Symbols and Nomenclature

- GrandMasterClock
  - ClockSource
    - ClockMaster
      - PortSync
  - LocalClock
    - SiteSync

- TimeAwareBridge
  - ClockTarget
    - ClockSlave
      - ClockMaster
    - PortSync
  - SynchronizedClock
    - PortSync

- TimeClient
  - ClockTarget
    - ClockSlave
  - LocalClock
    - PortSync

T ... LocalClock (free running)
t' ... SynchronizedClock (syntonous with GM)
Problem Description

- gPTP-Sync-Messages (t’) sent from GM to the Bridges/Clients
- pDelay-Messages based on Local-Clock (T) information
- How can we ensure – in safety terms – at any one point in Real-Time (R), the GM and all (relevant) clients have synchronized to the same clock-counter value, within an accepted accuracy?
  \[ t(R) \approx t'_A(R) \approx t'_B(R) \]
**Redundant GM/Client Connections**

How can Sync between GMs be assured?

- System can only start up, if both GMs are operational (and synchronized) -> reduced availability
- Bridges can still introduce constant offset to one domain
- Clients can compare domains locally but cannot compare each other

Only loss is a defined error, deviation is not
The “Byzantine” Bridge Error

Worst case: A Bridge introduces a constant offset $\Delta_{\text{error}}$ from the start-up (not a transient change)
Delay within the Timing-Stack

Does the application (ClockTarget) actually get the correct time?

Stack can introduce a delay

Local-Clock time-stamping can introduce a delay
Predict next originTimestamp

\[
T(R1 + pDelay) = t'(T) \pm \Delta_{error}
\]

\[
t'_R1(T) = t'(T) \pm \Delta_{error}
\]

before calling updateSlaveTime()

\[
t'_\text{master}(R_{syn}) \approx t'_R1(T(R_{syn}))
\]

compare prediction

SYNC/FollowUp
Use pDelay to probe SynchronizedClock

- **TimeAwareSystem**
  - R
  - $t'_S(T_S)$
  - $T_S(R_{req})$
  - $t'_S(T_S(R_{req}))$
  - $T_S(R_{req})$
  - $t'_S(T_S(R_{resp} + pDelay))$
  - $T_S(R_{resp} + pDelay)$
  - SynchronizedClock
  - **pDelay Source**

- **TimeAwareSystem**
  - $T_T$
  - LocalClock
  - $t'_T(T_T)$
  - $T_T(R_{req} + pDelay)$
  - $T_T(R_{resp})$
  - $t'_T(T_T(R_{resp}))$
  - SynchronizedClock
  - **pDelay Target**

- **compare in synchronized time**

IEEE contribution
Why neighbourRateRatio from pDelay?

Improved Start-Up time:
• Can start on each link after link-up
• pDelay and neighbourRateRatio are already known when first Sync-Message arrives
Stations on a Link can monitor each other

\[ t'_\text{master}(R_{\text{syn}}) \approx t'_{R1}(T(R_{\text{syn}})) \]

\[ t'_S(T_S(R_{\text{req}}) + p\text{Delay}) \approx t'_T(T_T(R_{\text{req}}) + p\text{Delay}) \]

\[ t'_S(T_S(R_{\text{resp}} + p\text{Delay})) \approx t'_T(T_T(R_{\text{resp}}) + p\text{Delay}) \]

**compare prediction**

**compare in synchronized time**
List of Data-Points per Link

<table>
<thead>
<tr>
<th>Event Message</th>
<th>Transmit time-stamp</th>
<th>Receive time-stamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sync-Message</td>
<td>$T_{\text{master}}(R_{\text{sync}})$&lt;br&gt;$t'<em>{\text{master}}(T(R</em>{\text{sync}}))$ [also in FollowUp-Msg]</td>
<td>$T_{\text{client}}(R_{\text{sync}}+\text{pDelay})$&lt;br&gt;$t'<em>{\text{client}}(T(R</em>{\text{sync}}+\text{pDelay}))$</td>
</tr>
<tr>
<td>pDelayRequest-Message</td>
<td>$T_S(R_{\text{req}})$&lt;br&gt;$t'<em>S(T_S(R</em>{\text{req}}))$</td>
<td>$T_T(R_{\text{req}}+\text{pDelay})$&lt;br&gt;$t'<em>T(T_T(R</em>{\text{req}}+\text{pDelay}))$ [also in pDelayFollowUp-Msg]</td>
</tr>
<tr>
<td>pDelayResponse-Message</td>
<td>$T_T(R_{\text{resp}})$&lt;br&gt;$t'<em>T(T_T(R</em>{\text{resp}}))$ [also in pDelayFollowUp-Msg]</td>
<td>$T_S(R_{\text{resp}}+\text{pDelay})$&lt;br&gt;$t'<em>S(T_S(R</em>{\text{resp}}+\text{pDelay}))$</td>
</tr>
</tbody>
</table>

Autosar has added interfaces to record these time-tuples, should IEEE802.1AS and IEEE1588 follow? The knowledge of Synchronized Time at each port requires Boundary Clocks

Do NOT specify use of the time-tuples in IEEE!

Boundary Clock: create new OriginTimeStamp for transmitted Sync-Message
Transparent Clock: keep incoming OriginTimeStamp, add CorrectionField in Local-Time to transmitted Sync-Message
THANK YOU

ETHERNOVIA  max.turner@ethernovia.com