MaxLatency Contribution

Astrit Ademaj
The issue of transforming the semantics of MaxLatency parameter from “Latency-semantics” to “Deadline-semantics” for time-aware streams was discussed in the previous calls.

- This contribution proposes adding of an additional parameter to address the “deadline requirements”

- In addition, there is an issue with the MaxLatency when this parameter is used with the „Latency-semantics” and specially in combination with preemption
  - This presentation describes the issues with the existing MaxLatency definition and proposes 3 different options (contributions) to address the given issue

- This contribution does not address the issues with the definition of the “reference points”.

Introduction
3.118 Latency – is currently defined as:

The delay experienced by a frame in the course of its propagation between two points in a network, measured from the time that a known reference point in the frame passes the first point to the time that the reference point in the frame passes the second point.

IEEE 802.1Qcc specifies a MaxLatency model and the CNC-CUC interface, and defines different reference points for:

- non-time-aware streams
  - 46.2.3.6.2 Latency shall use the definition of 3.118, with additional context as follows: The ‘known reference point in the frame’ is the message timestamp point specified in IEEE Std 802.1AS for various media (i.e. start of the frame). The ‘first point’ is in the Talker, at the reference plane marking the boundary between the network media and PHY (see IEEE Std 802.1AS). The ‘second point’ is in the Listener, at the reference plane marking the boundary between the network media and PHY.

- time-aware streams and
  - 46.2.3.6.2 When TSpecTimeAware is present: The ‘first point’ is assumed to occur at the start of the Interval, as if the Talker’s offsets (EarliestTransmitOffset and LatestTransmitOffset of 46.2.3.5) are both zero.

MaxLatency is a request parameter to the CNC for a particular stream (requirement to CNC for a particular stream set up by a listener*)
§3.118 Latency: The delay experienced by a frame in the course of its propagation between two points in a network, measured from the time that a known reference point in the frame passes the first point to the time that the reference point in the frame passes the second point.

46.2.3.6.2 When TSpecTimeAware is present: The 'first point' is assumed to occur at the start of the Interval, as if the Talker’s offsets (EarliestTransmitOffset and LatestTransmitOffset of 46.2.3.5) are both zero.

Issue: “MaxLatency” definition for time aware talkers (1)

MaxLatency definition becomes a "deadline" semantics for a time-aware talker

There are time intervals where the frame is not in propagation at all
Issue: “MaxLatency” definition for time aware talkers (2)

- MaxLatency definition is “transformed” to a “Deadline” definition for “time-aware” streams
- Latency requirements/constraints and “Deadline” requirements/constraints are different values
- With the existing definition there is possibility to express requirements with latency semantics for time-aware streams

Proposal 1: Introduce an explicit parameter for:
- “deadline” requirement for listeners and
- “deadline” response for listeners
Contribution: “Deadline” definition for time aware talkers

§46.2.3.6.xy \textbf{MaxDeadline} – is defined as:

The point in time within ‘Interval’ where the end of the last symbol of the FCS passes the reference plane (PHY).

\textbf{MaxDeadline} is a request parameter in the CNC-CUC interface

§46.2.3.6.xy \textbf{ListenerDeadline} – is defined as:

The point in time within ‘Interval’ where the end of the last symbol of the FCS passes the reference plane marking the boundary between the network media and PHY.

\textbf{ListenerDeadline} is the response parameter in the CNC-CUC interface. \textbf{ListenerDeadline} shall be equal or smaller than \textbf{MaxDeadline}

In case that \textbf{MaxDeadline} or \textbf{ListenerDeadline} value is larger than the \textbf{Interval} value, that means that the frame is being issued by the talker in the previous interval, and the frame is being expected in the following interval at the:

\textbf{Phase offset} = \textbf{ListenerDeadline} \% \textbf{Interval}
MaxLatency issue
End-To-End Application Latency

Application designer knows:
- Encoding Delay
- Network Stack Delay (Talker)
- Network Stack Delay (Listener)
- Decoding Delay

Application designer specifies:
- Application E2E Latency
Distributed control application

TSN Network is designed to use a CNC (*fully centralized model is assumed*)

Application designer knows (endsystem implementation specific parameters):

- **Encoding Delay**
- **Network Stack Delay (Talker Side)**
- **Network Stack Delay (Listener Side)**
- **Decoding Delay**

Application designer specifies **Application E2E Latency**

Application designer calculates **Network Latency** from the **Application E2E Latency**

Application designer does not need to know which (TSN) network features are used

**Application E2E Latency** is the time interval between:
- the timepoint when the talker application finishes the computation/generation of the stream data and
- the point in time when the stream data are available for the listener application layer

**Network Latency** is the time interval between
- the timepoint when the first bit „hits“ the reference point (PHY boundary) at the talker
- the timepoint when the last bit leaves the reference point (PHY boundary) at the listener
1. Application designer specifies stream parameters (e.g., talker, list of listeners MaxFrameSize, Interval,...)

2. Application designer specifies the **Application E2E Latency** for listeners

3. Application designer calculates the **Network Latency** by using **Application E2E Latency** as basis

   \[
   \text{Network Latency} = \text{Application E2E Latency} - \text{Encoding Delay} - \text{Network Stack Delay (Talker)} - \text{Network Stack Delay (Listener)} + \text{Decoding Delay}
   \]

4. Application designer provide among other parameters (e.g., Max Frame Size) the **Network Latency** for the „function“ that sends the request to the CUC/CNC

\[
\text{Network Latency} = t_3 - t_1 \quad \text{is the parameter of interest for the application designer}
\]

\[
\text{Max Latency} = t_2 - t_1 \quad \text{is the available parameter in 802.1Qcc (i.e., it has a different semantics than the Network Latency)}
\]
5. **NetworkLatency** needs to be translated in a parameter available in the 802.1Qcc. i.e., **MaxLatency**

   a) Translation from the **NetworkLatency** to the **MaxLatency** can be done if the listener link speed is known to the application

   b) *In case the preemption is active - this translation shall be done as if the frame is received in one piece.*

   \[
   \text{MaxLatency (Listener)} = \text{func}(\text{NetworkLatency, Listener\_link\_speed, MaxFrameSize, NO\_PREEMPTION, SINGLE\_FRAME\_PER\_INTERVAL})
   \]
Possible solutions (contributions)

Option 1: Keep the **MaxLatency** semantics as it is and add additional notes to clarify how the **MaxLatency** and the **AccumulatedLatency** shall be used (interpreted at the CNC/CUC and the Listener)

Option 2: Redefine the semantics of the **MaxLatency**

Option 3: Define additional Latency parameter for time-aware streams
In this case the application designer, the CNC and the Listener application shall do following calculations

1. Application designer defines \textit{NetworkLatency} for each Talker-Listener pair
   a) translates the \textit{NetworkLatency} to \textit{MaxLatency} for each Talker-Listener pair

2. CNC receives \textit{MaxLatency} as request parameter for stream configuration
   a) CNC calculates the \textit{NetworkLatency} based on the \textit{MaxLatency}, \textit{ListenerLinkSpeed}, and \textit{MaxFrameSize} (known at the CNC) as in case that no preemption is in place
   b) \textit{NetworkLatency} is the parameter of interest for the CNC
   c) In case of multiple frames per Interval, CNC includes in the \textit{NetworkLatency} the talker-jitter as well as IFG
3. CNC calculates the *accumulated network latency* (first bit transmitted - to last bit received)
   a) *accumulated network latency* shall also cover possible delays caused by frame preemption
   b) As *accumulated network latency* parameter is not available in Qcc model, the CNC calculates the *accumulated latency* based on *accumulated network latency* as in case of no preemption is in place (i.e., reduces the value for the time interval for the transmission of the frame of MaxDataSize and the LinkSpeed)
   c) CNC passes the *accumulated latency* to its listeners through the CUC

4. Listener receives the *accumulated latency* from the CUC/CNC
   a) Listener calculates the *accumulated network latency* (this is the parameter of interest for the listener) based on *accumulated latency* under the assumption that
      - no preemption was used and
      - the listener application knows its link speed
Example: 100 Mbit/s link speed, MaxFrameSize = 1518 bytes, 123\(\mu\)s is the time needed for this frame to pass over a single link

1. Application designer specifies NetworkLatency for a given listener = 2000\(\mu\)s
   a) translates the NetworkLatency to MaxLatency = 2000\(\mu\)s - 123\(\mu\) = 1877\(\mu\)s

2. CNC receives MaxLatency as request parameter 1877\(\mu\)s
   a) CNC derives the NetworkLatency based on the MaxLatency 1877\(\mu\)s + 123\(\mu\) = 2000\(\mu\)s

3. CNC generate its schedule and calculates the AccumulatedNetworkLatency which includes delays caused by preemption
   a) CNC can calculate that the frame is being preempted and that the first bit is received after 1000\(\mu\)s, but the last bit is received after 1800\(\mu\)s (meaning AccumulatedNetworkLatency = 1800\(\mu\)s)
   b) CNC derives the AccumulatedLatency as 1800\(\mu\)s - 123\(\mu\) = 1677\(\mu\)s
   c) CNC passes the AccumulatedLatency value of 1677\(\mu\)s to the listener through the CUC

4. Listener receives the AccumulatedLatency from the CUC 1677\(\mu\)s
   a) Listener calculates the AccumulatedNetworkLatency 1677\(\mu\)s + 123\(\mu\)s = 1800\(\mu\)s

Contribution Option 1: Keep the MaxLatency semantic - Example
Summary Contribution Option 1

1. Add the text for additional calculations for \textit{MaxLatency} and \textit{AccumulatedLatency} in case of preemption at the CNC and the listeners

2. Change the semantics of \textit{MaxLatency} for time-aware streams

3. Add the definition for \textit{MaxDeadline} and \textit{NetworkDeadline}
Change the definition of *MaxLatency* to cover the parameter *NetworkLatency* without the need to perform calculation at the CNC and at the Listener. Change the semantics from:

- the time interval from the first bit at the talker to the first bit at the listener
- the time interval from the first bit at the talker to the last bit at the listener

**Proposal:** change Sect. 46.2.3.6.2 to:

```
“MaxLatency shall use the definition of 3.118, with additional context as follows: The “known reference point in the frame” is the message timestamp point specified in IEEE Std 802.1AS for various media (i.e., start of the frame) for the “first point”,

and the end of the last symbol of the FCS for the “second point”.

The “first point” is in the Talker at the reference plane marking the boundary between the network media and PHY (see IEEE Std 802.1AS). The “second point” is in the Listener at the reference plane marking the boundary between the network media and PHY.”
```
Summary Contribution Option 2

1. Change the semantics of $MaxLatency$ and $AccumlatedLatency$
2. Change the semantics of $MaxLatency$ for time-aware streams
3. Add the definition for $MaxDeadline$ and $NetworkDeadline$
Contribution Option 3: **Define additional Parameters in Qdj**

- MaxLatency can be kept for non-time aware streams
- Define additional parameters with the semantics of first bit *the first bit at the talker and the last bit at the listener*

  - **NetworkMaxLatency** (Request parameter to the CNC)
  - **NetworkAccumulatedLatency** (Response parameter from the CNC)

**Proposal:** change Sect. 46.2.3.6.2.xy to:

*NetworkMaxLatency and NetworkAccumulatedLatency shall use the definition of 3.118, with additional context as follows: The “known reference point in the frame” is the message timestamp point specified in IEEE Std 802.1AS for various media (i.e., start of the frame) for the “first point”, and the end of the last symbol of the FCS for the “second point”.*

The “first point” is in the Talker at the reference plane marking the boundary between the network media and PHY (see IEEE Std 802.1AS). The “second point” is in the Listener at the reference plane marking the boundary between the network media and PHY.”
1. Add the definition for NetworkMaxLatency and NetworkAccumulatedLatency
2. Add the definition for MaxDeadline and NetworkDeadline
### Summary

<table>
<thead>
<tr>
<th>Option 1:</th>
<th>Option 2</th>
<th>Option 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Add text for extra calculations</td>
<td>• Change the semantics (MaxLatency and AccumulatedLatency)</td>
<td>• Define additional parameters</td>
</tr>
<tr>
<td>• Change the semantics for MaxLatency and time-aware streams</td>
<td>• add MaxDeadline and NetworkDeadline</td>
<td>• add MaxDeadline and NetworkDeadline</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Pros</th>
<th>Cons</th>
<th>Pros</th>
<th>Cons</th>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Calculations need to be performed by the listener application</td>
<td>User Friendly: No calculations need to be performed by the listener application</td>
<td>Clear interface</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Listener application need to know network details such as link speed</td>
<td>Listener application does not need to know network details such as link speed</td>
<td>---</td>
<td>Listener application does not need to know network details such as link speed</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Can be confusing in case of switched endstations with an internal link between the switch and the endstation part. Listener application needs to know the internal link speed</td>
<td>In case of switched endstations with an internal link, the internal link speed is known by the CNC, transparent to the listener application and application designer</td>
<td>---</td>
<td>In case of switched endstations with an internal link, the internal link speed is known by the CNC, transparent to the listener application and application designer</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>No backward compatibility issues for non-time-aware streams</td>
<td>Possible backward compatibility issues for time-aware streams</td>
<td>---</td>
<td>Possible backward compatibility issues for time-aware and non-time aware streams</td>
<td>No backward compatibility issues</td>
<td>---</td>
</tr>
</tbody>
</table>