per Port**

- **Timing Information sent eight times per second**

  – For 802.11: Eight Action Frames per second (encapsulated Followup)

  – For 802.3 : Eight Sync and eight Followup frames per second

- **For 802.3, PDelay performed once per second**

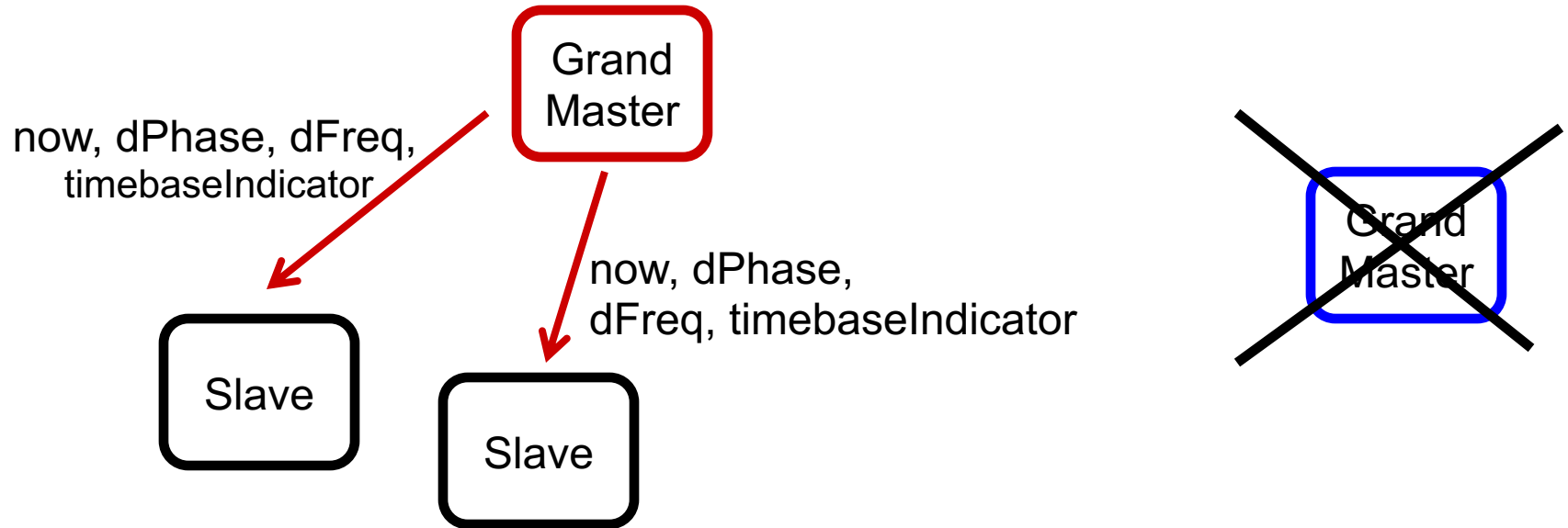 – For 802.3: Three frames per second in each direction per link

Non-default are intervals defined elsewhere

Handling of discontinuities

If I was GM, I would be 3 sec behind current GM, and 17 PPM slower

2nd best Grand Master

Grand Master

now
now
now

Slave

Slave

- **All potential GMs know their own time offset and intrinsic frequency offset (rateRatio) from the current GM**
- **Time offset may be close to zero (unless it knows UTC-traceable time, e.g., from GPS)**

Handling of discontinuities – new GM

now, dPhase, dFreq, timebaseIndicator

Grand Master

Slave

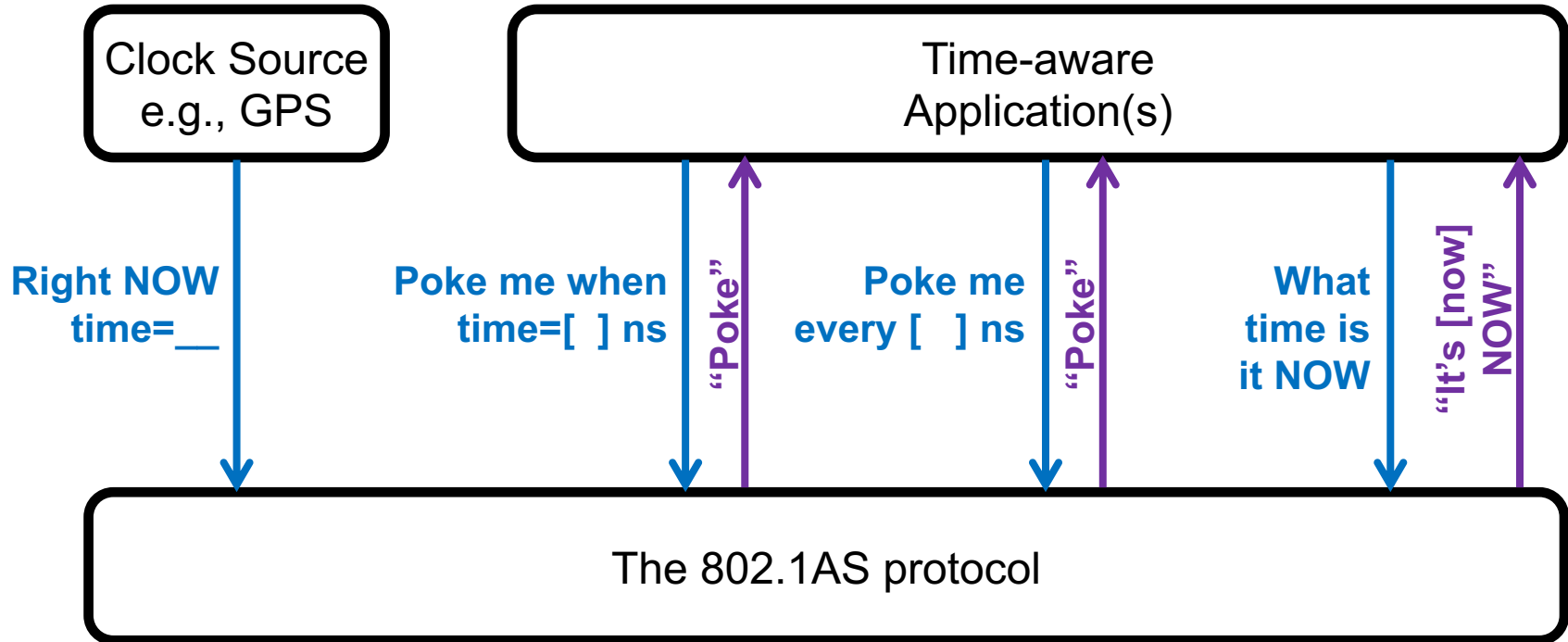now, dPhase, dFreq, timebaseIndicator

Slave

Grand Master

- **The new GM communicates extra info along with the time**
  - TimebaseIndicator (increments after each discontinuity)
  - dPhase: Last phase change (relative to the previous GM)
  - dFreq: Last frequency change (relative to the previous GM)
- **A GM also does this if it experiences a step change**
  - E.g. GPS regains lock or user manually sets the PTP time in the GM

Agenda

- **802.1AS goals / use cases**
- **Overview**
- **Grand Master selection**
- **Time propagation**
  - Media-independent
  - Media-dependent
- **Time *inside* the system**

Kevin B. Stanton

Four abstract application interfaces

```
Clock Source          Time-aware
e.g., GPS             Application(s)
```

Right NOW
time=___

Poke me when
time=[ ] ns

"Poke"

Poke me
every [   ] ns

"Poke"

What
time is
it NOW

"It's [now]
NOW"

The 802.1AS protocol

In 802.1AS these are called…

– ClockSourceTime
– ClockTargetTriggerGenerate
– ClockTargetClockGenerator
– ClockTargetEventCapture   …respectively

Regenerating the Talker's audio clock

Talker

Network

Frame 2 with 5 more audio samples. "The 1st sample should be presented at **t**=5**8**000.00**3**us"

Frame 1 with 9 audio samples. "The 1st sample should be presented at **t**=5**7**000.000us"

Listener

Listener

Listener: "Oops—I presented10th sample 0.00**9**us late.
It's OK—I'll just increase the D/A clock frequency slightly"

Audio samples Rendered

10

0

57000.000

58000.0**12**

t, actual 802.1AS time (in us)

5**8**000.0**12** us
— 5**8**000.00**3** us
0.00**9** us

- **Notes:**
  - Time (**t**) is 802.1AS time, known to all talkers & listeners
  - Nominal audio sample rate known beforehand, e.g.10KHz, 100us/sample above
  - 802.1AS time is "seconds since Jan 1, 1970 TAI" with precision of 1 nanosecond
  - This allows an ARBITRARY number of independent media clocks <u>simultaneously</u>
  - The Grand Master need not be the Talker

Relating 802.1AS time to media clock

- **"Presentation Time" added to header of media packets**
  - The Presentation Time is an 802.1AS time
  - Multiple methods for carrying Presentation Time
    - Layer-2: IEEE Std 1722™-2011 (based on IEC-61883)
    - Layer-3/IP: See IETF draft avtcore-clksrc (RTP Extension)
    - Layer-3/TCP/HTTP: TBD

- **Media is buffered in the renderer until the Presentation Time**
  - In some cases, thanks to M*RP and FQTSS (802.1Qat/Qav), Listener buffer size can be minimized
    - But this is out of scope of 802.1AS

Future work

- ## As of July 2014:

  – An amendment (soon, a revision) of 802.1AS, is under development to add "Enhancements and performance improvements"

    - More details can be found at
      http://www.ieee802.org/1/pages/802.1ASbt.html

References

Stds Available Free of Charge
http://standards.ieee.org/about/get/

- **IEEE Std. 1588™-2008**
  - The Definitive Precision Time Protocol (PTP) Definition
- **IEEE Std. 802.1AS™-2011**
  - Clause 12: Media-dependent layer specification for IEEE 802.11 links
  - Also, Technical and Editorial Corrections are published in
    - 802.1AS-2011 Corrigendum 1-2013
- **IEEE Std. 802.11™-2012**
  - Clause 6.3.57: Timing Measurement
  - Clause 10.23.5: Timing Measurement Procedure
- **API definition for use of 802.1AS (and related standards)**
  - http://www.avnu.org/files/static_page_files/C5E0B5F8-1D09-3519-ADB32F1F88E6C057/AVnu_SWAPIs_v1.0.pdf

Backup