Provisioning of Time-Sensitive and non-Time-Sensitive Flows: from Control to Data Plane

L. Velasco*, G. Graziadei, Y. El Kaisi, J. Villares, O. Muñoz, J. Vidal, and M. Ruiz
Universitat Politecnica de Catalunya (UPC), Barcelona, Spain
*luis.velasco@upc.edu

***Abstract*—Time-Sensitive Networking (TSN) standards provide scheduling and traffic shaping mechanisms to ensure the coexistence of Time-Sensitive (TS) and non-TS traffic classes on the same network infrastructure. Nonetheless, much effort is still needed on the operation of such TSN capable network infrastructure to ensure that the required performance of the different flows, defined in terms of key performance indicators, can be met once the flows are deployed in the network. In this paper, we focus on such aspects and propose a solution involving not only packet schedulers in the data plane, but also network-wide scheduling for TS flows, as well as performance estimation for non-TS flows.**

Keywords—Time-Sensitive Networking; Network Operation; Time-aware scheduling; Network Digital Twin.

I. 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 while maximizing resource utilization.

The rest of the paper is organized as follows. Section II presents the scenario targeted in this paper, which includes the control and the data planes. In the data plane, we consider TSN capable network devices supporting TS and non-TS traffic flows, as well as non-TSN capable devices supporting non-TS traffic only. The control plane is based on the Software-Defined Networking (SDN) paradigm and is designed to provide end-to-end connectivity. Two main systems in the control plane are the focus of Section III, the *TS Flow Scheduler Planner* (TS FSP) to plan TS flow time windows across a defined path and the NDT to evaluate the KPIs on non-TS flows. Both, TS FSP and NDT work under the assumption of the worst-case scenario, so as the performance of the flows is guaranteed. However, the data plane cannot be operated under the worst-case scenario as that would result in a poor resource utilization. In consequence, Section IV focuses on the mechanisms to be implemented in the local packet schedulers managing the network interfaces to maximize resource utilization to improve the performance of non-TS flows. Specifically, the design of a WiFi SF that guarantees that TS traffic is always served on time while using efficiently the remaining time-varying radio resources to enhance the performance of non-TS traffic flows. The proposed design takes advantage of the wireless channel variability by way of link adaptation and wireless scheduling. In the end, Section V draws some conclusions and highlights other important topics that need to be considered.

II. TSN Architecture Supporting TS and non-TS Traffic Mix

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

ISBN 978-3-903176-63-8 © 2024 IFIP

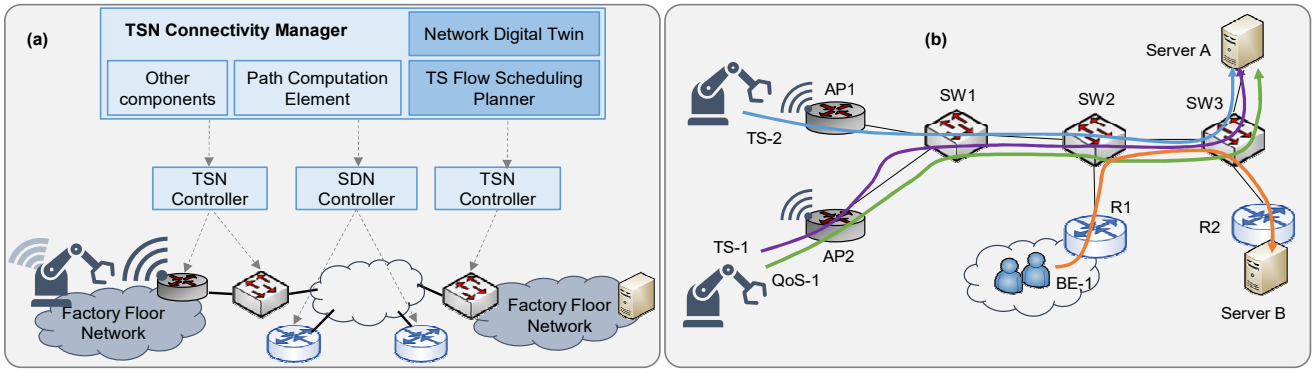

Fig. 1: Heterogenous TSN network and overview of the proposed architecture (a). Illustrative scenario (b).

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 defined for each flow, 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.

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.

III. Control Plane Algorithms for 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.

A. 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,

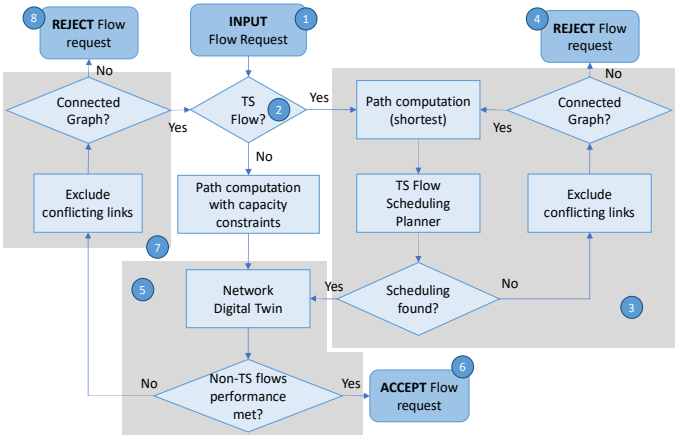

Fig. 2: Provisioning algorithm for TS and non-TS Flows

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.

B. 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

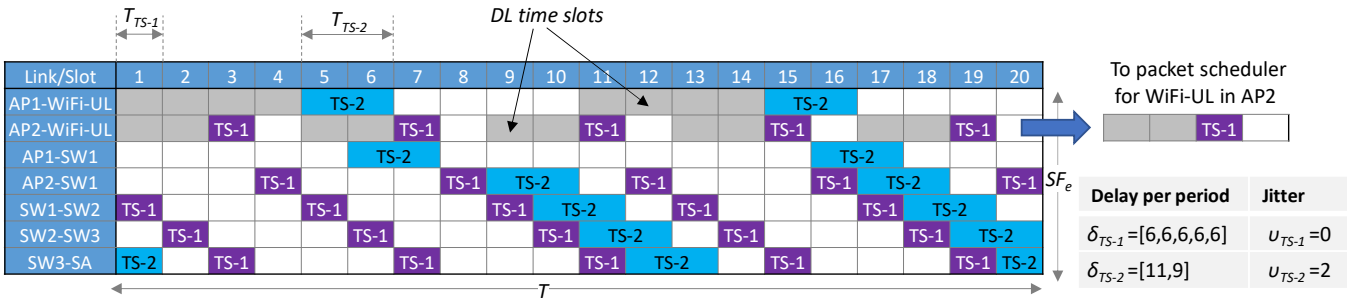

Fig. 3: Illustrative example of NSF' after TS-2 flow request

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

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 or, alternatively, the duration of each time slot τ_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 given or 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 υ_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, δ_r, υ_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 the scheduling of the existing TS flows can be made provided that their constraints are also 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.

In the case that 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-WiFi-UL, AP2-SW1, SW-SW2, SW2-SW3, SW3-SA], T_{TS-1} = 1, P_{TS-1} = 4, δ_{TS-1}=6, υ_{TS-1}=2$\}$, and TS-2 is defined as: $\{E_{TS-2}$=[AP1-WiFi-UL, AP1-SW1, SW-SW2, SW2-SW3, SW3-SA], T_{TS-2} = 2, P_{TS-2} = 10, δ_{TS-2}=12, υ_{TS-2}=2$\}$, and the duration of NSF is T=20tu. 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.

The TSN CM uses NSF' to provide the worst-case plan to the scheduling algorithm controlling packet forwarding in the interfaces of the network devices. Before that, every SF'_e needs to be processed, since they might include allocation blocks that repeat periodically. For instance, NSF' in Fig. 3 includes AP1-WiFi and AP2-WiFi interfaces with two and five repetitions, respectively, which are removed to produce the final SF_e version to be sent to the packet scheduler.

C. 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 microscopic 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 used as 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 (*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 a set of 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.

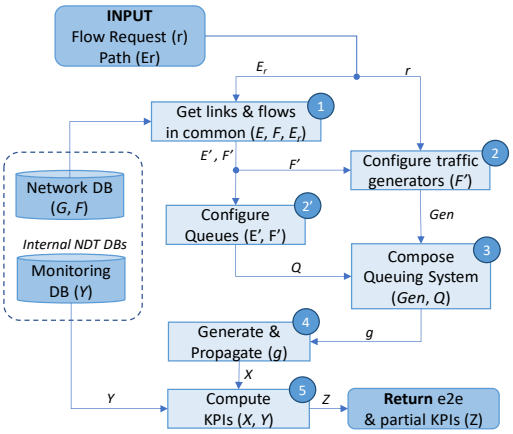

Fig. 4: NDT main procedure

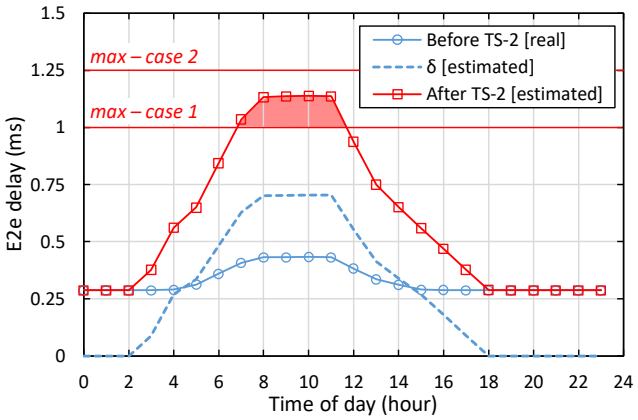

Fig. 5: KPI evaluation of QoS-1 flow during TS-2 request

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=10ms and T_r=1ms. 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: $\delta_{QoS\text{-}1}$ = 1ms and $\delta_{QoS\text{-}1}$ = 1.25ms. 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 <AP1_WiFi_UL, AP1-SW1, SW1-SW2, SW2-SW3, SW3-ServerA>. 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 $\Delta_{QoS\text{-}1}$ 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 $\Delta_{QoS\text{-}1}$ 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.

IV. Maximizing the Performance of TS and non-TS Traffic in the Data Plane

The provisioning process for TS and non-TS flows in the TSN CM described in the previous section considers, for each link of the network, their lowest possible speed. Based on that, the TS FSP produces as an output, a structure for the SF containing a set of pre-empted time slots, i.e., temporal windows, for TS traffic. Allocating TS flows within the pre-empted slots will ensure the feasibility of their successive transmissions across the network. In this section, we analyze how the dynamic scheduler at each network interface decides how to assign the available free slots among the different flows. In particular, we focus on wireless segments (see an illustrative example in Fig. 6). The more straightforward allocation of data plane packets compatible with the described output of the TS FSP, as well as with the standard 802.1Qbv, is using the pre-empted slots exclusively for their corresponding TS flows and using the remaining resources for non-TS flows. Nevertheless, this strategy may be inefficient because the link speed cannot be assumed to be constant over time for wireless links.

A. Packet scheduling in wireless segments

In a wireless link, e.g., based on Wi-Fi6, the minimum allocation unit is a resource block (RB) that has a fixed time and frequency duration. The number of bits that can be transmitted in an RB thus vary over time since such a number depends on the selected modulation and coding scheme (MCS), which in turn depends on the state of the propagation channel and the target reliability. Accordingly, the number of RBs that are required to transmit a given amount of TS traffic is variable and hence, a more efficient data plane can be achieved by designing the WiFi SF to take advantage of this variability.

Let us assume that the radio resource manager (RRM) at the WiFi AP monitors frequently the quality of the wireless channel with the tenant nodes, including attenuation, interference levels, and the highest MCS ensuring quality and reliability over the wireless channel. By adjusting the transmission bitrate of each flow according to this MCS (*link adaptation*), we will have additional free resource blocks at each SF, w.r.t. those considered by the TS FSP using the worst-case MCS. Such extra resources can then be used to improve the performance of non-TS flows.

Following the above approach, the number of available resource blocks for both DL and UL, T_{DL}^{free} and T_{UL}^{free}, changes from SF to SF due to the randomness of the transmission bitrate for the different TS flows. At the beginning of every SF, a dynamic scheduler at each AP decides how to assign the available free slots among the different flows. In addition to the information of the state of the channel e.g., MCS, (Channel State Information -CSI), the states of the queues with the number of packets waiting in the queues and their associated priorities (Queue State Information -QSI), and timeliness requirement and time-stamps of packets in queues (Packet Information -PI), can support better-informed scheduling decisions.

Based on the available information, more or less sophisticated approaches can be applied for the scheduling of data packets. Round-robin (RR) is the simplest approach, where the different flows are served following a predefined order and therefore, it requires little to no information and does not consider QoS guarantees. Instead, other schedulers focus on maximizing a given figure of merit, such as throughput, delay, etc. For example, the Largest-Weighted-Delay-First (LWDF) scheduler [5] prioritizes flows according to their value for the following utility function:

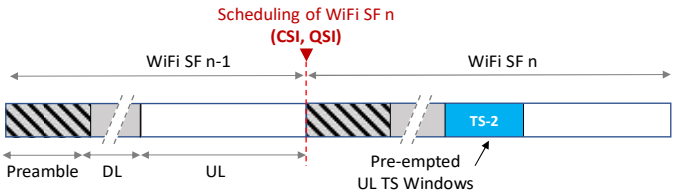

Fig. 6: Example of a WiFi SF in AP1 UL with pre-empted slots.

$$U_n = R_n^{WiFi} Q_n, \quad (1)$$

where Q_n stands for the state of the n-th flow buffer (queue) at the beginning of the SF (number of bits in the queue) and R_n^{WiFi} is the instantaneous bitrate of flow n during this SF. The flow maximizing the utility U_n is scheduled first and will be transmitted using as many slots as required to empty its buffer or until it has consumed all the available slots (T_{DL}^{free} or T_{UL}^{free}). If some slots are still free, the second flow with the highest utility is scheduled to transmit following the same procedure, and so on. To minimize latency, the available slots are assigned sequentially in the SF. Other scheduling approaches are also possible. For example, using utility functions that depend on the long-term throughput of each non-TS flow ensures that network resources are fairly shared in the long term and the QoS targets of each flow are satisfied [6]. The same scheduling algorithms can be adopted for both DL and UL, with some adjustments for the UL. Specifically, each UL flow has to include an interframe spacing and a short preamble of duration $T_{p,UL}$. This overhead can be partially reduced if the same station generates several UL flows. Also, if the scheduler requires QSI, as for the LWDF scheduler, all stations having active UL flows have to report their buffer size Q_n by the beginning of the SF.

B. Illustrative Simulation Results

To evaluate the impact of smart radio management of the windows designed by the TS FSP, we have simulated a typical WiFi scenario with multiple robots connected to the same AP. Every robot generates several TS flows consisting of a periodic stream of fixed-length packets. For simplicity, we considered all the flows share the same period P_f and thus, the SF duration is $T = P_f$. Following the NSF design in the previous section, windows of T_f slots are reserved in the WiFi SF to serve the TS flows (*pre-empted slots*) assuming the worst-case MCS. However, to accommodate as much non-TS traffic as possible, we use link adaptation and select the best MCS possible according to the channel conditions, and consequently, more RBs will be available in the wireless link to allocate non-TS traffic flows.

Simulations have been carried out for two competing designs of the wireless link SF: *i*) *Baseline* design, where pre-empted slots are used exclusively by TS traffic, while the non-TS traffic is allocated on other non-pre-empted RBs. An RR scheduler is

Table 1: Simulation parameters list.

Simulation Parameter		Value
Superframe duration		10 ms
Slot duration (OFDM symbol)		4 µs
Bandwidth		20 MHz
Number of data subcarriers		48
MCS list (802.11a/g)		0, ..., 6
SF time for DL and UL		50%
Superframe preamble [7]		24 slots
UL overhead ($T_{p,UL}$) [7]		2 slots
Packet Loss Rate		10^{-4}
Average Received SNR		25.3 dB
Delay Spread [7]		50 ns
Power Delay Profile		Exponential
Fading distribution		Rayleigh
Coherence Time [7]		30 ms
Doppler Spectrum		Block fading
TS traffic flows	Packet length (B)	30 bytes
	Periodicity	10 ms
Non-TS traffic flows (with QoS)	Packet length (B)	90 bytes
	Periodicity	5 ms

adopted to select the non-TS flows that are transmitted in every new SF; and *ii*) *Improved* design, where part of the non-TS traffic is also allocated in empty pre-empted RBs. Since the MCS of TSN flows changes over time, the number of free RBs within the TS windows also varies. In consequence, following the LWDF approach, non-TS flows are scheduled using a dynamic scheduler that prioritizes those flows that can be transmitted with a faster MCS and have more packets in the queue waiting to be transmitted.

Fig. 7 presents the results of the evaluation. As a convenient KPI for non-TS QoS traffic, we have selected the 10% outage delay, i.e., the value of the packet delay with a Cumulative Density Function (CDF) value of 90%. The plots show the 10% outage delay of non-TS UL flows as a function of their aggregated traffic throughput (served load), where the delay is normalized w.r.t. the SF duration. Simulations are carried out considering that 50%, 75%, and 100% of the SF duration corresponds to pre-empted slots for TS traffic. The simulation scenario is detailed in Table 1. We observe that the proposed design outperforms the baseline design in terms of throughput without increasing the 10% outage delay. As a reference, when TS windows occupy 50% of the SF, the throughput of non-TS traffic improves from 8 Mb/s up to 13 Mb/s with a 10% outage delay of about 1.5 SFs. Note that, this delay cannot be reduced unless we adopt pre-emptive scheduling, as with TS traffic. This is because packets of non-TS uplink flows arriving during a given SF are not scheduled until the second half of the next SF, since the first half is dedicated to DL flows (see Fig. 6). We observe that the throughput improvement is higher when the TS windows span over a greater portion of the SF, for example, 75%. In that case, the UL throughput increases from 4 Mb/s to 11 Mb/s. Remarkably, 10 Mb/s can still be supported using the proposed SF design despite the whole SF being initially allocated for TS traffic.

V. Conclusions

A complete solution for the provisioning of TS and non-TS flows involving elements in the control and data planes have been outlined in this paper. A general algorithm for flow

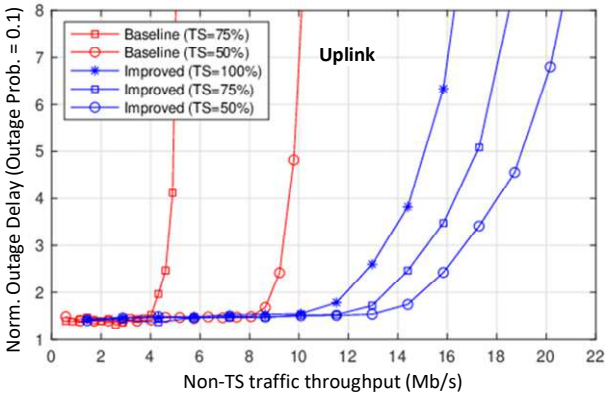

Fig. 7: 10% Outage delay vs. throughput of non-TS traffic.

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 network scenario was used to showcase the operation of the TS FSP and the NDT upon the request for a new TS flow provisioning. The worst-case scenario is assumed during the provisioning phase to ensure the performance of the flows. In the data plane (specifically in wireless), local packet schedulers operating the network interfaces maximize resource utilization by allocating resources initially reserved for TS flows when better MCS can be assigned to TS flows, thus improving the performance of non-TS flows.

It is worth mentioning other important topics not covered in this paper, in particular the multidomain scenario, where e2e TS flows need to be established across a number of heterogeneous network domains that might or might not have TSN capabilities. In such scenario, the provisioning process becomes much more challenging, since several PCEs, NDTs and TS FSPs might be involved, which requires considering some sort of coordination.

Acknowledgment

This project has received funding from the NextGeneration UNICO5G TIMING (TSI-063000-2021-145), the SNS JU though the European Union's Horizon Europe under G.A. No. 101095890 (PREDICT-6G), and the AEI IBON (PID2020-114135RB-I00) projects, and the ICREA institution.

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., vol. 150, 2020.

[4] L. Velasco and M. Ruiz, "Supporting time-sensitive and best-effort traffic on a common metro infrastructure," Comm. Letters, vol. 24, 2020.

[5] K. Ramanan *et al.*, "Largest weighted delay first scheduling: large deviations and optimality," Annals Applied Prob., vol. 11, pp. 1-48, 2001.

[6] E. Calvo *et al.*, "Downlink coordinated radio resource management in cellular networks with partial CSI," IEEE Trans. on Signal Processing, vol. 60, pp. 1420-1431, 2012.

[7] O. Seijo *et al.*, "SHARP: A novel hybrid architecture for industrial wireless sensor and actuator networks," in proc. IEEE WFCS, 2018.