IEC/IEEE 60802
Synchronization requirements and solution examples

Update after September 2020 interim

Günter Steindl (Siemens AG)
Scope

Industrial automation systems used in the areas of
- Factory automation
- Process automation
- Power generation
- Public transportation
- Building automation
- ...

... do have different environmental requirements.

The environmental conditions from these areas together with the device lifetime and the basic Ethernet PHY frequency quality are defining the requirements for the oscillators.

Many industrial automation products are build based on just one oscillator, which is the only source to drive a local time.

This applies to grandmaster, PTP relay and PTP end instance in this area.
Basic Requirements

Lifetime\(^1\) > 10years
\(df/dt \leq \text{“Area specific”}\)
Upper bound to \(f/f_N\) <= “Link speed”

Requirements:
1) For the whole lifetime of the device under all supported environmental conditions, the frequency shall stay in the “upper bounds” and shall not change faster as stated by \(df/dt\).
2) The sync deviation shall stay under all above stated conditions below |1µs|

<table>
<thead>
<tr>
<th>Data rate</th>
<th>Upper bound(^2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10/100Mbit/s</td>
<td>100ppm</td>
</tr>
<tr>
<td>1Gbit/s</td>
<td>50ppm</td>
</tr>
<tr>
<td>&gt;1Gbit/s</td>
<td>??? (assumed to be &lt;= 50ppm)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Public Transportation</th>
<th>Factory Automation</th>
<th>Process Automation</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>(df/dt)</td>
<td>&lt; 3ppm</td>
<td>&lt; 0,3ppm</td>
<td>&lt; 0,3ppm</td>
<td>??</td>
</tr>
</tbody>
</table>

1) Typical value for industrial automation. The real value depends on the product.
2) Given by IEEE802.3 MAU type requirements.
Constraints

There are a lot of sources for frequency deviation affecting error contribution of Grandmasters, PTP relays and PTP end instances. They are all just looked at summed up for the purpose of this contribution.

Production deviation, temperature deviation, aging, voltage ripple, shock and vibration all together shall not create a frequency deviation greater than the “upper bounds” from the previous slides.
Same is valid for the speed of frequency change which shall not be greater than “df/dt” from the previous slides.

The values are not derived from measuring one oscillator – they need to be guaranteed by the oscillator vendor.

Assumption:
Worst case happens if both, the grandmaster and the end-instances are moving from the opposite endpoints of the frequency with the maximum allowed df/dt to the other endpoints, undetected due the transmission delay of the sync message, while still staying inside the stated f/fN limit.

Working Clock and Global Time:
The stated constraints are valid for both.
The IEC/IEEE 60802 requirements for Working Clock seems only achievable with optimizations of the IEEE802.1AS model.
Parameter assumptions for simulation

- Grandmaster
- PTP end instance
- PTP relay
- Link delay

Constraints:
No oscillator ever shows a df/dt of more than the stated limits. Thus, sinusoidal behavior shall be used for the simulation. Maximum values derived from the usage of Gigabit data rate applies.
Assumption for simulation

Grandmaster

- Oscillator
  - $f/f_N \pm/\mp 50$ ppm

- Behavior of the local oscillator
  (a) random movement with max $df/dt$
  (b) use complete range with max $df/dt$
    - (I) $df/dt = 3$ ppm/s [start value]
    - (II) $df/dt = 0,3$ ppms/s [fallback]

- Sync interval
  - 31,25 ms

- Timestamp precision (500 MHz clock)
  - 2 ns

- Origin Time precision (500 MHz clock)
  - 2 ns

Additional error sources are ignored for this simulation.
Assumption for simulation
PTP end instance / PTP relay

- Oscillator
  - \( f/fN \pm 50 \text{ ppm} \)

- Behavior of the local oscillator
  (a) random starting point
  (b) worst case shift between GM and PTP end instance
  - (I) \( \frac{df}{dt} = 3 \text{ ppm/s} \) [start value]
  - (II) \( \frac{df}{dt} = 0.3 \text{ ppms/s} \) [fallback]

- Sync interval
  - 31.25 ms

- Timestamp precision (500 MHz clock)
  - 2 ns

- Origin Time precision (500 MHz clock)
  - 2 ns

- Sync residence time
  - 4 ms
  - syncLocked mode

- Rate calculation/compensation
  - Sliding window, window size seven sync messages
  - Filter median
  - Calculated rate values greater than \( |250\text{ppm}| \) are assumed to be invalid

- Offset calculation/compensation
  - KiKo (65) and KpKo (11) filter

- Number of hops
  - 100 (simply 100 bridged end stations in linear topology)
Assumption for simulation
Link delay

- $P_{\text{delay}}$ residence time
  - 10 ms

- Timestamp precision (500 MHz clock)
  - 2 ns

- Link asymmetry
  - 0 ns

- Additional error (PHY rx/tx delay variation)
  - +/- 8 ns

Errors in link delay compensation are as critical as errors in residence delay due to the required number of 98/100 hops.
Contributors feedback
(Measured Sync deviation value in real world with real products)

An in available products implemented IEEE 802.1AS model based solution using
- 31.25 ms sync interval, 1 s pdelay interval
- syncLocked mode
- Sync residence time (minimum 1ms, majority 4ms and maximum 10ms)
- 2.5 ns Timestamp granularity
- 100 hops, linear topology with GM as first device
- Sinusoidal 3 ppm df/dt for the GM
  Production, temperature, age, voltage variation, ... noise df/dt for the PTP end instances
- Upper bound f/fN 100 ppm

Additionally, a sliding window using the sync messages for GM rate calculation in the PTP relays/instances is implemented.

**Measured values** based on this setup
Sync deviation measured (at the sync out signal pins) between GM and any of the PTP instances is below |300 ns|
Model assumption for simulation
Basic timer model

The used timer model of this contribution is build as shown in the figure on this slide.

The rate compensated frequency (rate to the GM) is used for the 64 bit continuous running timer and as basis for the PTP time of the PTP end instance.

-> 64 bit continuous running timer used to rate compensate residence time

The offset correction is done by an additional rate compensation for the PTP end instance time.

-> used for PTP time

Examples:
Rate compensation
Entity which contains a register which defines the number of ticks after which instead of +1, either +0 or +2 is done.

Offset compensation
Entity which contains two registers
- one which defines the number of ticks after which instead of +1, either +0 or +2 is done.
- one which defines the overall number of “correction” actions to be done
PTP relay model

The PTP relay uses the sliding window (window size seven sync messages (210ms)) calculated and filtered (median over the last seven calculated values) rate to the GM for the residence time of the sync message.

Editors note: Median and not arithmetic mean is used.
PTP end instance model

The PTP end instance uses the same model as the PTP relay to calculate the rate.

Additionally, the offset between the GM and the end instance time is calculated and an offset correction factor calculated.

Rate = Frequency deviation between GM and PTP end instance
Offset correction factor = Correction (faster/slower) of the frequency used for the PTP end instance time.

The Offset correction factor is calculated by the PI filter of the PTP end instance to minimize the offset between GM and PTP end instance.

The correction speed (maximum Offset correction factor) is limited due to the application timing requirements. Otherwise, local time intervals would get “too small” or “too large”.

New rate(PTP end instance time) = Oscillator + Rate + Offset correction factor
Is used to drive the PTP end instance timescale.

Both, Rate and Offset correction factor are calculated (PI controller with Kp/Ki) and updated with every received sync frame following the stated model.

<table>
<thead>
<tr>
<th></th>
<th>Maximum Offset correction factor</th>
<th>Kp / Ki (KpKo/KiKo)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Working Clock</td>
<td>&lt; 250ppm</td>
<td>11 / 65</td>
</tr>
<tr>
<td>Global Time</td>
<td>&lt; 250ppm</td>
<td>11 / 20</td>
</tr>
</tbody>
</table>
Architecture model
Free running time

Syntonized time

Synchronized time

Offset compensation

Control loop

Rate compensation

Rate calculation and filtering

Residence time value

Pdelay value

TC sync forwarding

Oszillator

Rate
  a) From sync frame (peer2peer)
  b) Rate calculated from sync frame

Port 1

Port 2

Unrestricted
Model constraints

PTP relay
PTP relay instances calculate and use the rate for the supported timescales. Calculation is done by “median over the last seven calculated values” as shown on previous slides. The syntonized time is used to time stamp the sync messages. This concept reduces the effort at the sync forwarding process (adding local delays) to subtraction and addition of integer values.

PTP end instance
PTP end instances maintain a synchronized time for the supported timescales. The maintenance is done by comparing the syntonized time driven and offset compensated local synchronized time with the assumed GM time from the sync message. The new offset compensation value is calculate by a control loop using “KiKo and KpKo” as shown on previous slides.
**GM, per timescale:**
- monotonous running rate controlled timer

**Per timescale:**
- monotonous running rate controlled timer
- used to timestamp both, sync and Pdelay messages, to reduce rate handling at the forwarding process to simple subtraction and addition

---

**Neighbor:**
- monotonous running timer

**Neighbor:**
- monotonous running timer
- used to timestamp both, sync and Pdelay messages, calculating residence and pdelay values need advanced math

---

**Synchronized time**
- End to end Rate calculation
  Calculation interval ~210ms
  Filter “Median”

**Syntonized time**
- Peer to peer Rate calculation
  Calculation interval :=
  Pdelay transmit interval
  Filter ???

---

**Free running time**
- Peer to peer Rate calculation
  Calculation interval :=
  Pdelay transmit interval
  Filter ???

**Free running time**
Synchronization startup time
Open issues

Short startup time, meaning the time after the Working Clock maintained by a PTP end instances achieves the state “stable inside the $|1 \mu s|$ deviation window”, is an automation requirement.

Today, excluding initial Pdelay measurement, each hop consumes around 1.4 s (rate and offset are calculated and compensated).

“accumulatedRateRatio” in addition to the sync message based rate and offset calculation may offer optimization capabilities.

Wrong “stable inside the $|1 \mu s|$ deviation window” detection shall be avoided.
Model constraints
Pdelay

Pdelay
Pdelay measurement do have some error sources which contribute errors (offset and variation). The variation do need some repetition together with filtering to find a good (containing the “zero” mean error) value for further use. (wireline connection) Offset need to be reduced/avoided by hardware design or administrative compensated.

Speedup start of synchronization
After power-on or link-up use a Pdelay interval of 250ms for the first 2s and then switch back to 1s Thus, after around 1,5s (assuming 6 consecutive Pdelay measurements create a value as stated above) the Pdelay value seems to be good enough.
Model constraints

Sync

Rate
Using all, the peer rate value, the rate value from the sync frame and the self calculated peer value allows an early good estimation of the GM rate and thus, a faster synchronization.

Speedup start of synchronization
Synchronization relies on stable/correct Pdelay values. Having these after 1,5 s, together with the above rate handling, should allow a sync deviation of less than $|1\mu s|$ after 1,5s.

Goal:
Pdelay is running since power-on.
GM starts sending sync messages
Measure sync-out signals between GM and each end instance
Check whether they do have a deviation of less than $|1\mu s|$ after 1,5s
Questions?