

Provisioning of Time-Sensitive and non-Time-Sensitive Flows on a Common Infrastructure

L. Velasco and M. Ruiz

Optical Communications Group (GCO), Universitat Politècnica de Catalunya (UPC), Barcelona, Spain;

**e-mail: luis.velasco@upc.edu*

ABSTRACT

Provisioning Time-Sensitive (TS) flows on dedicated infrastructures requires just calculating the time-slot dedicated to that service, which is then allocated in the superframes used by the schedulers running in the different nodes in the network. In such scenario, there is no impact on the flows already being served by the network as every TS flow has dedicated resources supporting it. However, in heterogeneous scenarios supporting TS and non-TS flows, the impact on existing non-TS flows can be very severe, and needs to be analyzed. In this paper, we overview a provisioning process that includes the use of a Network Flow Scheduler for reserving time slots for TS flows and a Digital Twin to accurately estimate the performance of the new and already deployed non-TS flows.

Keywords: Digital twin, QoS estimation, Time-Sensitive Networking

1. INTRODUCTION

Time-Sensitive Networking (TSN) consists of a set of standards defined by the IEEE 802.1 working group [1], which includes the 802.1Qbv time-aware scheduler. That scheduler works by defining a superframe (SF) of fixed length as a set of time slots that repeats over time. Each time slot can be assigned to a single flow so as to guarantee that Time-Sensitive (TS) flows meet the required performance, defined in terms of Key Performance Indicators (KPI), such as end-to-end (e2e) delay and delay variation (jitter). In this way, traffic flows of multiple classes can coexist. The allocation of such time slots, however, needs to be faced from a network perspective to ensure e2e performance. In that regard, the authors in [2] studied the combinability of multiple TS flows, while leaving resources available for non-TS flows. They proposed the deterministic network scheduling for TS flows, a planning problem to be solved beforehand for all the flows. Once resources are allocated to TS flows, packet schedulers assign resources to non-TS flows dynamically.

Our approach is different, as we target a more realistic scenario where individual flow provisioning requests arrive and each one needs to be accepted or rejected based on the possibility to provide the required performance for the new flow request, as well as to ensure that the performance of already established TS and non-TS flows is guaranteed. In fact, the use of a *Network Digital Twin* (NDT) can be of paramount importance to estimate the performance of traffic flows near-real-time, as shown in our previous work in [3], where both service traffic and queues behavior in packet nodes were modeled. Extensions proposed in [4] consider time awareness, thus supporting IEEE 802.1Qbv. In this work, we present a solution based on the above elements and show how they need to coordinate to operate a TS infrastructure that guarantees performance. The *TS Flow Scheduler Planner* (TS FSP) plans TS flow time windows across a defined path and the NDT evaluates the KPIs on non-TS flows. Both, TS FSP and NDT work assume a worst-case scenario, so as the performance of the flows is guaranteed.

2. INTRODUCTION

Fig. 1a presents the heterogenous TSN network scenario considered in this paper that includes network nodes with and without TSN capabilities. Although the network supports both TS and non-TS packet flows, which are mixed in some of the network interfaces, TS flows are exclusively supported through TSN-capable devices. An illustrative example is presented in Fig. 1b, which includes two TSN-capable WiFi access points (AP), three TSN Ethernet switches, and two non-TSN capable packet routers. The network connects two robotic arms, two servers, and a number of users. Two TS flows (denoted TS-1 and TS-2) are routed through a path connecting the robotic arms to their controller running in Server A. Additionally, one of the robotic arms generates a video flow that requires some QoS performance (QoS-1). In another part of the network, the users are connected to Server B, although that traffic is considered of low priority, e.g., best effort (BE).

To control such a heterogenous network, we rely on SDN controllers with (denoted TSN controller) and without TSN capabilities, which use a south-bound interface to program the different network devices under their control. A TSN Connectivity Manager (CM) provides e2e control and includes, among other components: *i*) a *Path computation element* (PCE) implementing algorithms with different policies that are applied as a function of the type of flow that needs to be provisioned; *ii*) a TS-FSP in charge of producing worst-case scheduling for the TS flows to be deployed in the network; and *iii*) a NDT that evaluates a set of KPIs of non-TS flows before new (TS or non-TS) flows are deployed. The NDT considers non-TS flows with different priorities, e.g., QoS committed and BE. Note that although all flows are served on a particular path, TS flows have specific resources that are reserved along their path, whereas non-TS ones use the remaining resources, which are assigned by their priority.

This work was partially funded by the European Union through the NextGeneration UNICO5G TIMING (TSI-063000-2021-145) project.

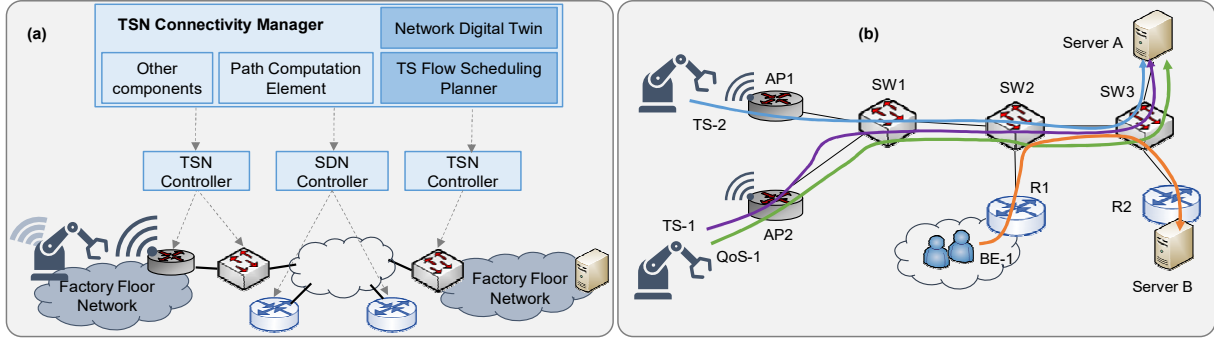


Fig. 1: Heterogenous TSN network and overview of the proposed architecture (a). Illustrative scenario (b).

When a new TS flow request arrives at the TSN CM, a provisioning process is followed that includes path computation, scheduling planning (in the case of a TS flow), and performance evaluation. In the case that the flow request is accepted, the TSN CM uses SDN controllers' north-bound interfaces to send them precise instructions for the new flow. Specifically, in the case of a TS flow, the TSN CM sends the computed network scheduling plan to the TSN controllers that will subsequently provide that plan to the packet schedulers running in the TSN-capable nodes. In the case of a non-TS flow, SDN controllers might be also involved in the provisioning process.

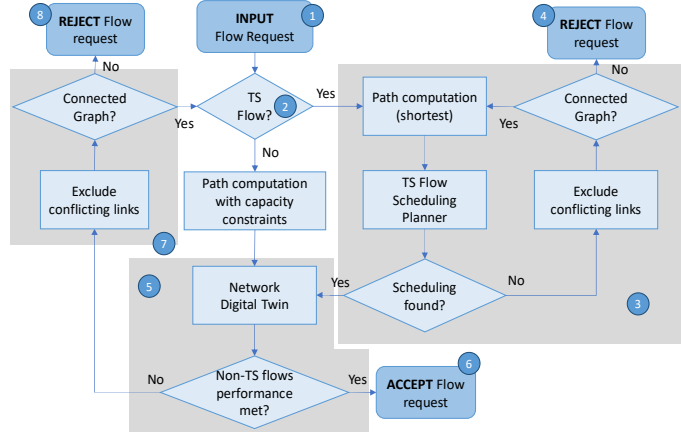


Fig. 2: Provisioning algorithm for TS and non-TS Flows

3. FLOW SCHEDULING AND PERFORMANCE ESTIMATION

This section overviews the provisioning process of TS and non-TS flows in the TSN CM. The process starts when a new flow request arrives at the TSN CM and ends with that request being accepted or rejected. Details of the two main components of the process, the TS-FSP and the NDT, are also provided.

Provisioning TS and non-TS flows

The general algorithm running in the TSN CM for flow provisioning is presented in Fig. 2. The algorithm starts when a flow request arrives specifying the characteristics of the flow, including the end-points, class of service (e.g., TS, QoS, and BE) KPI requirements, if any, traffic profile including periodicity in the case of a TS flow, and others (step 1 in Fig. 2). The algorithm follows a different procedure for TS and non-TS flows (2). In the case of TS flows (3), the shortest path is computed on a subgraph that includes the end-points of the flow and nodes with TSN capabilities. Next, the TS-FSP module finds a scheduling plan for the new TS flow, and changes in the scheduling of already deployed TS flows, so as to meet the requirements. If no scheduling plan is found, conflicting links and disconnected partitions not including the end-points of the requested flow are removed from the subgraph. If the resulting subgraph is disconnected, no resources are available for the new TS flow request, which is rejected (4). If a scheduling plan is found, the NDT is called to estimate the performance of the non-TS flows already being served as if the TS flow were setup (5). This is a crucial step, as the new TS flow will be assigned resources to detriment of non-TS flows, which will impact their KPIs. In case the performance of QoS committed flows can be guaranteed, the new request is accepted (6). Otherwise, a procedure that excludes the conflicting link, similar to the one introduced above is followed (7) until a solution is found or the request is finally rejected (8). Note that non-TS flows provisioning follow a similar procedure except for the scheduling plan.

TS Flow Scheduling Planner (TS-FSP)

TS-FSP is executed for a TS flow request to be served on a computed path with the objective of reserving resources along that path to support the flow. Changes in the scheduling of already deployed TS flows might be needed, so TS-FSP needs to determine the new resource allocation for those TS flows which resource allocation changes. The resources to be allocated for the TS flows are a set of time windows, with duration specific for each TS flow, on every hop along the defined path. The resource allocation repeats with a given periodicity, which is also specific to the TS flow. In the case that the required resources cannot be reserved for the TS flow, the flow request is blocked. The transmission on any link e in the network is organized in terms of a SF, which consists of a set of time slots in the range $[1..t_{max}]$, each of duration τ_e , where each time slot t can be allocated to only one TS flow. To that end, a resource allocation *window* (T_f) with a number of contiguous time slots for each TS flow f is

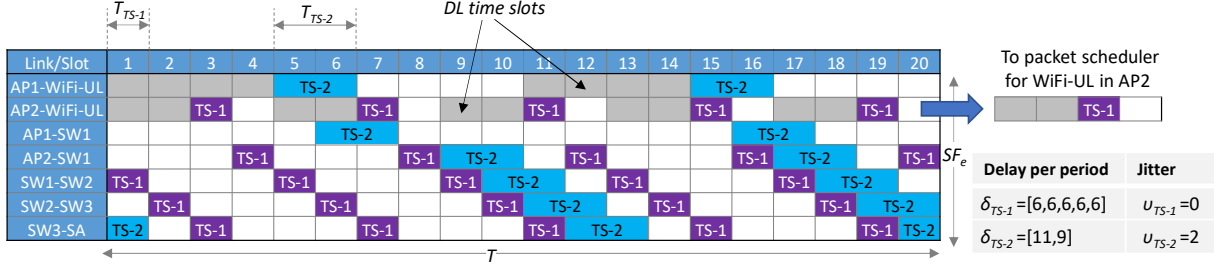


Fig. 3: Illustrative example of NSF' after TS-2 flow request

computed, whose aggregated capacity considers the specifics of flow f . Formally, TS-FSP can be stated as: Given:

- The network topology $G(E)$, modelled as a set of *directed* links E . Each link is characterized by: *i*) the speed of the interfaces B_e . In the case that an interface can offer different speeds (e.g., it is common that wireless interfaces can adapt their modulation format as a function of the quality of the signal of the receiver), the lowest speed is considered. This assumption guarantees the performance of the TS flows even under the worst-case scenario; *ii*) its transmission delay d_e .
- The duration T of the SF. We assume a fixed duration that limits the longer periodicity of the TS flows that can be served. In every link e , a superframe SF_e , in the form of an ordered list, is defined. Note that the duration of each time slot τ_e is defined by speed B_e . In addition, each link can be *full-duplex* or *half-duplex*, where the former links have time slots available during the whole SF_e duration, whereas the latter links have time slots available only during part of the SF_e duration.
- The set of TS flows F already deployed in the network. Each TS flow f is being served through a path defined by a set of links E_f . In addition, the TS flow f has a window T_f of time slots reserved in each link e , which repeats during the SF with periodicity P_f . Finally, the maximum delay that f can support is defined by δ_f and the maximum jitter is defined by v_f . The delay is computed for every period and the jitter is computed as the difference between the maximum and minimum delay in the different periods.
- The current scheduling plan of the network $NSF = \{SF_e, \forall e \in E\}$. Every SF_e defines the allocations of slots to flows, i.e., $SF_e = [s_{ft}]$, where every s_{ft} identifies the TS flow f to which slot t is allocated to; 0 otherwise.
- A new TS flow request $r = (E_r, T_r, P_r, \delta_r, v_r)$. The new request r can be served iff time slots can be reserved along path E_r , satisfying the size of the allocation window T_r , and the periodicity P_r , so that delay and jitter constraints are met. Changes in existing TS flows can be made provided also that their constraints are met.

Objective: To minimize total jitter and the number of TS flows that change their resource allocation, as a way to minimize jitter transients every time a new TS flow is established.

If it is feasible to serve the TS flow request, the new scheduling plan for the network $NSF' = \{SF'_e\}$ is returned.

Fig. 3 illustrates NSF' for a simplified scenario based on Fig. 1b, where only the relevant links for the TS-1 (existing) and TS-2 (request) flows are depicted, no transmission delay is considered, and all the interfaces have the same speed. Time-related values are expressed in time units (tu). TS-1 is defined as: $\{E_{TS-1} = [AP2\text{-WiFi-UL}, AP2\text{-SW1}, SW1\text{-SW2}, SW2\text{-SW3}, SW3\text{-SA}], T_{TS-1} = 1, P_{TS-1} = 4, \delta_{TS-1} = 6, v_{TS-1} = 2\}$, and TS-2 is defined as: $\{E_{TS-2} = [AP1\text{-WiFi-UL}, AP1\text{-SW1}, SW1\text{-SW2}, SW2\text{-SW3}, SW3\text{-SA}], T_{TS-2} = 2, P_{TS-2} = 10, \delta_{TS-2} = [11, 9], v_{TS-2} = 2\}$, and the duration of NSF is $T = 20$ tu. Since links AP1-WiFi-UL and AP2-WiFi-UL are *half-duplex* (uplink, UL), some of the time slots are not available because they are reserved for the downlink (DL) direction. Under such NSF', the delay of flow TS-1 is 6 for every period, so its jitter is 0. We assume that the first bit of the TS flow is available at the start of the SF and after every period. Therefore, the delay for every period of a flow can be easily computed by subtracting the starting time of the window of the last allocation in the path to the time slot where the data is available. In the case of flow TS-2, the delay is 11 and 9 for the first and the second periods, respectively, which translates into a jitter of 2. In consequence, the TS flow request is accepted.

Network Digital Twin (NDT)

Estimation of KPIs of requested and already deployed non-TS flows is based on emulating a *partition* of the real network scenario defined by the path though out the requested flow will be deployed. Emulation is based on three components, i.e., *generators, queues, links, and sinks*, that can reproduce the expected traffic, as well as the real network devices and links with high accuracy and fine granularity (see [3]). Specifically, *generators* produce synthetic flow traffic at two different levels: *i*) at macroscopic level, the *scale* (traffic intensity) is generated according to a periodical profile (e.g., daily) and with coarse resolution (e.g., one value per hour); and *ii*) at microscopic level and for each scale value, a fine resolution calculation (e.g., at μs scale) of a short period (1 to 10 seconds) is conducted with traffic flows generated following probability distributions characterizing inter-arrival burst and packet time, and burst and packet size [3]. In addition, *flow queue models* are based on the time-dependent ones in [4] and used to emulate TS-capable interfaces. Queue service rate is pre-empted at the beginning of a time period for a duration that depends on the amount and interval length of the existing and/or requested TS

flows on that interface, while the remaining time in the period is available for non-TS flows according to their priority. Finally, *links* emulate the transmission delay in network links and *sinks* are end-points of flows for KPIs evaluation purposes.

The propagation of the generated traffic for the flows through the defined queuing system results in metrics, such as queued traffic, that are afterwards used to compose *flow* KPIs, such as e2e delay. Without loss of generality, the NDT produces two types of KPIs estimation: *i*) *e2e*, which are provided only for non-TS requests; and *ii*) *variations* (Δ), computed for already deployed flows as the KPI increment or decrement for each flow if the new request would be finally deployed, in the network partition defined by the path of the request. Finally, the NDT includes two databases (DB) that are conveniently updated during network operation: *i*) the network DB stores the current status of the network topology (active nodes and links), as well as the details of the already deployed flows; and *ii*) a monitoring DB with real e2e performance measurements (delay, throughput, and others) of the existing flows. It is worth mentioning that the availability of fine grain, segmented measurements is of paramount importance for the accuracy of KPI estimations produced by the NDT.

Fig. 4 shows the details of the performance evaluation process that is executed by the NDT during the provisioning process of flow request r on path E_r (see Fig. 2). The first step consists in retrieving the set of links

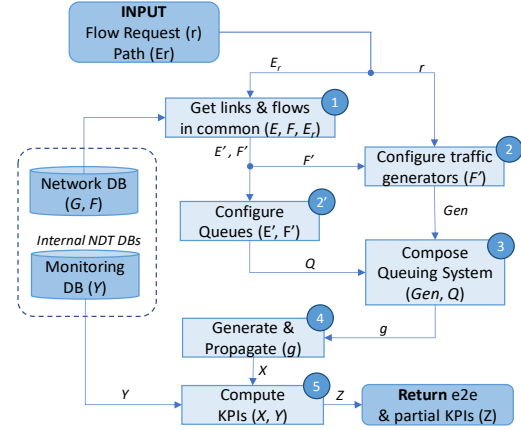


Fig. 4: NDT main procedure

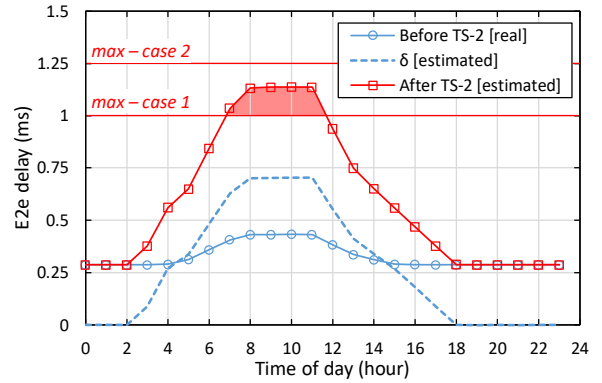


Fig. 5: KPI evaluation of QoS-1 flow during TS-2 request

(E') and existing flows (F') from the internal network DB that share links and interfaces with request r (step 1 in Fig. 4). These subsets feed two different processes running in parallel: on the one hand, traffic generators are built and configured to generate traffic according to r and F' specifications (2), and on the other hand, queues are configured to emulate the network subset E' (2'). The outputs of both processes are used by a queuing system composer (3) that concatenates the queues and bonds the generators to the beginning of each flow and/or segment. The propagation of the generated traffic through the composed queuing system (4) creates metrics X that, jointly with the available monitoring data Y of the existing flows, are used to estimate the KPIs (5) that are finally returned.

For illustrative purposes, let us evaluate the scenario in Fig. 1b triggered by the request TS-2; we assume the duration of the SF $T=10$ ms and $T_r=1$ ms. Since the request is a TS flow, the NDT will be used to estimate the performance of the non-TS flows already deployed. In particular, let us examine the performance of flow QoS-1, which follows a daily traffic profile; two cases are considered for the maximum delay: $\delta_{QoS-1} = 1$ ms and $\delta_{QoS-1} = 1.25$ ms. Fig. 5 shows the e2e delay monitored for QoS-1 currently available in the monitoring DB before TS-2 is established. Note that the delay requirement (in red) is achieved at any time along the day. Now, the NDT solves the queueing system representing the network partition defined by the path $\langle AP1_WiFi_UL, AP1-SW1, SW1-SW2, SW2-SW3, SW3-ServerA \rangle$. Note that the generators of flows QoS-1 and TS-1 are bonded to SW1, whereas the one of BE-1 is connected to SW2. The obtained estimated variation Δ_{QoS-1} is also plotted in Fig. 5, as well as the estimated e2e delay after TS-2 provisioning, which is the result of adding the estimated Δ_{QoS-1} to the real monitored e2e delay without TS-2. We observe that the delay requirement of QoS-1 will be violated at some time during the daily period for case 1, which will prevent TS-2 flow request to be accepted. However, in case 2 with a more relaxed maximum delay for flow QoS-1, no violation is observed and TS-2 flow request can be accepted.

4. CONCLUSIONS

A solution for the provisioning of TS and non-TS flows has been outlined in this paper. A general algorithm for flow provisioning running in the control plane was presented that involves: *i*) a TS FSP to plan TS flow time windows across a defined path; and *ii*) an NDT to evaluate the KPIs on non-TS flows. An illustrative scenario was used to showcase the operation of the TS FSP and the NDT upon the request for a new TS flow provisioning.

REFERENCES

- [1] IEEE Time-Sensitive Networking Task Group. [On-line] <https://1.ieee802.org/tsn/>.
- [2] Y. Lu *et al.*, "An Intelligent Deterministic Scheduling Method for Ultra-Low Latency Communication in Edge Enabled Industrial IoT," IEEE Trans. on Industrial Informatics, vol. 19, pp. 1756-1767, 2023.
- [3] A. Bernal *et al.*, "Near real-time estimation of end-to-end performance in converged fixed-mobile networks," Comp. Comms., 2020.
- [4] L. Velasco and M. Ruiz, "Supporting time-sensitive and best-effort traffic on a common metro infrastructure," Comm. Letters, 2020.