Comments for P802.1Qdq/D0.0

Jie Xiao, Min Zha, Li Xu
{xiaojie23, zhamin, innon.xu}@huawei.com
Huawei Technology Co., Ltd
Comments #1

• Current text

4 X.2 Bursty Traffic Requiring Bounded Latency

This clause defines the traffic type handled by a bridged network and its parameters that describe this type of traffic.

Figure X-3 illustrates the bursty traffic pattern. Each data block has a bounded latency. The bounded latency is assumed to be pre-determined by an application or set manually by an operator of an application. It defines the maximum time from the reference point at the application in the Talker to the reference point at the Listener. In view of the characteristics of some data transmission with a large interval between clusters that can exceed several tens of milliseconds or event-driven data generation by IoT devices [B1], the traffic treated here is sporadic, with condition that the next frame cluster never arrives until the entire corresponding queue in a bridge becomes empty.

Figure X-3—Burst traffic of fragmented block data

14 Figure X-4 illustrates the detailed traffic pattern and queues of the traffic type to be defined here in the application’s point of view. The traffic is described by the three given parameters: Data Size, Bounded Latency and Minimum Cluster Interval. At the time $t_0$, a transmitting application sends a block data $Data(i)$ whose size is equal to or less than "Data Size" and may be greater than frames the bridged network can handle. The whole block data requires to reach the corresponding receiving application through the bridged network by the time $t_1$ that is equal to or less than $t_0$ plus "Bounded Latency." In addition, the transmitting application puts the subsequent block data at time $t_{1+i}$, which should be equal to or greater than $t_0$ plus 27 "Minimum Cluster Interval."

Weakness: there are multiple words or phrases to express a burst, which are confusing.

• Suggested text

4 X.2 Bursty Traffic Requiring Bounded Latency

This clause defines the traffic type handled by a bridged network and its parameters that describe this type of traffic.

Figure X-3 illustrates the bursty traffic pattern. Each data block has a bounded latency. The bounded latency is assumed to be pre-determined by an application or set manually by an operator of an application. It defines the maximum time from the reference point at the application in the Talker to the reference point at the Listener. In view of the characteristics of some data transmission with a large interval between clusters that can exceed several tens of milliseconds or event-driven data generation by IoT devices [B1], the traffic treated here is sporadic, with condition that the next frame cluster never arrives until the entire corresponding queue in a bridge becomes empty.

Figure X-3—Burst traffic of fragmented block data

14 Figure X-4 illustrates the detailed traffic pattern and queues of the traffic type to be defined here in the application’s point of view. The traffic is described by the three given parameters: Data Size, Bounded Latency and Minimum Cluster Interval. At the time $t_0$, a transmitting application sends a block data $Data(i)$ whose size is equal to or less than "Data Size" and may be greater than frames the bridged network can handle. The whole block data requires to reach the corresponding receiving application through the bridged network by the time $t_1$ that is equal to or less than $t_0$ plus "Bounded Latency." In addition, the transmitting application puts the subsequent block data at time $t_{1+i}$, which should be equal to or greater than $t_0$ plus 27 "Minimum Cluster Interval."

‘data block’ can be replaced by ‘frame cluster’
Data size should be reduced linearly rather than vertically.

Like the above curve, similar problem also exists.

Figure X-4—Traffic pattern in an application’s point of view
DeliveryTime defines the time from the first bit sent from a talker to the last bit received at a listener.

Accordingly, the formulas (Y-2) & (Y-3) should be modified


deliveryTime(i) = \frac{\sum_{k=1}^{i-1} \text{frameLength}(n)}{\text{shapingRate}} + \text{accumulatedLatency} \quad (Y - 2)

deliveryTime(i) = \sum_{k=1}^{i} \frac{\text{frameLength}(k)}{\text{shapingRate}} + \text{accumulatedLatency} \quad (Y - 2)

\text{requiredMinimumShapingRate} = \frac{\sum_{k=1}^{i} \text{frameLength}(k)}{\text{targetLatency}} \quad (Y - 3)

\text{requiredMinimumShapingRate} = \frac{\text{dataSize}}{\text{targetLatency}} \quad (Y - 3)
Weakness:
① The parameter ‘portTcMaxLatency’ is not used in the whole article, so it can be deleted.
② Consistent issues: the parameter ‘AccumulatedLatency’ should be accumulatedLatency, the same as in (X-5). Other parameters, such as Data Size (dataSize), Bounded Latency (boundedLatency) should also follow a consistent style.
③ Add description. (see right)

X.3 Accumulated Latency and Bridged Network

X.3.1 Accumulate Latency

Per-hop latency imposed by a bridged network against a single frame is discussed in Clause 35 and Annex L. Latency between Talker and Listener (hereinafter referred to as Network Latency) is derived from the sum of per-hop latency along the path between them. In bridged networks controlled by SRP, the value portTcMaxLatency and AccumulatedLatency can be obtained from the system, while it can be manually evaluated by the same way in networks without SRP.

The value AccumulatedLatency is used as one of inputs of the calculating procedure for shaper parameter settings defined in Y.4.

As the calculation of accumulatedLatency (in Equation (V-6) in Annex V of IEEE Std 802.1Qcr, 2020) is highly related to other streams in the system, when the system load changes (especially new higher-class streams are added), accumulatedLatency should be recalculated. Correspondingly, the SRP may also need to be re-executed.

Equation (V-6) in Annex V of IEEE Std 802.1Qcr:

\[
d_{a,v}(f) = \sum_{k=1}^{n} d_{G_{1,v},w_{a}}(k,f) + \sum_{k=2}^{n} d_{D_{1,v},w_{a}}(k) + \sum_{k=2}^{n} d_{D_{1,v},w_{a}}(k) + \sum_{k=2}^{n} d_{E_{1,v},w_{a}}(k),
\]  

(V-6)

\[
d_{G_{1,v},w_{a}}(k,f) = \max_{k \in F_{d}(k,f)} \left[ \sum_{g \in F_{d}(k,g) \cap F_{d}(k,h)} h_{max}(k,g) - l_{max}(h) + l_{I_{F},w_{a}}(k,h) + \frac{l_{min}(h)}{R(k)} \right],
\]  

(V-2)
Comments #5

• Current text

Figure X-6—Frame propagation within bounded latency while minimizing over-provision of bandwidth reservation

The figure title is not complete.

• Suggested text

Figure X-6—Frame propagation within bounded latency while minimizing over-provision of bandwidth reservation
All formulas’ indices in the new draft are lost, but they are still cited in the text.

\[
\text{requiredMinimumShapingRate} = \sum_{i=1}^{n-1} \frac{\text{frameLength}(k)}{\text{targetLatency}}
\]

\[
= \frac{\text{dataSize} - \text{frameLength}(n)}{\text{targetLatency}}
\]

(X-3)
#8

**Comments #7**

**Current text**

- **X.4.3 Asynchronous Traffic Shaping**

According to the definition of the ATS scheduler state machine in Clause 8.6.11 (IEEE Std 802.1Qc-2020), the Committed Burst Size should be equal to or greater than frames sent by the Talker. In this case, it is recommended to be equal to the Maximum SDU Size. Committed Information Rate is the data rate reserved for the stream and is recommended to be equal to the required minimum shaping rate shown in Equation (Y-3). The approximation discussed in clause X.4.1 can also be applied. These lead to the following settings values:

\[
\begin{align*}
\text{Committed Burst Size} &= \text{Maximum SDU Size} \\
\text{Committed Information Rate} &= \frac{\text{data size}}{\text{target latency}}
\end{align*}
\]

Since the ATS scheduler state machine operation (8.6.11) assumes that the frame sizes that are processed are less than or equal to the associated Committed Burst Size parameter (8.6.11.3.5), the Committed Burst Size is set to be the maximum size of frame. That is equal to the Maximum SDU Size as shown in Equation (Y-5).

**Suggested text**

- **X.4.3 Asynchronous Traffic Shaping**

According to the definition of the ATS scheduler state machine in Clause 8.6.11 (IEEE Std 802.1Qc-2020), the Committed Burst Size should be equal to or greater than frames sent by the Talker. In this case, it is recommended to be equal to the Maximum SDU Size. Committed Information Rate is the data rate reserved for the stream and is recommended to be equal to the required minimum shaping rate shown in Equation (Y-3). The approximation discussed in clause X.4.1 can also be applied. These lead to the following settings values:

\[
\begin{align*}
\text{Committed Burst Size} &= \text{Maximum SDU Size} \\
\text{Committed Information Rate} &= \frac{\text{data size}}{\text{target latency}}
\end{align*}
\]

Since the ATS scheduler state machine operation (8.6.11) assumes that the frame sizes that are processed are less than or equal to the associated Committed Burst Size parameter (8.6.11.3.5), the Committed Burst Size is set to be the maximum size of frame. That is equal to the Maximum SDU Size as shown in Equation (Y-5).

The ATS scheduler should also work for the case in which the Committed Burst Size is greater than Maximum SDU Size. It does not affect other traffic for which the long-term averaged shaping rate is less than or equal to the associated Committed Burst Size parameter (8.6.11.3.5), the Committed Burst Size is set to be the maximum size of frame. That is equal to the Maximum SDU Size as shown in Equation (Y-5).

However, a small value of the Committed Burst Size is desirable because the transient data rate, which is higher than the required minimum shaping rate, may be suppressed. This transient manner can be caused by the arrival of a new frame cluster at the shaper that has already accumulated large number of tokens causing some frames to be forwarded instantly. Such token-bucket state can occur when no frames arrive at the shaper for a period of time between clusters.

Similar issue also exists in the CBS scheduler. For example, when there are N classes of frame clusters arrive concurrently in the system, during the scheduling of the higher-class clusters, credits of the lowest-class cluster will be accumulated, which may result in consistent transmission of its frames. To alleviate this issue, we should also limit the credit upper-bound to the maximum SDU size.

1) As shown in the example, there is a possibility that the number of credit/token can be larger than one frame size.
2) Once the above happen, the assumption (frames are equally spaced) in the draft may not hold.
3) So we must limit the credit/token upper bounded by one frame size.
Conclusion and Proposal

Conclusion:
• The project will play a great role for the parameter setting of shapers and make TSN easy to work.
• The current draft is good and the comments are trying to make it better/perfect.

Proposal:
• It is proposed that these comments are taken into consideration or as input for the further drafting of the current draft.