Proposed enhancement of the definition of MaxLatency in 46.2.3.6.x of IEEE Std 802.1Qcc-2018

Date:	2021-05-05
Contributor(s)/Affiliation for this contribution:	Johannes Specht/Self

Individual contribution by Johannes Specht

Introduction

This contribution addresses two issues described in https://www.ieee802.org/1/files/public/docs2021/dj-ademaj-MaxLatency-problem-description-0412-v02.pdf.

a) The first issue is that preemption can defer frame completion in receivers beyond the frame length, and it should be transparent for users on whether the network (e.g., CNC) configures preemptible streams or not. This knowledge should solely be in the response of the network, not the user. The issue can be generalized
There are other transformations of frames in the network possible, including (but not limited to) different link speeds and probably the impact of tagging and tag removal. Some of these are a result of the network configuration, and not known at the time users provide their requirements (e.g., MaxLatency) to the network. This issue is addressed in 46.2.3.6.2 of this contribution. Moreover, some other minor issues are fixed.
b) The second issue is that MaxLatency, as defined in the beginning of 46.2.3.6.2, is redefined at the end of 46.2.3.6.2 as soon as TSpecTimeAware is present. This issue is addressed by addition of 46.2.3.6.3. The original intent of this re-definition is unclear to the author of this contribution after reading it carefully. It appears that several underlying assumptions are not documented in IEEE 802.1Qcc-2018, and the . The corresponding text in 46.2.3.6.2 is widely left unchanged in this contribution, and it is suggested to (i) discuss the purpose of this text on a technical level, and (ii) decide whether a semantic redefinition of MaxLatency in presence of TSpecTimeAware needs to be retained. The material in 46.2.3.6.3 is suggested only for the case that a redefinition of MaxLatency in presence of TSpecTimeAware is retained. If such a redefinition is removed, the proposal in 46.2.3.6.3 is not required.

All proposed changes are intended to retain backwards compatibility. As a result, no additions were made to account for effectively having S&F in the last hop. If this should be addressed, this should probably be done in the new parameter in 46.2.3.6.3.

Individual contribution by Johannes Specht

Change Table 46-10, as shown:

Name	Data type	Reference
NumSeamlessTrees	uint8	46.2.3.6.1
MaxLatency	uint32	46.2.3.6.2
MaxLatencyTimeUnaware	uint32	46.2.3.6.3

Change 46.2.3.6.2, as indicated:

46.2.3.6.2 MaxLatency

MaxLatency specifies the maximum latency from Talker to Listener(s) for any ~~a single frame of~~ data contained in frames ~~the~~ of the Stream.

MaxLatency is specified as an integer number of nanoseconds.

Latency shall use the definition of 3.1~~8~~20, with additional context as follows:

a) MaxLatency expresses a requirement for every ~~The~~ "~~known~~ reference point ~~in the frame~~" that is known by Talker and Listener, and for every frame of the Stream. The requirement applies for each particular reference point (e.g., a specific bit in a particular frame of a stream) individually, but MaxLetency is the same requirement that applies for all such individual reference points.

b) ~~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~~ at the Talker's end station at ~~the reference plane marking~~ the boundary ~~between~~ from ~~the network media and~~ PHY this end station to the physical medium ~~(see IEEE Std 802.1AS)~~.

c) The "second point" is ~~in~~ at the Listener's end station at the ~~reference plane marking the~~ boundary from ~~between~~ the ~~network media~~physical medium ~~and~~ to ~~PHY~~this end station.

NOTE— The definition in item a) is backwards compatible with the definition found in IEEE Std 802.1Qcc-2018 under the assumption that the link speeds at Talker and Listener end stations are identical, and frames of the Stream are received at a Listener's end station without any differences compared to the frames sent by the Talker end station. The definitions in item b) and item c) are identical to the definition of the "reference plane" in IEEE Std 802.1AS.

When this requirement is specified by the Talker, it must be satisfied for all Listeners.

When this requirement is specified by ~~the~~ a Listener, it must be satisfied for this Listener only.

If the UserToNetworkRequirements group is not provided within the Talker or Listener group, the network shall use the default value zero for this element.

The special value of zero represents usage of the initial value of Status. AccumulatedLatency as the maximum latency requirement. This effectively locks down the initial latency that the network calculates after successful configuration of the Stream, such that any subsequent increase in latency beyond that value causes the Stream to fail.

The assumption for when the "first point" occurs in the Talker depends on the presence of the TSpecTimeAware group in the Talker's TrafficSpecification.

~~)~~d) When TSpecTimeAware is not present

1) The Talker is assumed to transmit at an arbitrary time (not scheduled).

~~)~~e) When TSpecTimeAware is present

1) The "first point" ~~"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. The Talker's offsets are not typically zero, but use of the start of Interval for purposes of MaxLatency allows the Listener(s) to schedule their application independently from the Talker's offset configuration.
2) The Listener determines MaxLatency based on its scheduling of a read function in the application. Nevertheless, the time from frame reception (i.e., "second point") to execution of the read function is in the user's scope and therefore not included in MaxLatency.

Individual contribution by Johannes Specht

Kommentiert [j1]: Correction in the most recent Q-Rev required (I guess it will be published before Qdj).

Kommentiert [j2]: It is not clear to me why the original text limits to the start of the DA. I believe MaxLatency applies to any data in frames of the stream, not only the start of the DA.

Kommentiert [j3]: The abbreviation should be replaced, not only here (16 places in draft D0.2 of the upcoming 802.1Q revision). Further explanation is given in item 314 of the IEEE 802.1 Maintenance database: https://www.802-1.org/items/427/requests/317/pre

Kommentiert [j4]: This seems like a problematic redefinition of term "first point": In the earlier text of this clause and in 3.120, "first point" refers to a (physical) **location in a network**. In contrast, "first point" refers here (and subsequently) to particular **points in time**.

3) MaxLatency can be set to a value greater than the Talker's Interval in order to specify a longer latency requirement. For example, if the Talker's Interval is 500 µs and MaxLatency is 700 µs, the Listener receives the frame no later than 200 µs into the interval that follows the Talker's Interval.

Insert 46.2.3.6.3 after 46.2.3.6.2:

46.2.3.6.3 MaxLatencyTimeUnaware

MaxLatencyTimeUnaware is only present if it is supported by the user, and specifies the maximum latency from Talker to Listener(s) for a single frame of the Stream independent of TSpecTimeAware. The semantics of MaxLatencyTimeUnaware is defined as follows:

a) If TSpecTimeAware is not present, MaxLatencyTimeUnaware shall be identical to MaxLatency (46.2.3.6.2).
b) If TSpecTimeAware is present, MaxLatencyTimeUnaware shall be identical to MaxLatency in absence of TSpecTimeAware.

If MaxLatencyTimeUnaware is unsupported by the network, a network configuration can be determined based on MaxLatency only.

NOTE—MaxLatencyTimeUnaware has been introduced after the initial definition of MaxLatency to retain the semantics of MaxLatency unchanged for backwards compatibility. It is always possible to configure a network based on MaxLatency only, regardless of whether MaxLatencyTimeUnaware is provided by one or more users. However, MaxLatencyTimeUnaware enables additional configuration possibilities.

Kommentiert [j5]: I can just guess that one possible motivation of these items could be to to allow Listeners to schedule applications more independent from the configuration (offsets in an interval) as determined by the network.
However, I don't understand why the definitions in these bullet items help in this regard?

Kommentiert [j6]: Remarks:
- I didn't check whether optional parameters, in this case MaxLatencyTimeUnaware, are possible. If not, then this parameter might be better placed in a dedicated optional group.
- Some users may support MaxLatencyTimeUnaware while others do not support MaxLatencyTimeUnaware. In this case, A CNC can just fall back to solely operate based on MaxLatency.