Supporting Time-Sensitive and Best-Effort Traffic on a Common Metro Infrastructure

Luis Velasco and Marc Ruiz

***Abstract*—Considerable research and standardization efforts are being made to support time-sensitive traffic, e.g., generated by applications like Industry 4.0 and 5G fronthaul, on packet networks. This letter focuses on analyzing the impact of conveying time-sensitive traffic in operators' networks when such traffic is mixed with best-effort traffic. Extensions to a continuous $G/G/1/k$ queue model are proposed to evaluate two different Ethernet technologies, synchronous and asynchronous, supporting time-sensitive flows in terms of their influence on the performance of best-effort traffic. Results highlight pros and cons of those technologies to protect best-effort performance.**

***Index Terms*—Terms—Time-Sensitive Networking; Deterministic Networking; Synchronous and asynchronous Ethernet.**

I. Introduction

THE needed support for deterministic communication has been fostered by the development of the fourth industrial revolution (including Industry 4.0) and the introduction of new network technologies (e.g., like 5G and other wireless technologies). Two complementary sets of standards have focused on providing bounded Quality of Service (QoS) in terms of latency (delay), loss, and delay variation, as well as high reliability: the IEEE 802.1 Time-Sensitive Networking (TSN) [1] focuses on providing deterministic communication on standard Ethernet, while the IETF extends the scope to Layer 3 [2]; see the tutorial in [3]. The TSN technology can be centrally managed; a controller (named Central Network Controller-CNC- in the IEEE 802.1 terminology) computes the paths for *TSN flows* at requesting time and defines the schedule for TSN frames. To guarantee the throughput and delay QoS requirements, capacity (throughput and buffer) allocation in every TSN switch along the path of each flow is performed (see [4] for a bounded latency model).

Although the initial goal of both working groups has been focused on closed environments, interest to extend their scope to provide end-to-end solutions is increasing and some works can be found with case studies that include the application of TSN to Industry 4.0 and 5G fronthaul (see, e.g., [5], [6]). End-to-end TSN services entail the support of operators' transport networks that are currently carrying traffic from users, business, and datacenter, just to mention a few on a Best Effort (BE) basis; such traffic is commonly encapsulated into Multiprotocol Label Switching (MPLS) Label-Switched Paths (LSP) at Layer 2 for traffic engineering purposes.

As soon as operators' transport networks provide support to TSN traffic, scheduling and traffic shaping will guarantee the committed QoS of TSN flows and allow for the coexistence of TSN and BE traffic on the same network infrastructure. One problem of providing end-to-end TSN flows is synchronization of multiple administrative domains. Such synchronization requirements can be relaxed by adopting the IEEE 802.1Qcr standard that allows for asynchronous traffic shaping. In addition, TSN and BE traffic coexistence could have negative impact on the BE traffic, as resources allocated to TSN flows would reduce those available for the former. A solution to avoid such negative influence is to use separated physical (i.e., ports and links) or isolated virtual (*hard* slice) resources at the cost of overprovisioning. Another option that does not require network synchronization is the Integrated Hybrid Optical Network (IHON) demonstrated in [7], which is able to aggregate TSN traffic and detect and identify inter-packet gaps to insert BE traffic in between.

TSN and BE traffic mix scenarios can be modeled using the queuing theory with server variations. In this regard, a large number of works can be found in the literature considering server vacations and time-dependent breakdowns that limit the availability of the server (see, e.g., [8]). Such models can be used to derive expressions for queue system analysis, e.g., to estimate the mean processing time of a single entity in the queue. However, these models present limitations when applied to different distributions for both input traffic and server rates, and they can be hardly used to model complex systems with multiple queues and traffic mixes.

A different efficient flow-based approach was proposed by the authors in [9] based on a continuous queuing model for network flows analysis, named CURSA-SQ. The CURSA-SQ queue model is a continuous $G/G/1/k$ model with a First-In-First-Out (FIFO) discipline based on the logistic function. CURSA-SQ can be applied for a wide range of scenarios, such as generating realistic data for Machine Learning training purposes [10] and for accurate, scalable, and predictive near real-time estimation of end-to-end KPIs in fixed and converged fixed-mobile networks [11]. In this letter, we extend CURSA-SQ to model network interfaces supporting both BE traffic and TSN simultaneously under different TSN standards. The extended model is used to accurately estimate throughput and delay of each traffic flow, which can be used, e.g., for off-line network planning, and route selection during flow provisioning and re-optimization.

The rest of the letter is organized as follows. To motivate studying the impact on the BE traffic, Section II presents

Manuscript received February 6, 2020.

The research leading to these results has received funding from the European Commission for the H2020-ICT-2016-2 METRO-HAUL project (G.A. 761727), from the AEI/FEDER TWINS project (TEC2017-90097-R), and from the Catalan Institution for Research and Advanced Studies (ICREA).

Luis Velasco (lvelasco@ac.upc.edu) and Marc Ruiz are with the Optical Communications Group (GCO), at Universitat Politècnica de Catalunya (UPC), Barcelona, Spain.

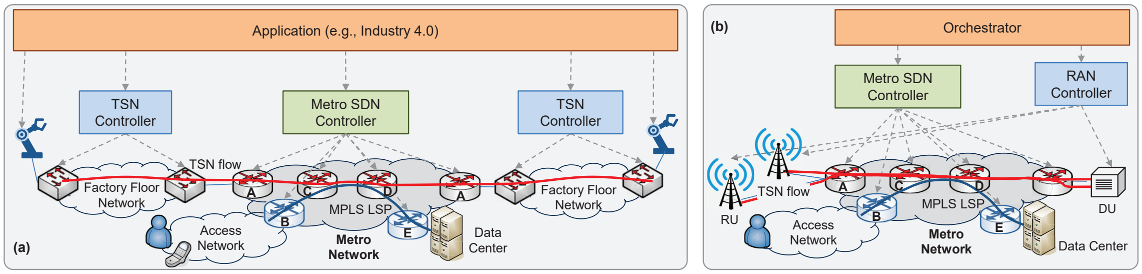

Fig. 1. Illustrative architectures supporting TSN and BE traffic over a Metro network. TSN traffic from Industry 4.0 (a) and RAN fronthaul (b).

scenarios where TSN traffic is mixed with BE traffic on a metro network. Section III proposes extensions to CURSA-SQ for synchronous and asynchronous TSN models. The discussion is supported by the numerical results in Section IV. Finally, Section V draws the main conclusions.

II. Providing end-to-end TSN flows over a non-dedicated Transport Metro Network

Let us assume that a telecom metro transport network supports both TSN and BE traffic flows. Fig. 1 presents two illustrative scenarios where TSN flows are generated by an Industry 4.0 application communicating factory floor networks (Fig. 1a) and for 5G fronthaul between Radio Units (RU) and Distributed Units (DU) [12]] (Fig. 1b). Typically, Industry 4.0 applications generate low bitrate (e.g., 100 Mb/s) flows, whereas 5G fronthaul flows are of higher bitrate (e.g., $1Gb/s$). In both cases, the TSN flows are conveyed on a metro network, which also supports BE traffic based on MPLS. In particular, nodes C and D in the metro network in Fig. 1a-b support a mix of TSN and BE traffic. We assume that a Software-Defined Networking (SDN) controller is in charge of provisioning MPLS LSPs for BE traffic, as well as for TSN flows on the metro network. In addition, other managers and controllers might be used for the factory networks and the Radio Access Network (RAN), as well as for end-to-end resource orchestration (omitted for the sake of simplicity).

When the application/orchestrator makes a provisioning request for TSN flows, it specifies the requirements for those flows; the SDN metro controller determines whether the QoS requirements can be served and computes the paths and the scheduling. However, even though the TSN request can be served, an evaluation should be carried out to determine how serving the new TSN request will impact on the BE services. In the case that not enough resources are currently available in the metro network to guarantee a low impact on the BE traffic, reconfiguration can be carried out prior accepting the request.

As previously introduced, there are different approaches to support TSN and each of them have different impact on the BE traffic. In the *synchronous* TSN model defined in IEEE 802.1Qbv, time slices are reserved to TSN traffic flows, whereas BE traffic is transmitted in between of consecutive protected TSN slices. Note that this option, although ensures the QoS of TSN flows, it might limit that of the BE traffic in the case that the protected time slices are not fully consumed by the TSN traffic. In addition, a guard band time is needed before the protected time slices to avoid contention, which reduces even more the time available for transmitting BE traffic. To improve that, the IEEE 802.1Qbu/802.3br standards enable splitting the transmission of a BE frame between two protected time slices. These standards, together with additional features, define profile B in the IEEE 802.1CM standard, oriented to transport fronthaul flows based on the (evolved) Common Public Radio Interface (eCPRI) [13] over Ethernet.

A different impact on the BE traffic can be expected from *asynchronous* TSN models like IEEE 802.1Qcr and IHON, as TSN flows use exactly the transmission time that they need and inter-packet gaps can be filled with BE frames of suitable size, which maximizes BE throughput and reduces its latency.

In the next section, we extend CURSA-SQ to model the technologies described above to reproduce TSN traffic, as well as BE traffic flows, aiming at studying the impact that mixing both traffics would have on the latter.

III. CURSA SQ Extensions for Time Sensitive Traffic

For the sake of clarity, eq. (1) presents the compact basic differential equation that models the dynamics of the state of a capacitated continuous queue system, denoted as $q(t)$ $(bytes)$, supporting a single flow (b/s) (see [9] for further details). Let $\hat{X}(\cdot)$ be the amount of input flow received and actually stored in the queue at time t (i.e., it depends on the state of the queue) and let $Y(\cdot)$ be the flow (b/s) leaving the queue, which depends on the state of the queue, as well as on the fixed server rate μ.

$$\frac{d}{dt}(q_i(t)) = q^{'}(t) = \hat{X}(q(t),t) - Y(q(t),\mu) \tag{1}$$

In this letter, however, we assume the scheme of the network interface detailed in Fig. 2a, where n individual TSN input flows (X_{TSN}) that arrive conveniently shaped, are combined with an aggregated input BE flow (X_{BE}). Each individual flow i is associated to one continuous capacitated queue system, where its state $q_i(t)$ depends on the input flow and on a server rate $\mu_i(t)$ variable with time; all the individual queue systems access the network interface characterized by a fixed server rate μ. Then, the extended differential equation that characterizes every individual flow in the interface is:

$$q_i^{'}(t) = \hat{X}(q_i(t),t)) - Y(q_i(t),\mu_i(t), \forall i \in TSN \cup \{BE\} \tag{2}$$

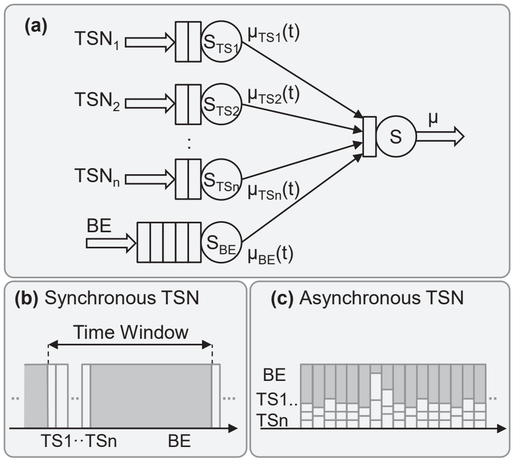

Fig. 2. Modeling TSN and BE traffic mixing

At this point, the key difference between BE and TSN flows is in terms of the instant server rate $\mu_{(\cdot)}(t)$; while the server rate for TSN flows is a function $f_i(\cdot)$ that depends on their own configuration parameters θ_i only (eq. (3)), that for the BE flow depends on the remaining resources available after assigning resources to each TSN flow i (eq. (4)), as resources are reserved for TSN flows at provisioning time.

$$\mu_i(t) = f_i(t; \theta_i), \forall i \in TSN \tag{3}$$

$$\mu_{BE}(t) = \mu - \sum_{i \in TSN} f_i(t|\theta_i) \tag{4}$$

$$f_i^{sync}(t; t_i, T_i) = \mu \cdot \pi(\frac{t\%T - t_i}{T_i}), \forall i \in TSN \tag{5}$$

$$f_i^{async}(t) \cong X_i(t), \forall i \in TSN \tag{6}$$

Two TSN models relating queue systems and network interfaces are defined based on the IEEE 802.1 standards: the *Synchronous TSN* (*sync*) and the *Asynchronous TSN* (*async*) model. For each TSN model, functions $f_i(\cdot)$ are detailed next.

The *Synchronous TSN* (*sync*) model targets at modeling the IEEE 802.1Qbv standard. It defines a time window of fixed length (T) and reserves time slices for every TSN flow; the rest of the time window that remains unassigned can be used to BE traffic (Fig. 2b). Thus, configuration parameters θ_i in the sync model are defined in terms of time. Equation (5) describes the server rate of every flow under the sync TSN model, where $\pi(\cdot)$ is the rectangular function that is 1 if the input value is in [+1/2, -1/2] and 0 otherwise, t_i is the center, and T_i the length of the time slice assigned to the flow. For the TSN flows, T_i is assigned to serve the expected input traffic rate X_{TSN} in T. Then, the summation of $f_i(\cdot)$ in eq. (4) results in μ if one of the TSN flows is served, which prevents the BE flow to be served at time t, and 0 otherwise, which gives the BE flow the whole capacity of the interface at that time.

On the other hand, the *Asynchronous TSN* (*async*) model targets at modelling IEEE 802.1Qcr. In this model, the TSN flows have higher priority than the BE one. Time is processed in small fragments of fixed size, where at every fragment the state of the queues is evaluated and served according to their priority (Fig. 2c). To avoid the recurrence caused by evaluating such queues state, f_{TSN} is conveniently approximated to the current input traffic rate X_{TSN} (eq. (6)). This decreases the available server capacity for the BE flow.

IV. Illustrative Numerical Results

In this section, we first assess the accuracy of the proposed extensions to CURSA-SQ for modelling TSN and BE traffic mixing and then, we illustrate the applicability of the proposed TSN modelling approaches through numerical studies focused on analyzing the impact of the TSN traffic over the BE one. Numerical results have been obtained by solving eq. (2) using an ordinary differential equation (ODE) solver implementing the Dormand-Prince method [14]. For the studies, we have reproduced a scenario where TSN traffic and BE traffic arrive to a node interface where they are combined; node C in Fig. 1a-b. We assume that TSN and BE flows are forwarded through a single $100Gb/s$ interface and the *sync* TSN model was dimensioned with a time window of length $T = 125$.

A. CURSA-SQ extensions validation

As already introduced, queue models with server vacation and breakdowns can be used to estimate the mean processing time g of a single entity in the queue. Such models include the fraction of time that the server is available (ρ) as a parameter to characterize the behavior of the queue; from [8], $g \approx 1/(\rho \cdot \mu)$, where μ is the maximum server rate. We can apply the previous equation for BE traffic under *sync* and *async* approaches just modeling ρ. Under the *sync* approach, ρ_{sync} can be fairly computed as the proportion of the remaining time within time window T available for BE traffic, i.e., before allocating reserved slices for TSN flows and the guard band. In the case of the *async* approach, ρ_{async}, can be approximated to the proportion of the remaining interface capacity after subtracting the aggregation of the TSN traffic flows.

To validate the CURSA-SQ model, we carried out a numerical evaluation assuming 500 100 Mb/s TSN flows (50 Gb/s aggregated TSN traffic) and $0.2\mu s$ time slices under the *sync* TSN model, so $\rho_{sync} = (125 - 100)/125 = 0.2$. For the *async* TSN model, $\rho_{async} = (100 - 50)/100 = 0.50$. To relax any assumption on the characteristics of the incoming BE traffic, which could reduce the applicability of the reference model, we loaded the BE traffic queue and solved CURSA-SQ model to compute the time needed to process all queued traffic. In addition, discrete *sync* and *async* simulation was run filling the queue with 1500-byte packets [9]. Calculations were repeated for a wide range of k bytes in the queue.

Fig. 3a shows the obtained results for the BE traffic, where it can be observed that all CURSA-SQ, approximation model and discrete simulation provide very similar values under *sync* and *async* TSN models (maximum discrepancy of 210 μs). In light of this, we conclude that the proposed extensions to the CURSA-SQ continuous model are accurate and can be used for the analysis of the BE traffic performance.

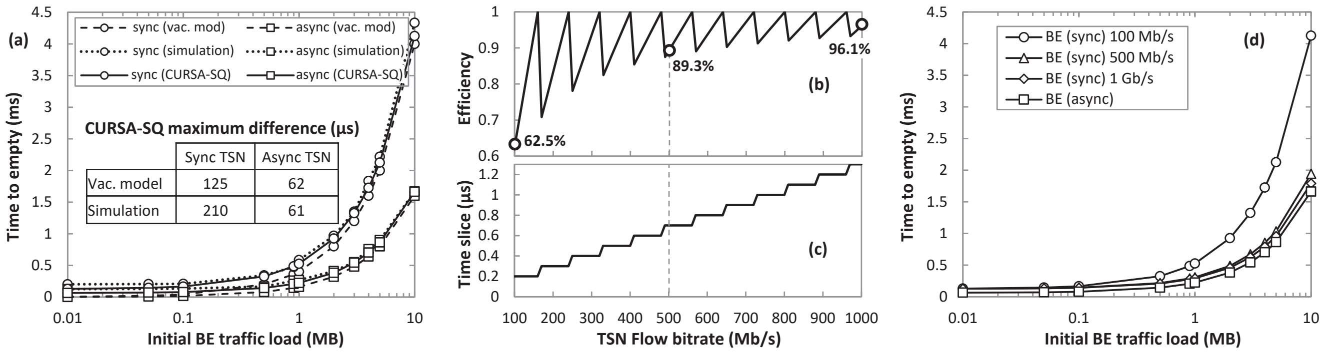

Fig. 3. (a) Accuracy of CURSA-SQ extensions. (b) Efficiency and time slice vs flow bitrate. (c) Impact of efficiency on sync TSN

B. Study of the impact of TSN traffic

Once the proposed extensions of CURSA-SQ for the targeted scenarios have been assessed, let us first study the impact of the duration of the time slice under the *sync* TSN model. We assume $0.1\mu s$ resolution (i.e., approximately the time to send a 1500-byte Ethernet frame over a $100Gb/s$ interface). Under this assumption, Fig. 3b plots the efficiency and Fig. 3c the duration of the time slice as a function of the TSN flow bitrate, from $100Mb/s$ (e.g., from an Industry 4.0 application) to $1Gb/s$ (e.g., from 5G fronthaul); the efficiency for $100Mb/s$ is as low as 62.5% and increases to 89.3% for $500Mb/s$ and to 96.1% for $1Gb/s$ flows. The impact of the efficiency of the selected time slice on the time needed to process the queued BE traffic is shown in Fig. 3d for the above bitrates and time slice efficiency. In view of this figure, we conclude that the *sync* TSN model can potentially provide similar results to those from the *async* one, provided that high efficiency is achieved with the selected time slot.

Let us now focus on evaluating relevant QoS performance metrics such as delay and delay variation introduced by packet nodes where both TSN and BE traffics are mixed. We consider that the TSN traffic consists of a variable number of $100Mb/s$ (Industry 4.0) or $1Gb/s$ (5G fronthaul) flows and maximum latency of $100\mu s$. Regarding the BE traffic, we applied the configuration and statistical properties in [9] to generate aggregated traffic flows with a Gaussian probability distribution that mixes users of background Internet services, as well as other specific services such as Video-On-Demand and online Gaming. Under such scenario, let us first analyze the impact on the probability distribution of the *sync* and *async* TSN models. Fig. 4a plots the Cumulative Distribution Function (CDF) at the input for the TSN and BE flows for the normalized load 0.8 with respect to the capacity of the interface ($100Gb/s$) and a mix 50-50, respectively (i.e., either 400 $100Mb/s$ or 40 $1Gb/s$ TSN flows, with $40Gb/s$ on-average BE traffic flows). The TSN traffic follows a clear deterministic *on-off* pattern, whereas a Gaussian distribution can be observed for the BE. The CDF of the output traffic flows is plotted in Fig. 4b, where it can be observed that the *sync* TSN model shapes the BE traffic, i.e., increases its variability and adds an on-off pattern. On the opposite, the *async* TSN model respects BE CDF in general terms.

Let us next analyze the QoS performance metrics of BE and TSN traffics for different normalized offered loads. Fig. 4c-d present the delay and the delay variation ratio (DVR) (computed as the ratio between maximum and average delay), respectively of the TSN and BE traffic under the two TSN models. In the graphs, the interface was loaded only with TSN traffic until reaching a normalized load equal to 0.4. Then, the TSN traffic load remained stable and BE traffic was injected increasing the normalized load from 0.5 to 1. In view of the plots, it is clear that both TSN models can perfectly transport the TSN traffic, as virtually zero delay and DVR are added when the TSN traffic is separated or mixed with BE traffic.

The effects on the BE, however, are very different and depend on the TSN flows configured in the case of the sync TSN model. The *a¡sync* TSN model with $100Mb/s$ TSN flows is very sensitive to the efficiency and it imposes a high delay to the BE traffic, even for low loads. However, the performance of the *sync* TSN model with $1Gb/s$ TSN flows is close to that of the *async* TSN model; both are able to use the resources remaining after serving the TSN traffic in a way that does not add perceptible delay to the BE traffic. For instance, at the normalized load 0.8, the delay experienced by the BE traffic under the *sync* TSN model with $100Mb/s$ flows would be around $300ms$, whereas it would be only $115\mu s$ with $1Gb/s$ flows, and $56\mu s$ under the *async* TSN model. The opposite effect can be observed for the DVR, where the *sync* TSN model with $100Mb/s$ TSN flows obtains values very close to 1, as result that the maximum and average delays get the same value (the BE traffic is close to saturation). In contrast, the *sync* TSN model with $1Gb/s$ flows and the *async* TSN model increases DVR by a factor of 3 for the BE traffic. Note that BE traffic is in general, tolerant to DVR, so such increment should not affect the performance of the related services.

Finally, let us analyze the QoS performance of both traffics when the total normalized offered load is kept constant to 0.8; here, the proportion of TSN traffic with respect to the total ranges from $0-0.8$, while the BE traffic reduces proportionally from $1-0.2$. Fig. 4e-f plot the obtained delay and DVR, where it can be observed that the delay experienced by the BE traffic under the sync TSN model with $100Mb/s$ flows highly increases when TSN traffic is injected. Interestingly, the opposite effect is observed with $1Gb/s$ flows and under the *async* TSN model, as the mean delay reduces in line with the BE input traffic reduction; this is the expected behavior

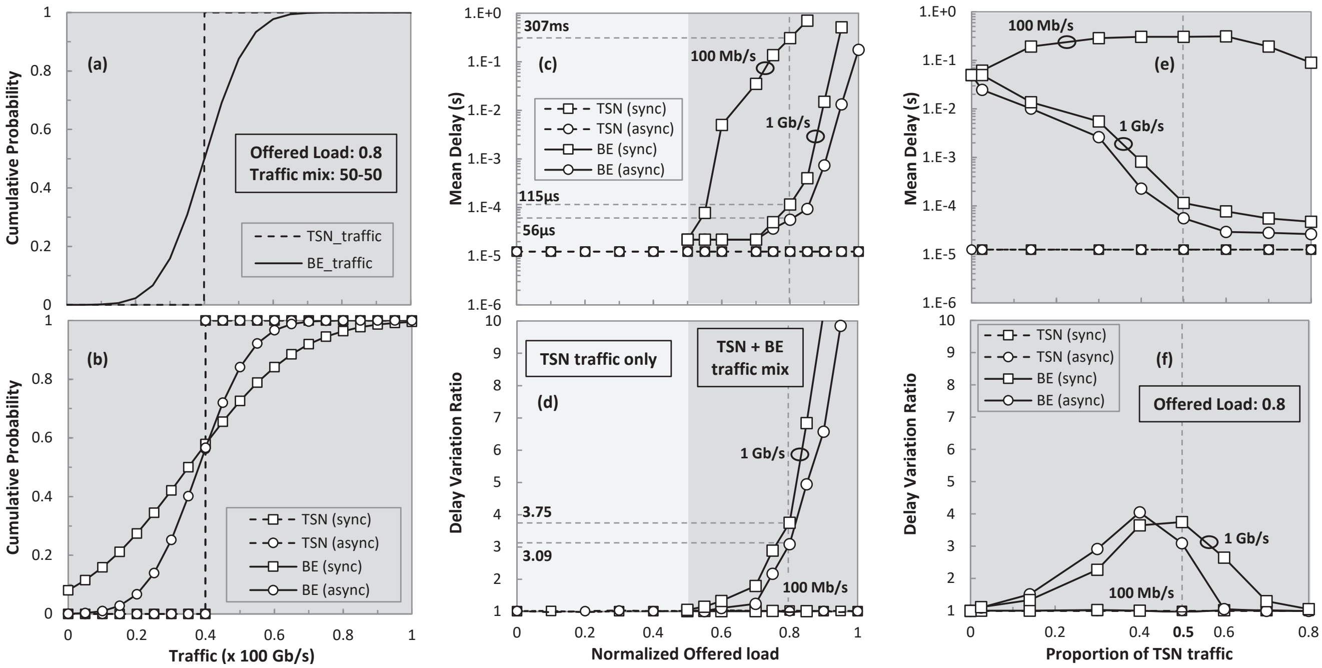

Fig. 4. Performance of TSN and BE traffic when they are mixed on a common network interface

when traffic reduces in a queue system with limited but enough capacity, and μ_{BE} adapts much better to the input traffic needs. Here DVR increases when the proportion of BE traffic reduces due to the increased traffic flow variance and the high load and decreases when BE traffic load reaches the point where enough resources are available; note that the results are in line with those in [15] for Ethernet passive optical networks.

V. Concluding Remarks

In view of the current efforts to standardize technologies to transport TSN flows over Ethernet, generated by applications like Industry 4.0 and 5G fronthaul, extensions for an efficient continuous $G/G/1/k$ queue model have been proposed aiming at studying the impact that serving such TSN flows over a common metro network that conveys BE traffic. Note that such coexistence would reduce costs for network operators coming from resource overprovisioning. Based on the current IEEE 802.1 standards, the extensions support two TSN models: *i*) *synchronous* TSN, based on IEEE 802.1Qbv and currently proposed as a solution for 5G fronthaul, and *ii*) *asynchronous*, based on IEEE 802.1Qcr.

The results show that both *synchronous* and *asynchronous* TSN models perfectly meet the requirements of TSN traffic flows. Nonetheless, the effects on the BE traffic are very different, as the *sync* TSN model shapes the BE traffic by adding *on-off* periods. In addition, the performance of the BE traffic heavily depends on the efficiency reached with the selected time slice in the cased of the *sync* TSN model and it can be as high as that under the *async* TSN model. Therefore, the sensitivity of services based on BE to the conditions imposed by the TSN model selected should be studied to assure low impact on the Quality of Experience.

In conclusion, the *sync* TSN model includes several parameters that need to be tuned (e.g., those related to time slices), which makes it noticeably adaptable to a large range of specific scenarios, but also makes its application more difficult to scenarios with traffic mixes, as it requires from a careful study of its configuration.

References

[1] IEEE Time-Sensitive Networking Task Group. [On-line] http://www.ieee802.org/1/pages/tsn.html.

[2] IETF Deterministic Networking Working Group [On-line] https://datatracker.ietf.org/wg/detnet/about/.

[3] A. Nasrallah *et al.*, "Ultra-Low Latency (ULL) Networks: The IEEE TSN and IETF DetNet Standards and Related 5G ULL Research," *IEEE Comm. Surveys Tutorials*, vol. 21, pp. 88-145, 2019.

[4] N. Finn *et al.* (Eds.) "DetNet Bounded Latency," IETF draft work-in-progress, 2019.

[5] Y. Pointurier *et al.*, "End-to-End Time-Sensitive Optical Networking: Challenges and Solutions," *IEEE/OSA Journal of Lightwave Technology (JLT)*, vol 37, pp. 1732-1741, 2019.

[6] J. Zou, S. Sasu, J. Messenger, J.P. Elbers, "Options for time-sensitive networking for 5G fronthaul," in Proc. ECOC, 2019.

[7] R. Veisllari, S. Bjornstad, J. Braute, "Experimental Demonstration of 100 Gb/s Optical Network Transport and Aggregation for Ethernet Fronthaul with Low and Bounded Delay," in Proc. OFC, 2018.

[8] M. Sikha, R. Manivasakan, "On the Inter-Departure Times in M/D/1/Bon Queue with Queue-Length Dependent Service and Deterministic/Exponential Vacations," *IEEE Access*, vol. 7, pp. 1439-1453, 2019.

[9] M. Ruiz, F. Coltraro, and L. Velasco, "CURSA-SQ: A Methodology for Service-Centric Traffic Flow Analysis," *IEEE/OSA J. Optical Comm. and Netw. (JOCN)*, vol. 10, pp. 773-784, 2018.

[10] D. Rafique and L. Velasco, "Machine Learning for Optical Network Automation: Overview, Architecture and Applications," *IEEE/OSA J. Optical Comm. and Netw. (JOCN)*, vol. 10, pp. D126-D143, 2018.

[11] A. Bernal *et al.*, "Near Real-Time Estimation of End-to-End Performance in Converged Fixed-Mobile Networks," *Elsevier Computer Communications*, vol. 150, pp. 393-404, 2020.

[12] "NG-RAN; Architecture description," *ETSI TS 138 401*, v15.2, 2018.

[13] Common Public Radio Interface (CPRI). [On-line] http://www.cpri.info/.

[14] J. Dormand and P. Prince, "A family of embedded Runge–Kutta formulae," *J. of Comp. and Applied Maths*, vol. 6, pp. 19-26, 1980.

[15] R. Raad *et al.*, "DBA algorithms for improved throughput and latency performance in CDMA-based next-generation ethernet PONs," *OSA Continuum*, vol. 2, pp. 3107-3126, 2019.