Annex Z

(Informative)

TrafficSpecification settings for bursty traffic with bounded latency

This annex clarifies the definition and use of TrafficSpecification (TSpec) for bursty traffic generated by time-sensitive (i.e. bounded latency) application in this standard and gives recommendations for TSpec settings for traffic shaping. The target traffic in this annex comprises of clusters of frames with a non-continuous and intermittent nature, assumed to be generated from Internet of Things (IoT) devices.

The objective is to mitigate temporarily high network load due to burtsy traffic when it shares the same port with other traffic by applying shaping. As an example of real-time camera inspection system to detect defects for the manufacturing site [Z1], a cluster of data frame is generated when products are set and is delivered to the system within 500 msec. The clusters of frames are shaped to meet certain delivery time tolerance defined by application requirements. In other words, the delivery time tolerance is the time allowed for which each cluster is received at the Listener after it is sent from the Talker, while avoiding over-provisioning of reserved bandwidth. TSpecs for MSRP and Token bucket are considered in this annex.

<< Editor’s Note: MSRP TSpec and Token bucket TSpec appear in P802.1Qdd/D0.2, subject to change>>

<<Editor Note: Add the following reference
[Z1] IEEE 802 Nendica Report: Flexible Factory IoT: Use Cases and Communication Requirements for Wired and Wireless Bridged Networks, April, 2020>>

Z.1 Feature of TSN network with Bursty Traffic

Z.1.1 Targeted Traffic Characteristic

Figure Z-1 shows an example of a bursty traffic pattern. In this example, unlike audio/video streaming, a group of frames, namely “Cluster of frames” are transmitted intermittently in a non-continuous manner. The cluster of frames occur sporadically, i.e., not periodically, implying T1≠T2 in Figure Z-1.

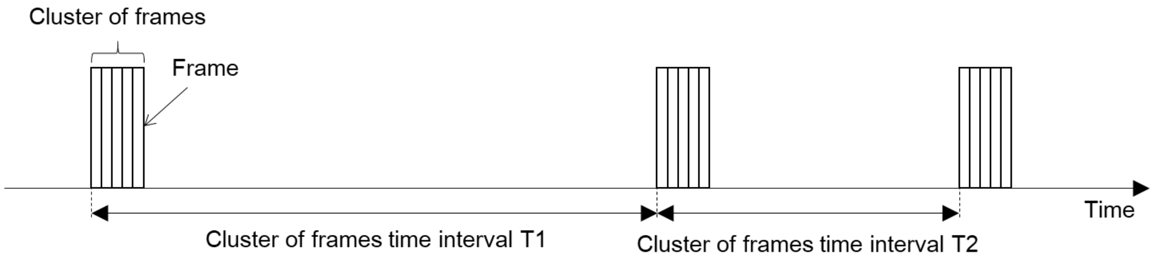

Figure Z-1— Example of bursty traffic pattern

Each cluster of frames has delivery time tolerance. The delivery time tolerance 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 which is more than several tens milliseconds or event-driven data generation by IoT devices [Z1], the traffic treated here is sporadic, meaning that the next cluster of frames never arrives until the entire corresponding queue in a bridge becomes empty.

Z.1.2 Network Structure

Figure Z-2 shows an example of the network configuration under consideration. This network comprises Talkers, Listeners, and bridges, which connect directly or indirectly to each other. Each stream of traffic is generated by a Talker, and is sent to the Listener via bridges across its route. .

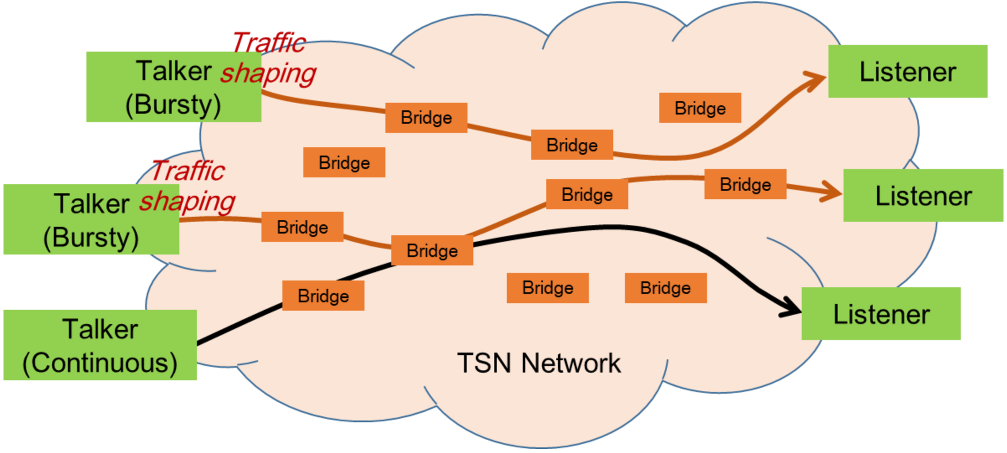

Figure Z-2 — An example of network structure under consideration

There are multiple streams flowing through this network, and they may flow into a common bridge. Traffic shaping is performed in the Talker and resource reservation is performed in bridges based on TSpec provided by the Talker. The specific traffic shaping method is based either on the credit-based shaper transmission selection algorithm (8.6.8.2) or on the ATS transmission selection algorithm (8.6.8.5).

<< Editor's Note: referring the ATS transmission selection algorithm (8.6.8.5) in the P802.1Qcr/D2.3 >>

The Talker is assumed to have enough buffer memory for data size from each cluster of frames to perform the traffic shaping. This assumption is based on sporadic nature of bursty traffic generated by the kind of IoT devices mentioned above.

Z.2 Overall Frame Transmission Delay

Z.2.1 Delivery Time

The flow of frames from the Talker to the Listener is shown in Figure Z-3.

The data size of each cluster comprising n-frames is equivalent to the sum of frame lengths.

$$dataSize = \sum_{k=1}^{n} frameLength(k) \qquad (Z-1)$$

Bursty traffic is shaped by the Talker. As a result of traffic shaping, the interval in which the Talker sends each frame becomes equal to the frame length divided by the shaping rate. Then, at the input of a Listener, the delivery time of this cluster of frames (as shown in Figure Z-3) becomes as follows:

$$deliveryTime = accumlatedLatency + \frac{\sum_{k=1}^{n-1} frameLength(k)}{shapingRate} \qquad (Z-2)$$

The shaping rate, within the traffic shaper, is set so that the delivery time is within the delivery time tolerance required by the application. The accumulatedLatency is the sum of delays of a stream in all the bridges across the route from the Talker to the Listener as given in Equation (V-6) in Annex V (of IEEE P802.1Qcr/D2.2). The accumulatedLatency is regarded as the propagation delay from the Talker to the Listener.

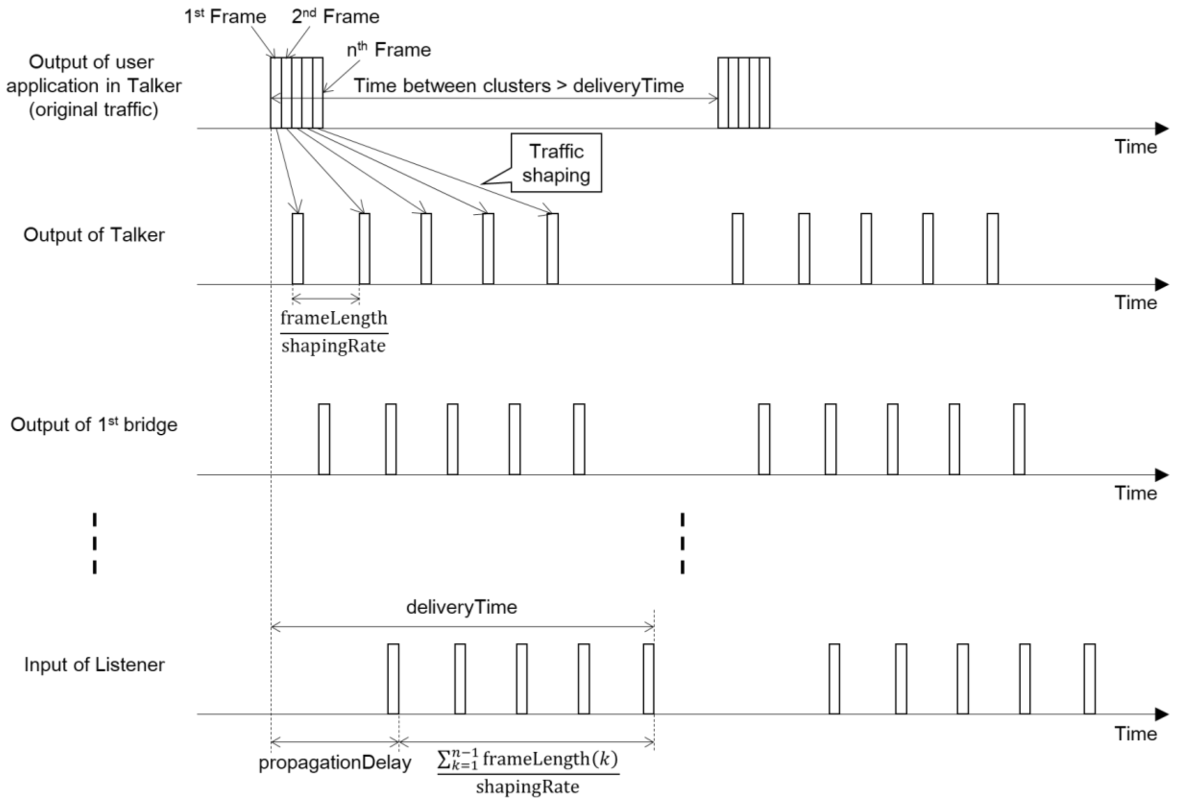

Figure Z-3— Frame propagation from Talker to Listener

Z.2.2 Accumulated Latency acquisition

In the previous clause, deliveryTime for the given shaping rate are derived by using dataSize and accumulatedLatency. The accumulatedLatency is calculated along the path of the stream. In addition, the path is selected by the traffic requirement including its data rate.

Z.2.2.1 Fully distributed model

When the fully distributed model (IEEE Std 802.1Qcc-2018, Clause 46.1.3.1) without a CNC (Centralized Network Configuration) is applied to the TSN network, no complete information is available in advance for computing the accumulated latency at all nodes along possible paths between the Talker and Listener. Therefore, an iterative method, which may result in local/approximate solution, is used to address this problem.

In the fully distributed model, UNI (User Network Interface) is used to exchange information related to propagation delay between bridges. The Talker may use MaxLatency element of UserToNetworkRequirements group (IEEE Std 802.1Qcc-2018, Clause 46.2.3.6.2) and AccumulatedLatency group (IEEE Std 802.1Qcc-2018, Clause 46.2.5.2) in order to obtain accumulatedLatency. The UNI specification requires the Talker to request joining a target stream. That is, the Talker cannot obtain the information before requesting to join a stream. Therefore, the Talker has to request to join a stream first with a tentative amount of the accumulatedLatency and then request to join again with the amount obtained by the first request. The first reservation and the second one are not guaranteed to return the same values of the accumulatedLatency and the Talker will try to join with different TSpec and MaxLatency based on the previously obtained accumulatedLatency repeatedly until successful joining the target stream. This method can be applied to Stream Reservation Protocol. (IEEE Std 802.1Qcc-2018, Clause 35)

Z.2.2.2 Fully centralized model and centralized network/distributed user model

A CNC does not have the same problem as the fully distributed network described in clause Z.2.2.1. This is because when a CNC in the centralized network/distributed user model (IEEE Std 802.1Qcc-2018, Clause 46.1.3.2) or the fully centralized model (IEEE Std 802.1Qcc-2018, Clause 46.1.3.3) is used to configure the TSN network, the CNC obtains all information from the network directly. For example, the CNC reads the bridge delay (12.32.1) and propagation delay (12.32.2) from each bridge in order to compute accumulatedLatency (-see Annex U, Clause U2, step 5, IEEE Std 802.1Qcc-2018).

Z.3 Recommended TSpec Settings

In order to minimize over-provisioning of bandwidth reservation while ensuring the requirement for the delivery time is met, the bursty traffic should be shaped with the minimum shaping rate within the required delivery time tolerance (required minimum shaping rate). Frame propagation within delivery time tolerance while minimizing over-provision of bandwidth reservation is illustrated in Figure Z-4 and referred to as the target latency. From Figure Z-4, the target latency can be derived from delivery time tolerance and accumulatedLatency. The required minimum shaping rate for traffic shaping is equal to:

$$requiredMinimumShapingRate = \frac{\sum_{k=1}^{n-1} frameLength(k)}{targetLatency}$$

$$= \frac{dataSize - frameLength(n)}{targetLatency} \qquad (Z-3)$$

In practice, the required minimum shaping rate can be approximated to (dataSize/targetLatency), which is slightly larger than the exact value if the frame length is smaller than data size. Actually, regardless small or large value of n-th frame length compared with data size, it gives an additional delivery time margin to the delivery time tolerance.

If the Talker does not have enough memory buffer compared with the data size, it does not function any more.

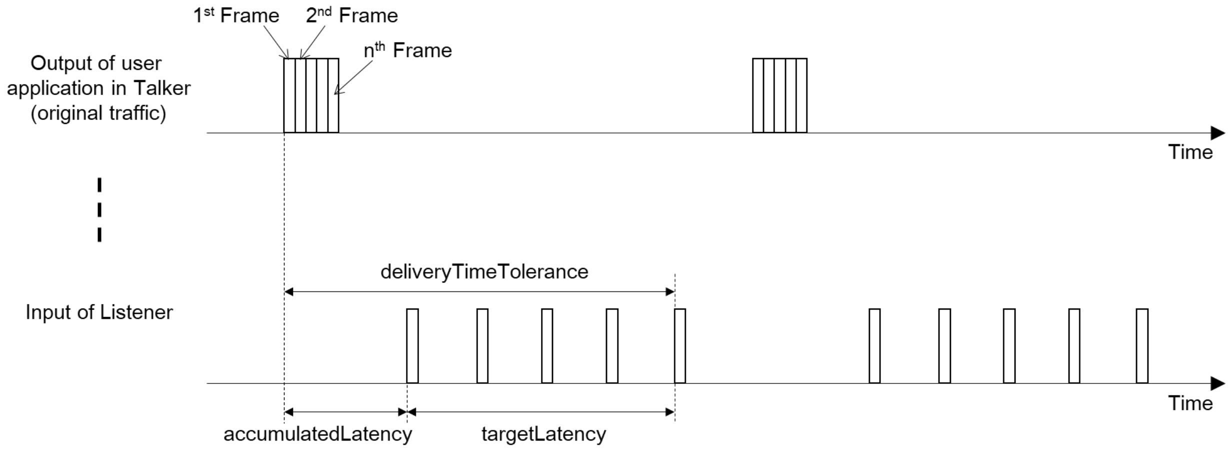

Figure Z-4— Frame propagation within delivery time tolerance while minimizing over-provision of bandwidth reservation

Z.3.1 Settings for MSRP TSpec

The MSRP TSpec is used in the credit-based shaper transmission selection algorithm. This type of TSpec is intended for use by reservations that compatibly supports AVB SR class A or SR class B. Unlike audio/video streaming, TSpec for bursty traffic, which characterizes the bandwidth that a stream can consume, needs to consider dataSize and targetLatency.

The TSpec parameters for MSRP are recommended to be set as follows:

$$MaxFrameSize = \min\left(\text{floor}\left(\frac{dataSize}{targetLatency} \times classMeasurementInterval\right), Maximum\ SDU\ Size\right) \quad (Z-5)$$

$$MaxIntervalFrames = \text{ceil}\left(\frac{1}{MaxFrameSize} \times \frac{dataSize}{targetLatency} \times classMeasurementInterval\right) \quad (Z-6)$$

The Maximum SDU (Service Data Unit) size is defined in (6.5.8). The MaxFrameSize is recommended to set the Maximum SDU Size. However, it should be smaller than the Maximum SDU size in case that classMeasurementInterval in (34.3) is shorter, i.e. the number of bytes within the classMeasurementInterval is smaller than the Maximum SDU size. The MaxIntrvalFrame needs to be guaranteed so as to become a positive integer (1 or lager value).

For the UNI TLVs, FirstValue defined in (35.2.2.10.6), and the values of TrafficSpecification TLV specified in (46.2.3.5):

<< Editor’s Note: see IEEE Std 802.1Qcc™-2018 >>

$$MaxFrameSize = \min\left(\text{floor}\left(\frac{dataSize}{targetLatency} \times Interval\right), Maximum\ SDU\ Size\right) \quad (Z-7)$$

$$MaxFramesPerInterval = \text{ceil}\left(\frac{1}{MaxFrameSize} \times \frac{dataSize}{targetLatency} \times Interval\right) \quad (Z-8)$$

The parameter “Interval” is referred in (46.2.3.5.1), which replaced classMeasurementInterval in Equations (Z-5) and (Z-6). The Interval is recommended to be set less than the delivery time tolerance for controlling the shaping rate during the shaping duration.

Z.3.2 Setting for Token Bucket TSpec

The TSpec parameters for Token bucket are used for Asynchronous Traffic Shaping (ATS). According to the definition of ATS scheduler state machine in Clause 8.6.11 (P802.1Qcr), CommittedBurstSize 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. CommittedInformationRate is the data rate reserved for the stream and is recommended to be equal to the requiredMinimumShapingRate shown in Equation (Z-3). The approximation discussed in Clause Z.3 can also be applied. These lead to the following settings values:

$$CommittedBurstSize = Maximum\ SDU\ Size \quad (Z-9)$$

$$CommittedInformationRate = \frac{dataSize}{targetLatency} \quad (Z-10)$$

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 CommittedBurstSize parameter (8.6.11.3.5), the CommittedBurstSize is set to be the maximum size of frame. That is equal to the Maximum SDU Size as shown in Equation (Z-9).

<< Editor’s Note: The ATS scheduler should also works for the case in which the CommittedBurstSize is greater than Maximum SDU Size. However, a small value of the CommittedBurstSize 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 cluster of frames 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. >>