Leveraging DRL for Traffic Prioritization in 5G and Beyond TSN-based Transport Networks

Jonathan Prados-Garzon[(1),(2)], Lorena Chinchilla-Romero[(1)], Pablo Muñoz[(1),(2)], Pablo Ameigeiras[(1),(2)], and Juan J. Ramos-Munoz[(1),(2)]
jpg@ugr.es, lorenachinchilla@ugr.es, pabloml@ugr.es, pameigeiras@ugr.es, jjramos@ugr.es
[(1)]Department of Signal Theory, Telematics and Communications (DTSTC), University of Granada, Spain.
[(2)]Research Center on Information and Communication Technologies, University of Granada, Spain.

***Abstract*—Time-Sensitive Networking (TSN) is expected to become a key layer 2 technology for 5G and Beyond (5GB) transport networks (TN) as it allows for services with stringent and deterministic quality-of-service constraints and their coexistence with non-performance-sensitive traffic. Autonomous solutions for configuring TSN-based TNs are essential to ensure the deterministic QoS requisites of the 5GB streams while facilitating the zero-touch management of the network and reducing the operational costs. However, due to the configuration flexibility offered by TSN networks, using exact optimization methods to develop such solutions usually results in algorithms with high computational complexity. In this work, we propose and evaluate an initial design of a Reinforcement Learning (RL)-based solution for the long-term configuration of asynchronous TSN-based 5GB TNs. We successfully validated the proper operation of the proposal for an industrial private 5G scenario.**

I. Introduction

Time-Sensitive Networking (TSN) is a set of standards specified by IEEE 802 as amendments to IEEE 802.1Q to ensure the stream's layer 2 (L2) deterministic transport. In other words, TSN can establish an L2 multi-hop path over the network for a given stream or flow with deterministic Quality-of-Service (QoS) guarantees in terms of latency, jitter, packet loss, and reliability. Furthermore, TSN enables the coexistence of both critical and non-performance-sensitive services. Due to these capabilities, TSN is recognized as a key technology to cost-effectively realize the 5G and Beyond (5GB) Transport Networks (TNs) [1], [2]. TNs refer to the underlying networks that provide connectivity among the functional entities of mobile networks, and their specification is out of the 3GPP scope [1].

The TSN capability for supporting the transport of flows with deterministic QoS mainly relies on sophisticated schedulers that handle the transmission of the frames at every TSN bridge's egress port. Here, we focus on asynchronous TSN-based TNs, i.e., the TSN bridges do not need a common and precise time reference to be synchronized. More precisely, we consider the TSN asynchronous traffic shaper (ATS) is used at every TSN bridge's output port. The ATS is a TSN scheduler [3] based on the Urgency-Based Shaper (UBS) proposed by Specht and Samii [4]. It includes the novel interleaved shaping concept that allows for a cost-effective per-flow traffic regulation without increasing the worst-case delay [5].

Configuring TSN networks is a complex task due to their intricate link schedulers and the configuration flexibility they offer. In particular, the configuration of asynchronous TSN networks has a combinatorial configuration complexity [6]–[8]. In this vein, Machine Learning techniques are appealing approaches to be explored for this task [1], [9].

In this work, we propose a novel multi-agent Reinforcement Learning (RL)-based solution for traffic prioritization in asynchronous TSN-based 5G TNs. The solution architecture has been devised for generality, i.e., applicable in a broad spectrum of scenarios. Moreover, using a multi-agent approach drastically reduces the required amount of data for training compared to using a monolithic agent. In [9], an online RL-based flow allocation solution is proposed, i.e., the flow configuration is computed right after it is created. In contrast to [9], here we propose an offline RL-based solution to compute long-term configurations for asynchronous TSN-based 5GB TNs. In a nutshell, the proposed solution in this work maps the 5G network slices onto the eight IEEE 802.1Q Traffic Classes (TCs), distributes the end-to-end (E2E) TN delay budgets of the TCs among the TN hops for every predefined path, and computes the prioritization of the TCs. Besides the solution architecture, a possible implementation to find feasible asynchronous TSN-based 5GB TNs configurations is described. A configuration is feasible if it meets the E2E TN delay budgets for all the 5GB slices. Nonetheless, the proposed solution design is also valid when an optimization goal is considered for the traffic prioritization (e.g., maximizing the TN operator profit or minimizing the 5GB streams rejection probability). Last, we carry out a simulation-based proof-of-concept (PoC) to validate the proposed solution. The 5GB slices' traffic characteristics in the PoC have been derived from the primary use cases for industrial private 5G networks.

II. System Model and Problem Statement

Let us assume a multi-tenant private 5G network consisting of a 5G System (5GS) whose components are interconnected through an L2 TSN network, as depicted in Fig. 1. There are predefined paths to interconnect every source-destination pair. There are a set of 5G network slices sharing the TSN-based TN. Each slice is mapped onto a VLAN and one of the eight IEEE 802.1Q traffic classes (TCs) that are encoded in the VLAN ID and the PCP fields of the IEEE 802.1Q header respectively. Observe that multiple slices might belong to the same TC. The aggregated data rate and the aggregated burst size of each slice are known in advance as it is typical for industrial 5G use cases (e.g., closed-loop control). Each slice has a deterministic E2E packet delay budget to be fulfilled. A percentage of this E2E delay budget is reserved for the TN

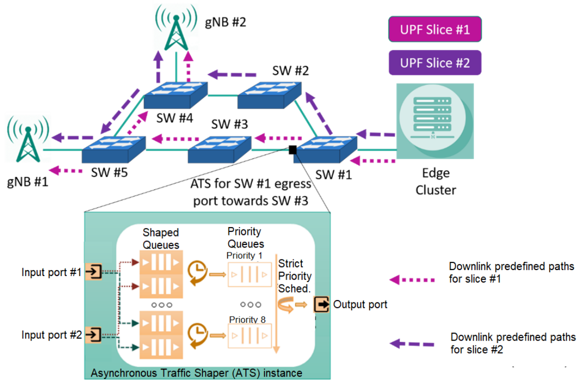

Fig. 1. System model.

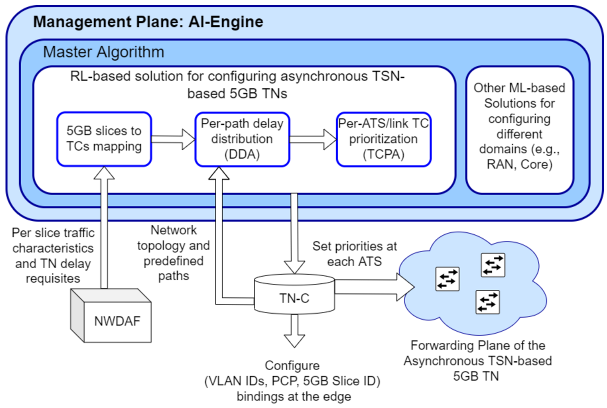

Fig. 2. High-level architecture of the RL-based traffic prioritization solution for the configuration of asynchronous TSN-based 5GB TNs.

domain (e.g., 10% of the E2E delay budget), referred to as the E2E TN delay budget.

Each TSN bridge's egress port includes an ATS instance encompassing two queuing stages: i) the interleaved shaping to cost-effectively realize a per-flow traffic regulation, and ii) traffic prioritization to transmit the frames according to a strict priority selection scheme. In this work, we consider that the priority of a frame is given by its TC. We assume the interleaved shaping stage enables the implementation of eight priority levels as considered in TSN standards by default. We refer the interested reader to [1], [3]–[5] for a more detailed explanation of the ATS operation and related concepts.

According to TSN standards, the worst-case delay $D_{f,\mathcal{R}}$ of a flow f traversing the path $\mathcal{R}$ can be computed as [3]:

$$D_{f,\mathcal{R}} \leq \sum_{e\in\mathcal{R}} D_{f,e} + \frac{l_f}{C_e} = \sum_{e\in\mathcal{R}} \frac{\sum_{z=1}^{p_f^{(e)}} \hat{b}_z + \bigvee_{z=p+1}^{P_e} l_z}{C_e - \sum_{z=1}^{p_f^{(e)}-1} \hat{r}_z} + \frac{l_f}{C_e} \quad (1)$$

Where $\hat{b}_z$, l_z, and $\hat{r}_z$ are the aggregated burst size, maximum frame size, and aggregated data rate at priority level z, respectively; C_e denotes the capacity of the ATS/link e; $p_f^{(e)}$ is the priority assigned to the flow f at ATS/link e; and l_f denotes the maximum frame size of the flow.

Under the premises stated above, the problem covered here consists in finding a feasible traffic prioritization of the 5G network slices (slice-to-priority level assignment) in the asynchronous TSN-based TN so that the E2E TN delay budgets of every slice are met.

III. RL-Based Solution for TSN-Based 5GB TNs

This section describes the proposed solution. First, we give an overview of the solution architecture. Then, we provide some implementation details to solve the problem stated in the previous section.

A. Solution Architecture

Figure 2 shows a sketch of the solution's design, which comprises three components and its context. The solution gets the required information, telemetry, and data analytics, used as inputs, from both the Network Data Analytics Function (NWDAF) of the 5GS control plane (CP) (e.g., slices traffic characteristics and requirements) and TN Controller (TN-C) of the TSN-based TN CP (e.g., network topology and links status and capacities). The TN-C is in charge of populating the solution's output. On the one hand, the TN-C shall communicate the slices-to-TCs assignments information to the involved entities responsible for this translation. On the other hand, it shall apply the computed TCs prioritization at every ATS/link. Below are listed the solution's components and their functionalities:

1) 5GB slices to TCs mapping: This block is in charge of assigning an IEEE 802.1Q to each slice. It clusters the 5G slices into eight groups (when the number of slices is greater than eight). In this way, the solution becomes independent of the number of slices to be accommodated in the TN. The E2E TN delay budget for a given TC is set to the most stringent delay requirement of all the slices mapped onto that TC.

2) Delay Distribution Agent (DDA): This component is responsible for distributing the E2E TN delay budget among the TN hops. This agent is invoked for every TC and TN source-destination path. It could also distribute the delay per TC for further configuration flexibility, though for simplicity, here we do it per path. To that end, it relies on the TC traffic characteristics (aggregated data rate, maximum frame size, and aggregated burstiness) at each hop and the nominal capacities of each link.

3) Per-ATS TCs Prioritization Agent (TCPA): This agent oversees performing the TCs prioritization at each TN device output port's frame scheduler. Therefore, either there is a dedicated agent instance or the same agent must be invoked one time per TN device output port.

The solution's components instances are run sequentially and in the same order as the list above. The output of the 5GB slices to TCs mapping is required to compute the per-ATS/link, and per-TC aggregated traffic characteristics and requirements that are used as input by DDA and TCPA instances. In the same way, the delay distribution issued by the DDA is required by TCPA instances.

B. Solution Components Implementation

Next, we provide details on the implementation details of each solution's component.

1) 5GB slices to TCs mapping: For this component, we use the k-means algorithm to cluster the 5GB slices into eight groups, each standing for an IEEE 802.1Q TC. The main features considered to characterize each slice and perform the clustering are: i) the E2E TN delay budget, ii) the aggregated data rate, iii) the aggregated burst size, and iv) the maximum frame length of the slice. These features are normalized and weighted to set their importance. The weights are adjusted using a trial-and-error approach. The weights are sampled, and each sample is ranked according to the reward function. The reward function considered here is proportional to the sum of the required priority levels at each hop once the whole TN configuration process finishes. The weights that result in the highest value of the reward function are selected.

2) Delay Distribution Agent (DDA): This block is realized as an RL agent that assigns a percentage of the E2E TN delay budget to a hop for each network path. If there is a conflict between the delay budget assigned to a given hop and TC by the DDA from different paths, then the most stringent delay constraint is considered for that hop and TC.

For each path, the DDA takes the following observations: i) per-TC and per-hop utilization (aggregated data rate of the TC divided by the link capacity), ii) per-TC and per-hop burst size, iii) per-TC and per-hop maximum frame size, and iv) per-TC TN E2E delay budget normalized by the maximum frame transmission time. Based on these observations, the DDA chooses one action from the set $A_{DDA} = \{0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$, where each action stands for the percentage of the E2E TN delay budget assigned to a given hop. Observe that we might use a larger set of actions to increase the granularity of the delay budget distribution. The first action assigns the first hop delay budget and so on for the rest of the hops. Here we assume a maximum network diameter of seven hops and, therefore, each episode comprises seven steps/actions at most. For each action, the agent receives an instant reward of +10 if the TCPA finds a feasible TCs prioritization given the percentage of the E2E TN delay budget allocated to the corresponding hop. Otherwise, the agent is penalized with -10. Also, the agent is rewarded with +50 at the end of the episode if the sum of the percentage of delay budgets assigned to each hop of the path equals 100. That is to encourage the agent to consume the whole delay budget.

3) Per-ATS TCs Prioritization Agent (TCPA): This RL agent performs the TC-to-priority assignment at every ATS/link. Below are the primary parts of the RL model for the TCPA.

Observations: Per-TC characteristics and setup, namely, Per-TC utilization (aggregated data rate of the TC divided by the link capacity), Per-TC maximum burst size, Per-TC maximum frame size, Per-TC hop delay budget divided by the maximum frame transmission time, and Per-TC current assigned priority.

Actions: The set of TCPA's actions considered here is $A_{TCPA} = \{\downarrow_1, \downarrow_2, \downarrow_3, \downarrow_4, \downarrow_5, \downarrow_6, \downarrow_7, \downarrow_8\}$, where the action $\downarrow_\tau$ stands for the agent decreases the priority level of the TC $\tau \in [1, 8]$. At the beginning of each episode, the highest priority level (priority 1) is assigned to all the eight TCs. At each step, the priority level of the respective TC is lowered according to the action issued by the agent.

Reward: The main idea behind the proposed reward is that decreasing the priority level of a given TC only increases its delay. However, it either decreases or does not affect the delay of the other traffic classes. Below is a summary of the reward function: Each action is immediately rewarded with $+N$, where N is the number of delay requirements met due to the action (before the action, they were not met). Each step has a default reward of -0.5 to minimize the required number of steps. If the problem is solved at any time (the delay requirement is fulfilled for all the traffic classes, which is checked using equation (1)), the episode is finished (terminal state), and the agent is rewarded with +100.

Terminal states: Besides the terminal state reached when the problem is solved, each episode has a maximum number of 28 steps. As we are considering eight TCs and eight priority levels, this number of steps is enough to enable the agent to configure any TC prioritization in an episode.

TABLE I
DELAY REQUIREMENTS AND PRIORITIZATION FOR ATS LINK USED IN THE HYPERPARAMETERS STUDY. THE CAPACITY OF THE LINK IS 100 GBPS AND THE UTILIZATION IS 27.45%.

TC	Delay budget	Delay	Prio
#1	1.265 µs	1.094 µs	4
#2	1.2282 µs	1.094 µs	4
#3	1.261 µs	0.764 µs	3
#4	0.80671 µs	0.764 µs	3
#5	0.58371 µs	0.493 µs	2
#6	0.43462 µs	0.36 µs	1
#7	1.4688 µs	1.329 µs	5
#8	2.2174 µs	1.329 µs	5

IV. PROOF OF CONCEPT (POC)

This section includes the description of a simulation-based PoC to test the proper operation of the proposed RL-based solution for traffic prioritization in asynchronous TSN-based TNs. First, we provide the configuration and training. Next, we present the results obtained for an industrial private 5G scenario that support the validity of the proposed solution.

A. Agents Configuration and Training

The different RL agents were developed in Python using Stable Baselines3 in PyTorch. OpenAI Gym was used to develop the different training environments. The training processes and tests were carried out on a server with two processors Intel(R) Xeon(R) CPU E5-2603 v4 @ 1.70GHz and 32 GB of RAM.

First, the hyperparameters (i.e., the parameters that serve to control the learning process) were tuned to improve the efficiency of the training processes and the performance of the resulting agents. For this purpose, we used the grid-search technique. Due to this approach, the agents were trained using a single scenario for each combination of hyperparameters. For instance, the scenario considered for tuning the hyperparameters of the TCPA consisted of an ATS scheduling the frame transmissions in an Ethernet link with a capacity of 100 Gbps and an aggregated utilization 27.45%. The second column in Table I includes the link delay budget considered. The fourth and third columns in Table I include the TC prioritization found by the TCPA when it converges and the resulting frame delay as a consequence of that prioritization, respectively. Figure 3 shows the respective TCPA's mean reward versus the number of steps for a learning rate of 0.001 and different values of the discount factor. As a result of the study described above, the hyperparameters configuration used is included in Table II.

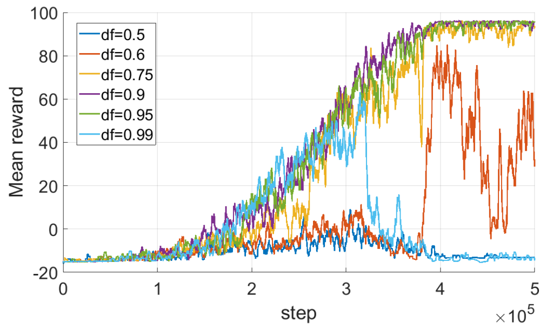

Fig. 3. Mean reward versus the number of steps for a learning rate of 0.001 and different values of the discount factor.

TABLE II
HYPERPARAMETERS SETUP.

DQN Agent hyperparameter	Configuration
RL method	DQN with critic network
Learning rate	0.001
Maximum number of steps per episode	28
Mini-batch size	32
Discount factor	0.9
Experience buffer length	10000
Target update frequency	4
Target update method	Periodic
Critic Network	2 hidden layers with 256 neurons each
Epsilon Max.	1
Epsilon Min.	0.05
Epsilon Fraction	0.5

Next, we trained the agents for generalization, i.e., to make them agnostic to the specific scenario features. To that end, a scenario generator was developed to create scenarios databases with a diversity of features. A database of 100 solvable scenarios was generated using the scenario generator for the agents' training. Around 80 million steps were required to achieve the convergence of the agents using that database.

B. Solution Testing

Figure 4 depicts the scenario considered to test the validity of the proposed RL-based solution for traffic prioritization in asynchronous TSN-based TNs. The TN interconnects three server racks, each hosting eight UPF instances (one instance per slice), with two gNBs. There are 24 5G slices, each tailored for a specific industrial service and delay constraint. Eight industrial use cases were considered to set up the traffic characteristics and E2E TN delay budget of the different slices, namely, motion control, control-to-control, mobile control panels, mobile robots, massive wireless industrial networks, closed-loop process control, process monitoring, and plant asset management.

Figure 5 shows the obtained E2E TN worst-case delay per TC and path and the per TC E2E TN delay budget (labeled as "E2E PDB" and with the respective values explicitly included). As observed, all the delay requirements are met, thus validating the proper operation of the solution for the scenario considered.

V. Conclusion

In this work, we have proposed a novel RL-based solution for configuring asynchronous TSN-based 5GB TNs, emphasizing the generalization capacity. The simulation-based PoC carried out in this work validates the proper operation of the proposal.

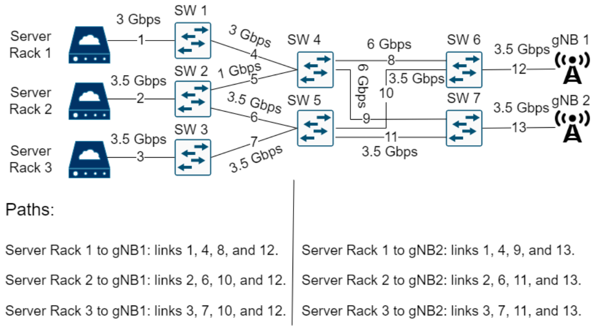

Fig. 4. TN topology and predefined paths considered in the PoC.

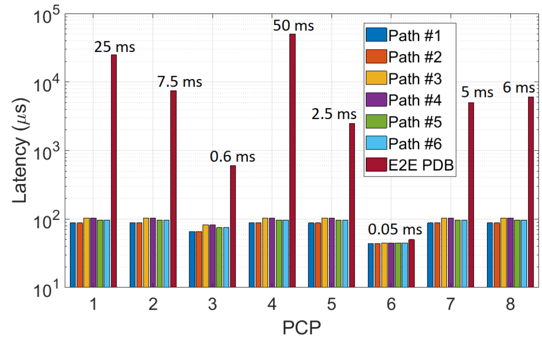

Fig. 5. TN packet delay per path and per PCP.

Acknowledgements

This work has been partially funded by the H2020 research and innovation project 5G-CLARITY (Grant No. 871428), and national research projects TRUE5G (PID2019-108713RB-C53), and 6G-CHRONOS (TSI-063000-2021-28).

References

[1] J. Prados-Garzon and T. Taleb, "Asynchronous time-sensitive networking for 5g backhauling," *IEEE Network*, vol. 35, no. 2, pp. 144–151, 2021.

[2] L. Chinchilla-Romero, J. Prados-Garzon, P. Ameigeiras, P. Muñoz, and J. M. Lopez-Soler, "5g infrastructure network slicing: E2e mean delay model and effectiveness assessment to reduce downtimes in industry 4.0," *Sensors*, vol. 22, no. 1, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/1/229

[3] "IEEE Draft Standard for Local and metropolitan area networks–Bridges and Bridged Networks Amendment: Asynchronous Traffic Shaping," *IEEE P802.1Qcr/D2.1, February 2020*, pp. 1–152, 2020.

[4] J. Specht and S. Samii, "Urgency-based scheduler for time-sensitive switched ethernet networks," in *2016 28th Euromicro Conference on Real-Time Systems (ECRTS)*, July 2016, pp. 75–85.

[5] J.-Y. Le Boudec, "A theory of traffic regulators for deterministic networks with application to interleaved regulators," *IEEE/ACM Trans. Netw.*, vol. 26, no. 6, pp. 2721–2733, Dec. 2018.

[6] J. Specht and S. Samii, "Synthesis of queue and priority assignment for asynchronous traffic shaping in switched ethernet," in *2017 IEEE Real-Time Systems Symposium (RTSS)*, 2017, pp. 178–187.

[7] J. Prados-Garzon, T. Taleb, and M. Bagaa, "Optimization of flow allocation in asynchronous deterministic 5g transport networks by leveraging data analytics," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2021.

[8] J. Prados-Garzon, L. Chinchilla-Romero, P. Ameigeiras, P. Muñoz, and J. M. Lopez-Soler, "Asynchronous time-sensitive networking for industrial networks," in *2021 Joint European Conf. on Networks and Communications 6G Summit (EuCNC/6G Summit)*, 2021, pp. 130–135.

[9] J. Prados-Garzon, T. Taleb, and M. Bagaa, "Learnet: Reinforcement learning based flow scheduling for asynchronous deterministic networks," in *2020 IEEE Int. Conf. on Communications (ICC)*, 2020, pp. 1–6.