Talker Scheduled Traffic Support
for Ultra Low Latency

Johannes Specht, University of Duisburg-Essen
Markus Jochim, General Motors Company
Introduction

• The Time Aware Shaper (TAS) enables low latency and meets e.g. Automotive Requirements
• TAS requires End Stations and Bridges to operate in synchronized TDMA-like schedules
• This slide deck proposes Talker Scheduled Traffic Support (TSTS) as an alternative to TAS in bridges:
  – TSTS can simplify the scheduled traffic concept, ...
  – ... decrease complexity in bridges and ...
  – ... still meet ultra low latency Automotive Requirements
Assumptions

• 802.1 gets Preemption!
  – TAS would not be possible without preemption since Guard Windows would be way too large

• Preemption Performance
  – Preemption of a frame takes requires a reasonable short worst case delay from preemption until preemtting class can transmit, e.g. 84 byte times $t_{MaxPreemption}$
Assumptions

• Talker behavior stays the same as it is for Scheduled Traffic!
  – Talkers implement the Time Aware Shaper (TAS)
    Bridges implement TSTS instead of TAS in this proposal

  – All talkers sending low latency traffic are synchronized with good precision, e.g. 1 µs
    With TAS in bridges, imprecise or async. talkers couldn’t reach low latency: The frames of these talkers would be queued until the next TAS window

  – Talker always fulfill their contracts for scheduled traffic, e.g. period, phase and max. frame size
    If talkers would violate contracts, TAS in bridges couldn’t protect scheduled traffic of other (non contract-violating) talkers
Recap: Time Aware Shaper

### Legend
- **A**: Scheduled frame transmission start (after arrival on egress port)
- **A**: Frame transmission of A
- **B**: Lower priority frame transmission start (may be taken from queue)
- **B**: Frame or framelet transmission of B

<table>
<thead>
<tr>
<th>Case 1: No low priority transmission</th>
<th>Case 2: Long low transmission before scheduled transmission</th>
<th>Case 3: No frame for scheduled transmission</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sched. Traffic</td>
<td>Latency of A = 0</td>
<td>Latency of A = 0</td>
</tr>
<tr>
<td>Lower priority</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transm.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Cas 1: No low priority transmission

- Scheduled traffic (A)
- Lower priority traffic
- Transmission (A)
- Latency of A = 0

→ **No additional jitter per Hop!**

Case 2: Long low transmission before scheduled transmission

- Scheduled traffic (A)
- Lower priority traffic (B)
- Transmission (A, B)
- Latency of A = 0

→ **Bandwidth lost!**

Case 3: No frame for scheduled transmission

- Scheduled traffic (A)
- Lower priority traffic (B)
- Transmission (B, B)

March 17-22, 2013, Orlando, Florida, USA
Removing TAS and Preemption from bridges ...

Legend:
- A: Scheduled frame transmission start (after arrival on egress port)
- A: Frame transmission of A
- B: Lower priority frame transmission start (may be taken from queue)
- B: Frame or framelet transmission of B

Case 1: No low priority transmission
- Latency of A = 0

Case 2: Long low transmission before scheduled transmission
- Latency of A ≤ max. frame

Case 3: No frame for scheduled transmission

→ Large additional Jitter per Hop!

→ No Bandwidth lost!

18.03.2013
Basic Idea (first part): Only used mechanism in bridges: Preemption

Case 1: No low priority transmission
- Latency of A = 0
- \( \rightarrow \text{Small Jitter per Hop: } t_{\text{MaxPreemption}} \)

Case 2: Long low transmission before scheduled transmission
- Latency of A \( \leq t_{\text{MaxPreemption}} \)

Case 3: No frame for scheduled transmission
- \( \rightarrow \text{No Bandwidth lost!} \)

Legend:
- A: Scheduled frame transmission start (after arrival on egress port)
- B: Lower priority frame transmission start (may be taken from queue)
- B: Frame or framelet transmission of B
Basic Idea (second & last part): Always wait for $t_{MaxPreemption}$

<table>
<thead>
<tr>
<th>Case 1: No low priority transmission</th>
<th>Case 2: Long low transmission before scheduled transmission</th>
<th>Case 3: No frame for scheduled transmission</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sched. Traffic</td>
<td>Latency of A = $t_{MaxPreemption}$</td>
<td>Latency of A = $t_{MaxPreemption}$</td>
</tr>
<tr>
<td>Lower priority</td>
<td>$\Rightarrow$ No additional Jitter per Hop!</td>
<td>$\Rightarrow$ No Bandwidth lost!</td>
</tr>
<tr>
<td>Transm.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Latency of A = $t_{MaxPreemption}$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Legend**
- A: Scheduled frame transmission start (after arrival on egress port)
- A: Frame transmission of A
- B: Lower priority frame transmission start (may be taken from queue)
- B: Frame or framelet transmission of B

IEEE 802.1 Plenary Meeting
March 17-22, 2013, Orlando, Florida, USA
But why is waiting needed?

- If there is no need for preemption, transmission could start ASAP, i.e. at frame arrival on egress port ...
- ... then transmissions would accumulate a jitter of $t_{\text{MaxPreemption}}$ per hop.
- Considering multiple hops, this could lead to race conditions between multiple scheduled traffic streams.
Summarized: Egress Operation in Bridges

When a scheduled frame becomes ready for transmission:

1. **Preempt** the current lower priority frame if present and ...
2. **wait** a constant time of $t_{\text{MaxPreemption}}$ while **holding** transmission permission
3. **Send** the frame

**Note:**
Waiting is required per queue/port: traffic is scheduled and thus there’s no queuing like with e.g. best effort traffic
## Comparison: Latency

<table>
<thead>
<tr>
<th>Payload Size</th>
<th>TAS</th>
<th>Hops</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Payload Size</th>
<th>TSTS</th>
<th>Hops</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Payload Size</th>
<th>Comparison</th>
<th>Hops</th>
</tr>
</thead>
<tbody>
<tr>
<td>64 [Byte]</td>
<td>71.62%</td>
<td>65.43%</td>
</tr>
<tr>
<td>128 [Byte]</td>
<td>80.19%</td>
<td>75.22%</td>
</tr>
<tr>
<td>256 [Byte]</td>
<td>87.65%</td>
<td>84.18%</td>
</tr>
<tr>
<td>512 [Byte]</td>
<td>92.95%</td>
<td>90.82%</td>
</tr>
<tr>
<td>1024 [Byte]</td>
<td>96.21%</td>
<td>95.01%</td>
</tr>
</tbody>
</table>

### Assumptions:

- **100 Mbps**: no cable delays
- **no cut through (reception incl. IPG before forwarding)**
- **$t_{MaxPreemption} = 84$ bytes**: no forwarding delays

---

IEEE 802.1 Plenary Meeting  
March 17-22, 2013, Orlando, Florida, USA
Comparison: Jitter and Latency

The latency of TSTS is higher than the latency of TAS, however:

- TSTS fulfills the Automotive control data class requirements as presented by an OEM (cmp. [1]/green cells in prev. slide):
  - 100 µs Latency@5 Hops for 128 byte payload
- The ratio between TAS and of TSTS is not that bad, e.g. ~72% in the automotive example
- If the low latency frame (A) is missing, bandwidth utilization by lower priorities is higher (cmp. Case 3 in prev. slides)
- The mechanism is simple!

Comparison: Complexity

No need to synchronize egress ports ...
• ... bridges might even be time-unaware*
• The reduced jitter may enable further use-cases beside Automotive ultra low latency control applications, e.g. clock sync. across time-unaware parts of a network

Less gates/transistors on data plane in bridges:
• TAS requires time gates (special logic), Gate driver and gate event memory (in the best case a “large enough“ dedicated SRAM) per Port, mechanisms for gate event configuration, etc.
• The proposed mechanism requires a reasonable small FSM per TSTS queue per port
Comparison: Configuration

No egress data plane configuration:
The only parameter is $t_{\text{MaxPreemption}}$ which is a constant – could either be fixed by upcoming standards or at latest during manufacturing of a bridge

- **Plug and Play use-cases:**
  - No Protocols needed to adjust the data plane/egress ports during runtime, although ...
  - ... there may be the need for protocols like SRP to guarantee „sufficient remaining bandwidth“ for reserved traffic, BPDUs, etc., but ...
  - ... this can be handled during resource allocation on the control plane exclusively, e.g. by rejecting conflicting allocation attempts

- **Automotive (and other engineered) use-cases:**
  - No need to configure bridges during network integration (OEM/Tier-1)
  - No need to identify and specify additional TAS-requirements like the minimum number of Gate Events, etc. (OEM)
  - Freedom to use additional Clock Sync. Protocols beside 802.1AS with high precision
Summary & Conclusions

• TS Traffic Support in Bridges was proposed as an alternative to/simplification of TAS
• TS Traffic Support has the following key aspects:
  – End 2 End Latency is a bit higher than with TAS, but ...
  – Bandwidth utilization by lower classes can be a bit lower
  – ... Automotive Ultra Low Latency Requirements can be fulfilled
  – No data plane configuration in bridges
  – Complexity of bridge implementations seems to be low
  → maybe there is some chip area left for ingress policing ;-)

18.03.2013
IEEE 802.1 Plenary Meeting
March 17-22, 2013, Orlando, Florida, USA
Thank you for your Attention!

Opinions, Questions, Ideas?

Johannes Specht
Dipl.-Inform. (FH)

Dependability of Computing Systems
Institute for Computer Science and
Business Information Systems (ICB)
Faculty of Economics and
Business Administration
University of Duisburg-Essen

Schützenbahn 70
Room SH 502
45127 Essen
GERMANY
T +49 (0)201 183-3914
F +49 (0)201 183-4573

specht@dc.uni-due.de
http://dc.uni-due.de