TTTech industrial

MaxLatency Problem Description

Astrit Ademaj

Introduction

- This slide set describes the issues with the existing MaxLatency Qcc definition
- The intention is to clarify the details of these issues to the TSN working group

IEEE 802.1Q and 802.1Qcc Definitions

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.**

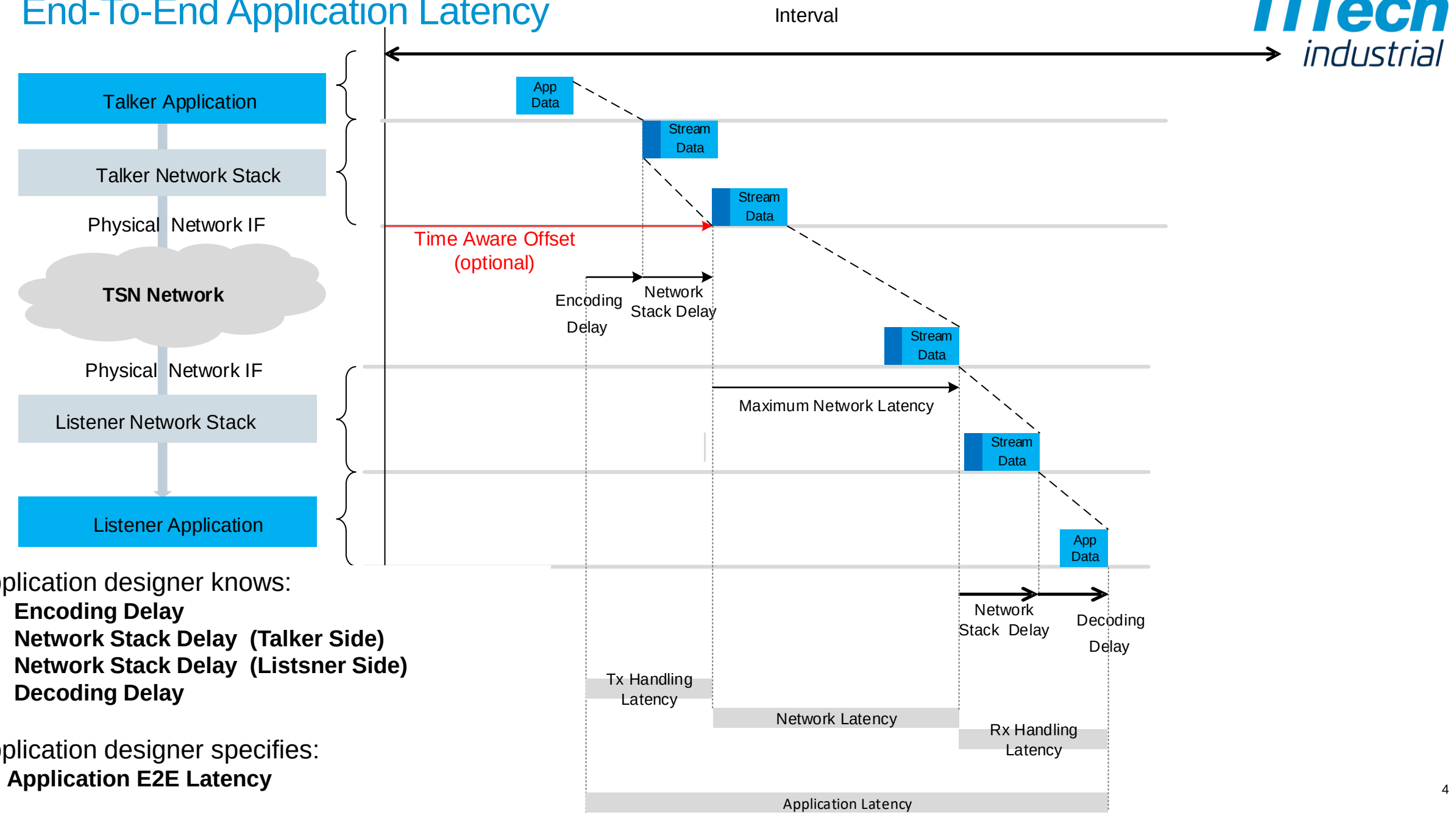
IEEE 802.1Qcc specifies a MaxLatency model and the CNC-CUC interface, and defines different reference points for:

- 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*)

* For the sake of simplicity currently we are assuming only the fully centralized model. There are other use cases (like the distributed user and centralized model) that need this parameter as well.

End-To-End Application Latency



Application designer knows:

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

Application designer specifies:

- **Application E2E Latency**

Application Model

- Distributed control application
- TSN Network is designed to use a CNC (*fully centralized model* is assumed*)
- Application designer knows (endsystem impementation 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 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 where the listener application is

Network Latency is the time interval between

- the timepoint when the first bit „hits“ the physical layer interface until at the talker
- the timepoint when the last bit leaves the physical layer interface at the listener

* For the sake of simplicity currently we are assuming only the fully centralized model.

Application Model

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

```
TxHandlingLatency = EncodingDelay + NetworkStackDelay(Talker)
RxHandlingLatency = NetworkStackDelay(Listener) + DecodingDelay
```

```
NetworkLatency =
        ApplicationE2ELatency – TxHandlingLatency – RxHandlingLatency
```

- Application designer provides the Network Latency as requirement to the network management tool/function.

`NetworkLatency` depends on the features used in the (TSN) network as well as the links speeds in the path of the stream

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

Configuration Workflow:

1. Application designer specifies the **Application E2E Latency**
2. Application designer calculates the **Network_Latency** by using **Application E2E Latency as basis**
3. Application designer provide among other parameters (e.g., Max Frame Size) the **Network Latency** to the „function“ or „stack“ that sends the request to the CUC/CNC
   i. **Network Latency** needs to be translated in a parameter available in the 802.1Qcc (802.1Qdj)
      a) MaxLatency is the forseen parameter in 802.1Qcc – but this has a different semantics as the **Network Latency**.
      b) Translation from the **Network Latency** to the **MaxLatency** can be done if the listener link-speed is known.
      c) In case the preeption is active - this translation shall be done as if the frame is received in one piece.

Application does not need to known which Network (TSN) features are used

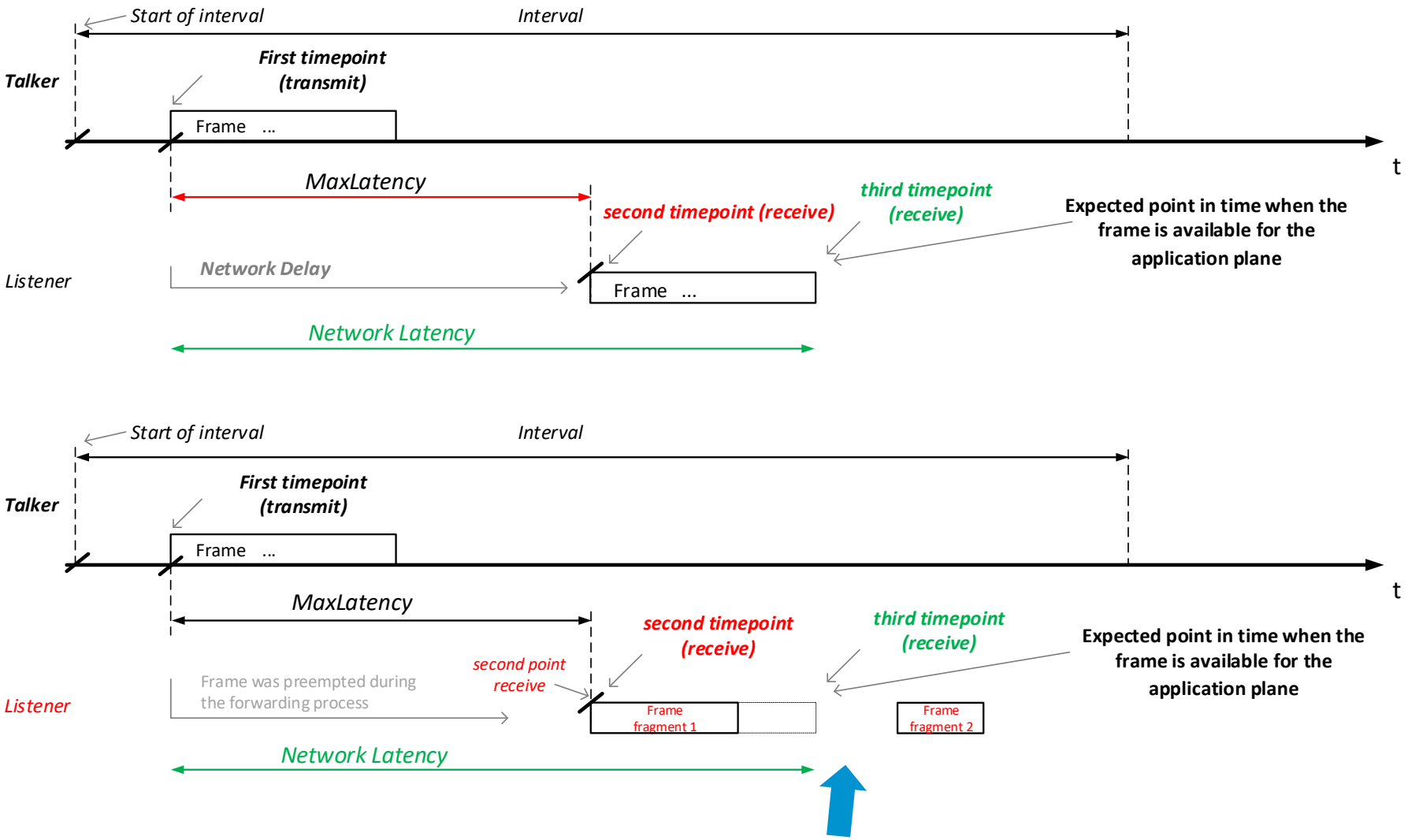
Application E2E Latency is the time interval between:

- the timepoint when the talker application finishes the computation/generation of the stream data and
- the point in time where the listener application is

Network Latency is the time interval between

- the timepoint when the first bit „hits“ the physical layer interface until at the talker
- the timepoint when the last bit leaves the physical layer interface at the listener

Issue : “MaxLatency” definition



In case of preempted packets (and the AccumulatedLatency=MaxLatency), with the actual “MaxLatency” definition, it is not possible for the listener to derive the worst-case point in time when the stream‘s frame is fully received, i.e. the point in time when the internal processing starts. The CNC and the listener needs to do additional translations

Problem #1

Network Latency (application side)

is a parameter provided by the application designer that specifies the time interval between the timepoint where

- the **first bit** at the talker and
- the **last bit** at the listener,

hits the reference plane

MaxLatency (network side)

In 802.1Qcc specifies the time interval between the timepoint where

- the **first bit** at the talker and
- the **first bit** at the listener,

hits the reference plane

Problem #1

Network Latency needs to be translated into MaxLatency as in case that preemption is not used used and the frame is received in one piece.

AccumulatedLatency needs to be translated at the CNC (when providing configuration output) and at the Listener (when using the AccumulatedLatency as input) as in case that preemption is not used and the frame is received in one piece.

“MaxLatency” definition for time aware talkers (1)

§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.

§46.2.3.6.2 MaxLatency – is defined as:

> 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)...

46.2.3.6.2 MaxLatency – redefinition in case of Tspec time-aware is present

> 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.

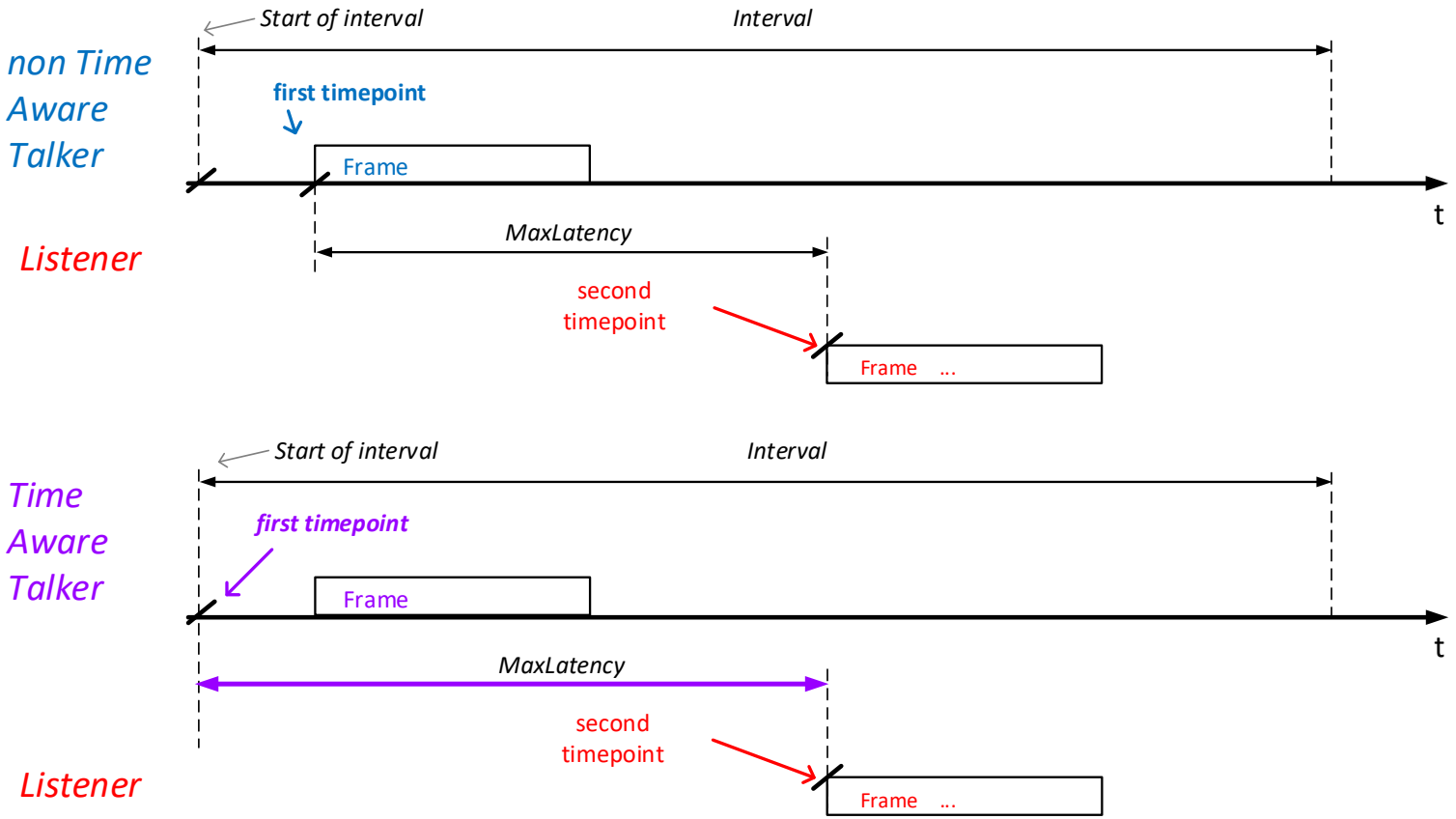
What does it mean?

Issue 2: “MaxLatency” definition for time aware talkers (2)

§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.

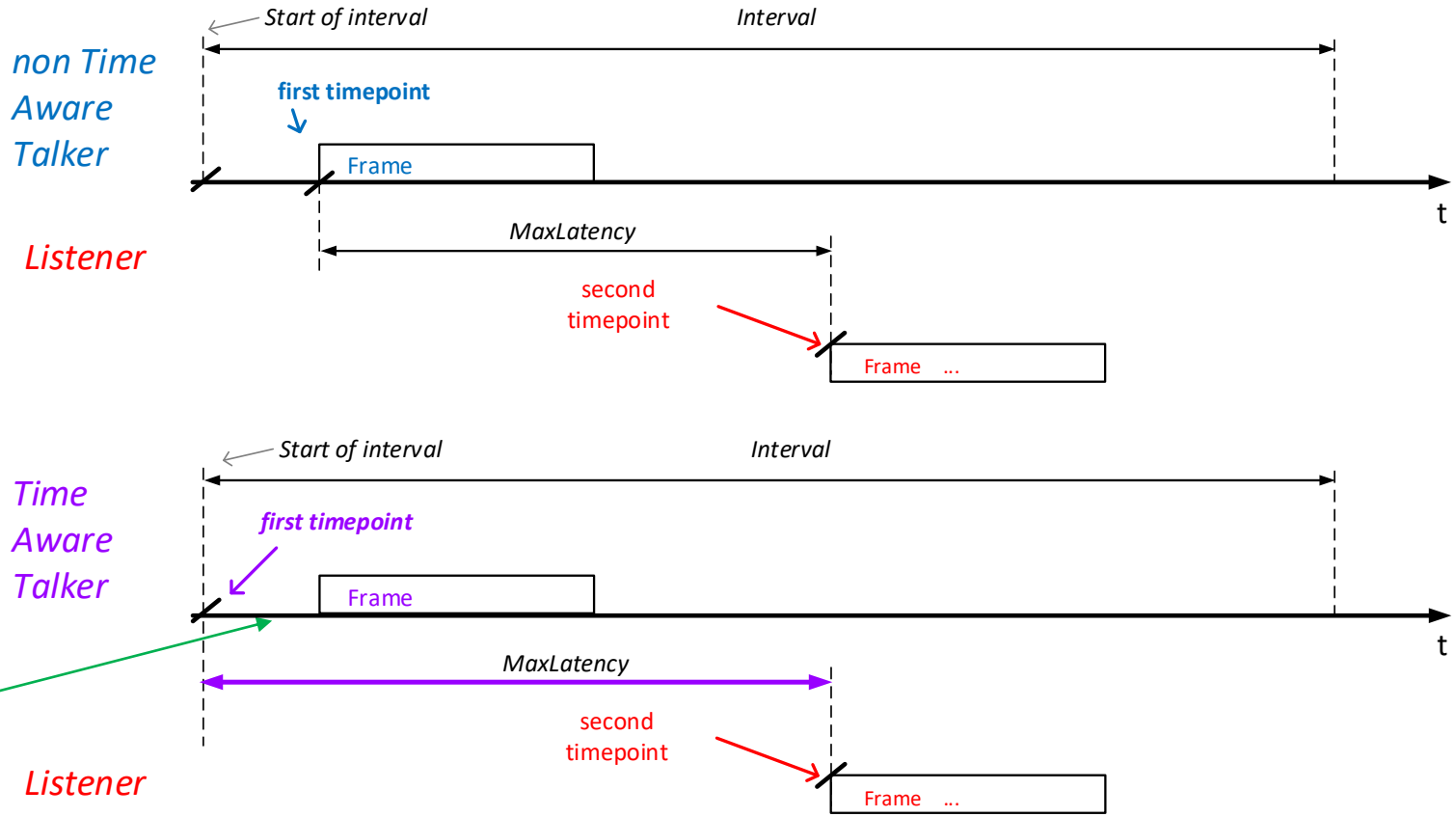
MaxLatency definition becomes a “deadline” semantics for a time aware talker



Issue 2: “MaxLatency” definition for time aware talkers (3)

§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.



There are time intervals where the frame is not in propagation at all

Problem #2

Problem #2.a

- *Semantics of MaxLatency is changed from („latency“ to „deadline“) if TSpecTimeAware is present*
- *This semantic redefinition conflicts with the initial definition of „Latency“ in §3.118 (see Slide 12)*

Problem #2.b

- *With the existing redifinition of the semantics from „latency“ to „deadline“ for time-aware streams, it is not possible to express „Latency“ requirements for time-aware streams.*
- *Currentyl only „deadline“ requirements can be expressed through the MaxLatency. Missing parameter to express a „latency“ requirement for time aware streams (such use cases exists).*

Problem #3:

- *The semantics of terms "first point" and "second point" in* 46.2.3.6.2 *is ambiguous.*
- *The terms are used to define (physical) locations in a switched network, but they are likewise used to define points in time.*