A Simulation Model of IEEE 802.1AS gPTP for Clock Synchronization in OMNeT++

Henning Puttnies, Peter Danielis, Enkhtuvshin Janchivnyambuu, Dirk Timmermann
University of Rostock, Germany
1. Motivation

- Real-time Ethernet systems
  - No open standard established
  - Only proprietary solutions (expensive)

- A standard-based approach is required
- IEEE 802.1 Time-Sensitive Networking (TSN) Task Group
- gPTP is a part of TSN standards (for sync)
2. Basics

- Overview of gPTP protocol

- Types of time-aware systems
  - End stations, bridges

- Types of ports
  - Master, slave, passive

- Time-aware systems only communicate gPTP information directly with other time-aware systems
  ➔ Hop by hop synchronization
2. Basics

- Best master clock selection (BMCS)

- All time-aware systems participate in BMCS
- Announce message: time-synchronization spanning tree vector
- Automatic changeover to a secondary grandmaster

```
<table>
<thead>
<tr>
<th>Master port</th>
<th>Slave port</th>
<th>Passive port</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>S</td>
<td>P</td>
</tr>
</tbody>
</table>
```
2. Basics

- Propagation delay measurement

\[ P_{\text{delay}} = \frac{(t_4 - t_1) - r \cdot (t_3 - t_2)}{2} \]

\[ r = \frac{f_{\text{requestor}}}{f_{\text{responder}}} \]
2. Basics

- Propagation delay measurement
  - Rate ratio

\[
r = \frac{t_{12} - t_{11}}{t_{22} - t_{21}}
\]

- Propagation delay measurement

- Rate ratio

\[
r = \frac{t_{12} - t_{11}}{t_{22} - t_{21}}
\]

- Propagation delay measurement

- Rate ratio

\[
r = \frac{t_{12} - t_{11}}{t_{22} - t_{21}}
\]

- Propagation delay measurement

- Rate ratio

\[
r = \frac{t_{12} - t_{11}}{t_{22} - t_{21}}
\]

- Propagation delay measurement

- Rate ratio

\[
r = \frac{t_{12} - t_{11}}{t_{22} - t_{21}}
\]

- Propagation delay measurement

- Rate ratio

\[
r = \frac{t_{12} - t_{11}}{t_{22} - t_{21}}
\]

- Propagation delay measurement

- Rate ratio

\[
r = \frac{t_{12} - t_{11}}{t_{22} - t_{21}}
\]

- Propagation delay measurement

- Rate ratio

\[
r = \frac{t_{12} - t_{11}}{t_{22} - t_{21}}
\]

- Propagation delay measurement

- Rate ratio

\[
r = \frac{t_{12} - t_{11}}{t_{22} - t_{21}}
\]

- Propagation delay measurement

- Rate ratio

\[
r = \frac{t_{12} - t_{11}}{t_{22} - t_{21}}
\]

- Propagation delay measurement

- Rate ratio

\[
r = \frac{t_{12} - t_{11}}{t_{22} - t_{21}}
\]
2. Basics: Transport of Sync. Information

- correctionField: Composed of propagation delay and residence time
- Slave: preciseOriginTimestamp + <delayToGM> ➔ Synced to GM time
3. Implementation

- **Scope of the project**
  - gPTP simulation model in OMNeT++ using the INET library
    - Integrate gPTP model seamlessly with other protocols from INET
    - Implement only time synchronization and propagation delay measurement
    - Best master clock not part of project
      Assumption: GM shall no be selected randomly
    - Implement simple clock with constant drift
3. Implementation

- Model of clock with constant drift

\[ \begin{align*}
T(t) &= t \\
C(t) &= at + b
\end{align*} \]
3. Implementation

- Model of clock with constant drift
3. Implementation

- **Model of gPTP functionalities**

  - Eth. interface with gPTP support: `EthernetInterfaceGPTP`
  - Simple module for gPTP functions: `etherGPTP`
3. Implementation

- Model of time-aware systems

- Simple module *tableGPTP*
- Simple module *clock*
- Compound module *EthernetInterfaceGPTP*
4. Evaluation: Simulation Setup

• Same setup as Lim et al.* (BMW + TUM)
• Evaluation:
  • Propagation delay measurement
  • Time difference to GM (before resynchronization)

Clock drift of time-aware systems in domain [ppm]

<table>
<thead>
<tr>
<th>Master</th>
<th>Bridge0</th>
<th>Bridge1</th>
<th>Bridge2</th>
<th>Slave0</th>
<th>Slave1</th>
<th>Slave2</th>
<th>Slave3</th>
<th>Slave4</th>
<th>Slave5</th>
<th>Slave6</th>
<th>Slave7</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>30</td>
<td>-15</td>
<td>20</td>
<td>-50</td>
<td>10</td>
<td>50</td>
<td>-5</td>
<td>-50</td>
<td>40</td>
<td>-15</td>
<td>-35</td>
</tr>
</tbody>
</table>

*Hyung-Taek Lim, Daniel Herrscher and Lars Volker “IEEE 802.1AS Time Synchronization in a switched Ethernet based In-Car Network”, IEEE VNC 2011
4. Evaluation: Propagation Delay Measurement

- Converge to 25 ns (absolute difference < 0.5 ns)
- Lim et al.: +/- 10 ns acceptable

<table>
<thead>
<tr>
<th>Node</th>
<th>Propagation delay[ns]</th>
<th>Error (%)</th>
<th>Absolute difference [ns]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slave 0</td>
<td>25.43</td>
<td>1.72%</td>
<td>0.43</td>
</tr>
<tr>
<td>Slave 1</td>
<td>25.43</td>
<td>1.72%</td>
<td>0.43</td>
</tr>
<tr>
<td>Slave 2</td>
<td>24.78</td>
<td>-0.88%</td>
<td>0.22</td>
</tr>
<tr>
<td>Slave 3</td>
<td>25.29</td>
<td>1.16%</td>
<td>0.29</td>
</tr>
<tr>
<td>Slave 4</td>
<td>25.29</td>
<td>1.16%</td>
<td>0.29</td>
</tr>
<tr>
<td>Slave 5</td>
<td>24.78</td>
<td>-0.88%</td>
<td>0.22</td>
</tr>
<tr>
<td>Slave 6</td>
<td>24.78</td>
<td>-0.88%</td>
<td>0.22</td>
</tr>
<tr>
<td>Slave 7</td>
<td>25.43</td>
<td>1.72%</td>
<td>0.43</td>
</tr>
<tr>
<td>Bridge 0</td>
<td>25</td>
<td>0.00%</td>
<td>0.00</td>
</tr>
<tr>
<td>Bridge 1</td>
<td>25.43</td>
<td>1.72%</td>
<td>0.43</td>
</tr>
<tr>
<td>Bridge 2</td>
<td>24.78</td>
<td>-0.88%</td>
<td>0.22</td>
</tr>
</tbody>
</table>
## 4. Evaluation: Time Difference to GM

- Time difference to GM (before resynchronization)
- As expected: e.g., for 125ms and +/- 50ppm $\Rightarrow$ +/- 6.25us

<table>
<thead>
<tr>
<th>Node</th>
<th>Time difference to GM before resynchronization in our implementation [µs]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sync interval 62.5 ms</td>
</tr>
<tr>
<td>Bridge 0</td>
<td>2.36</td>
</tr>
<tr>
<td>Slave 0</td>
<td>-3.12</td>
</tr>
<tr>
<td>Slave 1</td>
<td>0.63</td>
</tr>
<tr>
<td>Slave 7</td>
<td>-2.19</td>
</tr>
<tr>
<td>Bridge 1</td>
<td>-0.94</td>
</tr>
<tr>
<td>Slave 2</td>
<td>3.13</td>
</tr>
<tr>
<td>Slave 6</td>
<td>-0.94</td>
</tr>
<tr>
<td>Slave 5</td>
<td>2.50</td>
</tr>
<tr>
<td>Bridge 2</td>
<td>1.25</td>
</tr>
<tr>
<td>Slave 3</td>
<td>-0.31</td>
</tr>
<tr>
<td>Slave 4</td>
<td>-3.12</td>
</tr>
</tbody>
</table>
5. Conclusion

- We have contributed
  - Simulation model of gPTP
  - Models for time-aware systems: end-station and bridge
  - Simple clock model with constant drift
- Comparisons of results to literature
- Useful in simulating any networks based on the gPTP
- Entire system is publicly available*
- Future work: Utilize other the clock models

* https://gitlab.amd.e-technik.uni-rostock.de/peter.danielis/gptp-implementation
Thank you for your attention. Questions?