Mapping method of QoS requirements to TSpec for bursty traffic shaping

Akio Hasegawa[1], Satoko Itaya[2], Fumihide Kojima[2], Hajime Koto[2], Kenichi Maruhashi[3], Hiroki Nakano[4], Ayano Ohnishi[1], Hiroshi Ohue[5], Takeo Onishi[3], Toru Osuga[2], Shinchi Sato[6], and Nader Zein[7]

1. Advanced Telecommunications Research Institute International,

2. National Institute of Information and Communications Technology, 3. NEC Corporation,

4. NETREQS Co., Ltd, 5. Panasonic Corporation, 6. Fujitsu Limited, 7. NEC Europe Ltd.(NLE GmbH)

Abstract

This paper clarifies how to set Traffic Specification (TSpec) for bursty traffic that has a delivery time tolerance. In the case of bursty traffic, measured throughput changes depending on observation interval and therefore TSpec parameters is difficult to be determined. Inappropriate TSpec setting causes over-provisioning of the bandwidth or makes it unable to satisfy the requirement of the delivery time tolerance. To address this issue, Tspec mapping method for bursty traffic is provided which reduces over-provisioning and satisfies requirement of the delivery time tolerance at same time. In the paper, Credit-Based Shaper (CBS) and Asynchronous Traffic Shaping (ATS) are considered for busty traffic shaping method.

1 Background

Nowadays, many IoT devices are being introduced in networks. In some cases, in a factory environment, relatively large data are transmitted and enter into the network as bursty traffic. In addition to the surge in traffic due to these IoT devices, many applications have time latency constraints. Use cases include: a) intelligent tools, e.g., torque wrench systems to check proper actions by torque-time profile [1], b) computer-vision-scanning equipment for inspection, e.g., camera to check product quality of semiconductor wafers using high resolution images [2], c) worker support, e.g., AR (Augmented Reality) / MR (Mixed Reality) for direction and monitoring [3], and so on. Similar situations, where the number of IoT devices are generating sporadic and busty traffic, are also expected in warehouses, hospitals, airports/stations, and other networks.

In these applications, many IoT devices transmit data in parallel. Therefore, when multiple streams arrive simultaneously at a bridge, congestion may occur momentarily and the delay time increases dramatically. This raises possibility that some streams may not arrive within their bounded latencies. By shaping bursty traffic and securing the required bandwidth, it is possible to mitigate the impact of congestion even if multiple streams arrive simultaneously at the same bridge, and therefore more streams can satisfy their bounded latencies.

Figure 1 shows an example of bursty traffic pattern. In this example, a group of frames so-called "bunch of frames" are transmitted intermittently, not continuously. The bunch of frames occurs sporadically, i.e., not periodically, implying T1≠T2 in Fig. 1.

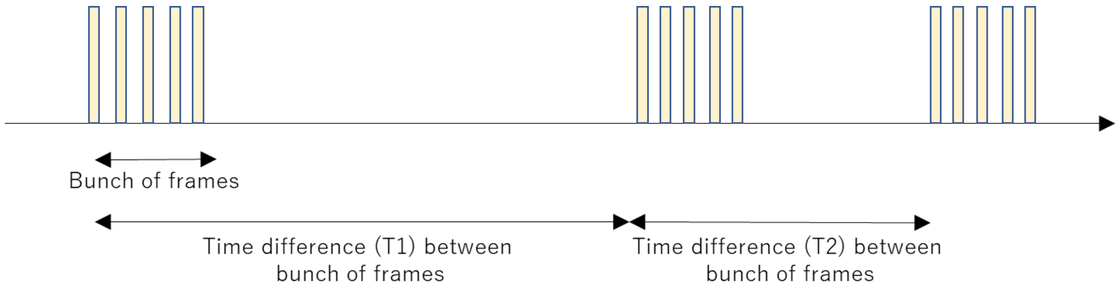

Fig.1. Example of bursty traffic pattern.

Each bunch of frames has delivery time tolerance. The delivery time tolerance is assumed to be pre-determined by the application or set manually by an operator of the application. It defines the maximum time from the fist bit sent from a sender of the application to the last bit received at a receiver of the application. The traffic is sporadic meaning that the next bunch of frames never come until the entire corresponding queue in a bridge becomes empty.

Figure 2 shows network configuration under consideration. This network comprises Talkers, Listeners, and bridges which connect directly or indirectly to each other. Each traffic is generated at a Talker, and is sent to the Listener via bridges on the route. The network between Talker and Listener is the subject of the IEEE Std 802.1Q architecture.

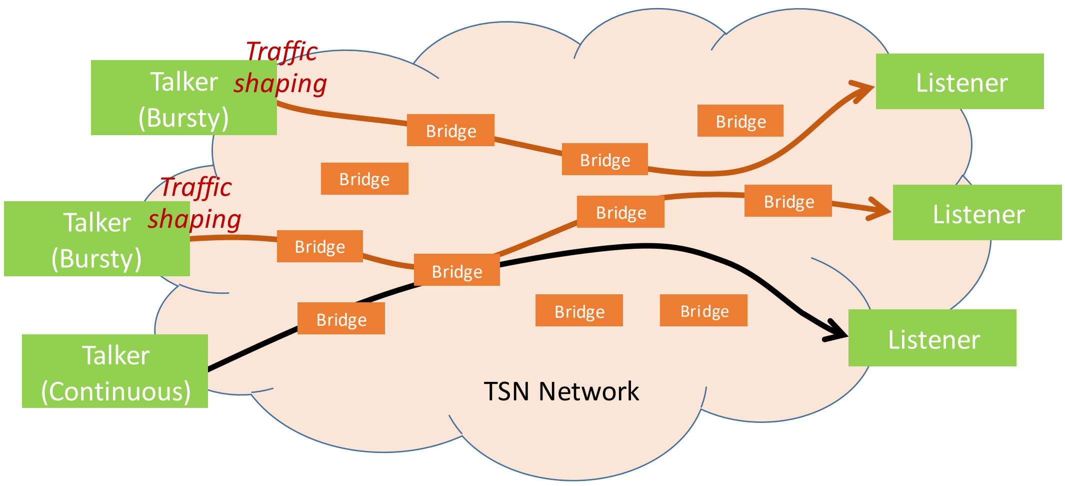

Fig.2. Network structure under consideration

There are multiple streams flowing through this network, and they may flow into a 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 described in Section 2. The Talker obtains information on application requirements from the application or the operator of the application.

2 Existing traffic shaping

In IEEE Std. 802.1Q, Credit-Based Shaper (CBS) is already specified and Asynchronous Traffic Shaping (ATS) will be included upon the completion of the project IEEE P802.1Qcr.

In IEEE Std. 802.1Q-2018, traffic specification (TSpec) is used for CBS in Forwarding and Queuing enhancements for Time-Sensitive Streams (FQTSS) and Stream Reservation Protocol (SRP). It characterizes end-point traffic which is a set of parameters consisting of the maximum number of bits per frame (MaxFrameSize) and the maximum number of frames transmitted in one classMeasurementInterval (MaxIntervalFrames), in a class measurement interval of 125 or 250 μs [4]. By appropriately configuring TSpec in the bridge, periodic traffic such as audio and video can be efficiently streamed along with added protection from other traffic. A bandwidth is calculated by using MaxIntervalFrames, MaxFrameSize and media-specific framing overhead per interval [5].

ATS uses TSpec Type 2 information to implement traffic shaping that does not require time synchronization [6]. In order to eliminate the need for time synchronization, ATS does not actually generate tokens, but calculates the time until tokens for one frame are collected using CommittedInformationRate of TSpec Type 2, by virtually implementing the token bucket algorithm to derive transmission time, i.e., eligibilityTime. By transmitting the frame according to the eligibilityTime, the ATS can shape the traffic at CommittedInformationRate.

3 Problem statement

It should be clarified that the problem, which the authors are addressing in this paper, is that there are no definition how to set TSpec and TSpec Type 2 for bursty traffic shaping.

For either CBS or ATS, TSpec parameters such as MaxIntervalFrames or CommittedInformationRate are important because they represent the flowing volume of streams. Therefore, an operator of the network needs to derive TSpec parameters appropriately. For the case of continuous traffic, Fig. 3(a) shows that the measured throughput is independent of the observation interval. On the other hand, for the case of bursty traffic, Fig.3(b) shows that throughput changes significantly depending on the observation interval. So, it is difficult to derive appropriate TSpec parameters for bursty traffic.

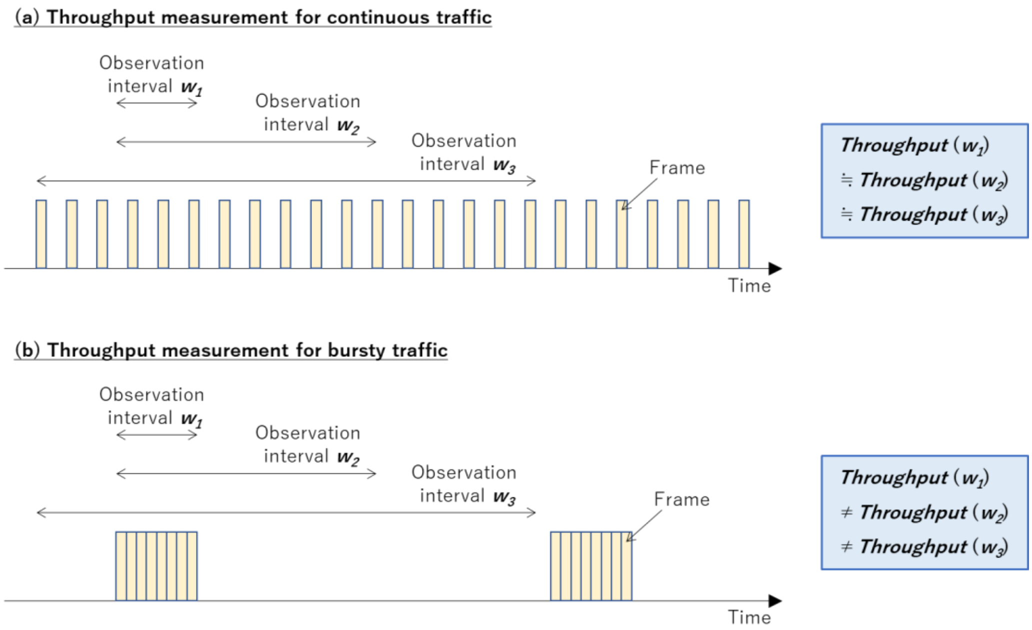

Fig. 3. Difference of measured throughput between for continuous and bursty traffic

When TSpec is derived based on the throughput measured over a short observation interval w1 as shown in Fig.3 (b), MaxIntervalFrames and CommittedInformationRate become higher than the minimum required value, resulting in excess bandwidth being reserved, and network resources are wasted (Over-provisioning). On the other hand, when TSpec is derived based on the throughput measured over a too long observation interval w3 as shown in Fig.3 (b), MaxIntervalFrames and CommittedInformationRate are underestimated with respect to the required flow rate, which results in the required delivery time tolerance not be satisfied. Thus, appropriate setting of interval for deriving TSpec (denoted by "calculation interval" afterward) is very important.

4 Mapping from QoS requirements to TSpec

In order to enable traffic shaping such as CBS or ATS to satisfy the requirement of the delivery time tolerance for bursty traffic, we propose a mapping method from data size of bunch of frames (Data Size) and Delivery Time Tolerance to TSpec or TSpec Type 2. Data Size and Delivery Time Tolerance are information from the application and are obtained by Talker.

The flow of frames from Talker to Listener is described in Fig.4. Bursty traffic is shaped by the Talker. As the result of traffic shaping, the interval in which the Talker sends each frame becomes "frame length divided" by "shaping rate". Then, at the output of a Listener to an application, the observed latency of this bunch of frames becomes as follows:

$$\text{observedLatency} = \text{propagationDelay} + \frac{\sum_{k=1}^{n-1} \text{frameLength}(k)}{\text{shapingRate}} \quad (1)$$

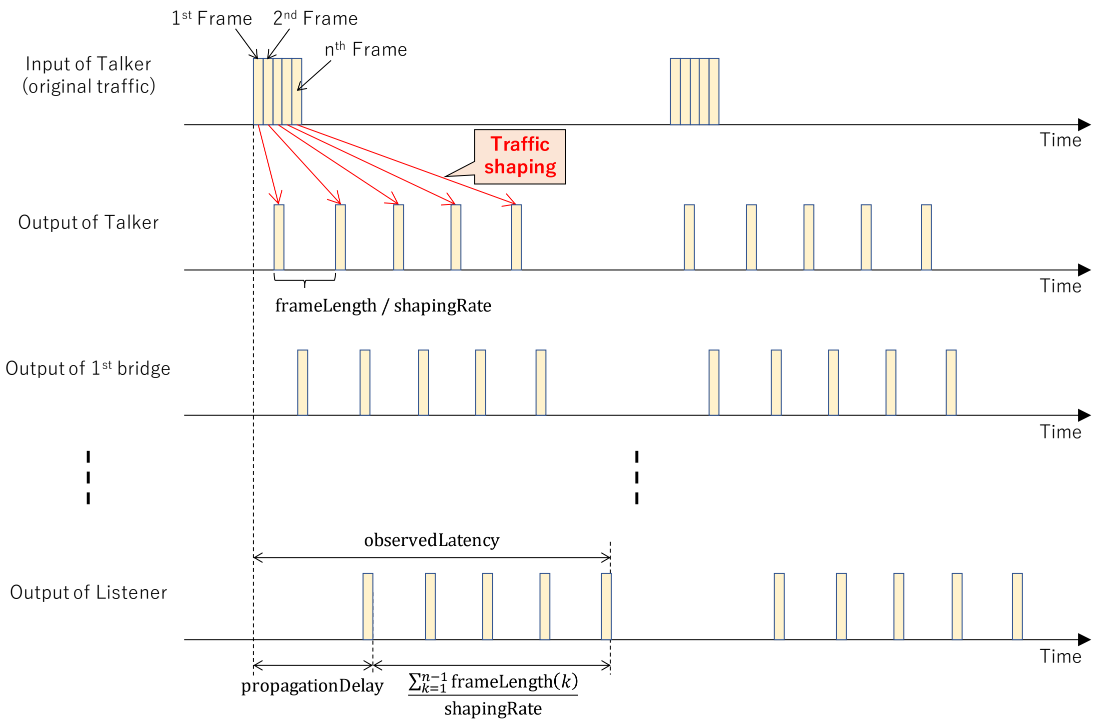

Fig.4. Frame propagation from Talker to Listener (general case)

In order to minimize over-provisioning while ensuring the requirement for Delivery Time Tolerance is met, the bursty traffic should be shaped until the observed latency becomes within the required Delivery Time Tolerance, as shown in Fig.5. Therefore, a value obtained by subtracting propagation delay from Delivery Time Tolerance is the appropriate calculation interval. We denote the appropriate calculation interval as Target Latency.

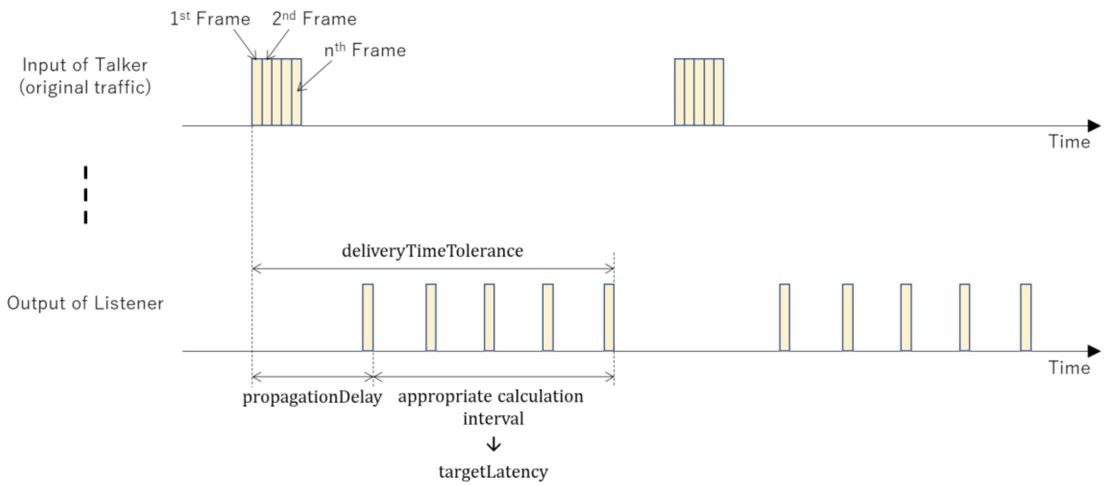

Fig.5. Frame propagation from Talker to Listener (appropriate case)

At a Talker, the Target Latency is calculated as follows using the sum of the propagation delays of the streams in all the bridges from the Talker to the Listener written in Annex V of IEEE P802.1Qcr/D2.0 [6]:

$$\text{targetLatency} = \text{deliveryTimeTolerance} - \left(\sum_{k=1}^{n} d_{MD,max}(k) + \sum_{k=2}^{n} d_{AT,max}(k) + \sum_{k=2}^{n} d_{PR,max}(k) \right) \quad (2)$$

Using Target Latency, the appropriate shaping rate is calculated as follows:

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

$$\Leftrightarrow \quad \text{appropriateShapingRate} = \frac{\sum_{k=1}^{n-1} \text{frameLength}(k)}{\text{targetLatency}}$$

$$= \frac{\text{dataSize} - \text{frameLength}(n)}{\text{targetLatency}} \quad (3)$$

Assuming the bunch of frames consists of a significantly large number of frames, the last frame in the bunch of frames is very small compared to Data Size. Therefore, the appropriate shaping rate can be approximated as follows:

$$\text{appropriateShapingRate} \fallingdotseq \frac{\text{dataSize}}{\text{targetLatency}} > \frac{\text{dataSize} - \text{frameLength}(n)}{\text{targetLatency}} \quad (4)$$

In order to set the CBS or ATS to shape at a rate equal to the appropriateShapingRate, Data Size and Target Latency should be mapped into TSpec or TSpec Type 2 as follows:

・**Mapping for TSpec**

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

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

The approximated appropriateShapingRate is slightly larger than the strict one as shown in Eq.(4). Thus, MaxFrameSize and MaxIntervalFrames become slightly larger. Using these TSpec parameters, the latency becomes shorter, and Delay Time Tolerance requirement is satisfied.

・**Mapping for TSpec Type 2**

$$\text{CommittedInformationRate} = \frac{\text{dataSize}}{\text{targetLatency}} \quad (7)$$

For the same reason as for TSpec above, Delay Time Tolerance requirement is satisfied.

5 Request for addition to standard

Since the mapping of application requirements to TSpec and TSpec Type 2 is undefined in the current IEEE Std. 802.1Q. It is suggested to add description of the mapping method for bursty traffic.

References

[1] Nendica Draft Report: "Flexible Factory IoT: Use Cases and Communication Requirements for Wired and Wireless Bridged Networks", https://mentor.ieee.org/802.1/dcn/19/1-19-0026-04-ICne.pdf

[2] https://www.wsj.com/articles/samsung-tests-how-5g-can-improve-chip-making-11565813658

[3] https://about.att.com/innovationblog/2019/06/5g_innovation_zone.html

[4] Clause 34.4 (Deriving actual bandwidth requirements from the size of the MSDU) ,IEEE Std 802.1Q™-2018, IEEE Standard for Local and Metropolitan Area Networks—Bridges and Bridged Networks

[5] Clause 35.2.2.8.4 (TSpec), IEEE Std 802.1Qcc™-2018, Amendment to IEEE Std 802.1Q™-2018 as amended by IEEE Std 802.1Qcp™-2018

[6] IEEE P802.1Qcr/D2.0, Draft Standard for Local and Metropolitan Area Networks—Bridges and Bridged Networks—Amendment: Asynchronous Traffic Shaping

Contacts

Contact: maruhashi@nec.com, Nader.Zein@EMEA.NEC.COM