

Optimization of Flow Allocation in Asynchronous Deterministic 5G Transport Networks by Leveraging Data Analytics

Jonathan Prados-Garzon, Tarik Taleb and Miloud Bagaa

Abstract—Time-Sensitive Networking (TSN) and Deterministic Networking (DetNet) technologies are increasingly recognized as key levers of the future 5G transport networks (TNs) due to their capabilities for providing deterministic Quality-of-Service and enabling the coexistence of critical and best-effort services. Additionally, they rely on programmable and cost-effective Ethernet-based forwarding planes. In this article, we address the flow allocation problem in 5G backhaul networks realized as asynchronous TSN networks, whose building block is the Asynchronous Traffic Shaper. We propose an offline solution, dubbed “Next Generation Transport Network Optimizer” (NEPTUNO), that combines exact optimization methods and heuristic techniques and leverages data analytics to solve the flow allocation problem. NEPTUNO aims to maximize the flow acceptance ratio while guaranteeing the deterministic Quality-of-service requirements of the critical flows. We carried out a performance evaluation of NEPTUNO in terms of the degree of optimality, execution time, and flow rejection ratio. Furthermore, we compare NEPTUNO with a novel online baseline solution for two different optimization goals. Online methods compute the flow’s allocation configuration right after the flow arrives at the network, whereas offline solutions like NEPTUNO compute a long-term configuration allocation for the whole network. Our results highlight the potential of the data analytics for the self-optimization of the future 5G TNs.

Transport Networks, QoS, Performance guarantees, Flow Allocation, Time-Sensitive Networking, 5G, Data Analytics, Asynchronous Traffic Shaper

I. INTRODUCTION

A. Motivation

Fifth Generation (5G) technology has the ambition to accommodate a wide range of services that demand diverse performance requirements. Among them are those categorized as Ultra-Reliable and Low Latency Communications (URLLCs), i.e., services with extremely stringent constraints in latency and reliability. Beyond the 5G capabilities to support URLLC traffic, a cost-effective transport network able to provide deterministic Quality of Service (QoS), i.e., bounded latency, jitter, and packet loss, is also crucial to deliver on the promises of 5G goals. Furthermore, to lower the costs, the same transport network infrastructure should enable the heterogeneous 5G services.

Jonathan Prados-Garzon, Tarik Taleb and Miloud Bagaa are with the Department of Communications and Networking, School of Electrical Engineering, Aalto University, Finland (e-mail: jonathan.prados-garzon@aalto.fi; tarik.taleb@aalto.fi; miloud.bagaa@aalto.fi). Tarik Taleb is also with the Networks and Systems research group at the Centre for Wireless Communications, Oulu University, Oulu, Finland.

Traditional network technologies fail to comply with the requirements mentioned above. On the one hand, commodity Ethernet cannot offer either deterministic QoS or latencies below tens of milliseconds [1], [2]. On the other side, although special-purpose Fieldbus technologies (e.g., Canbus, Profibus) and Industrial Ethernet might convey critical services, they can be prohibitively expensive for transport networks. Moreover, they are not suitable to support the coexistence of critical and non-performance sensitive (best effort) services simultaneously. Time-Sensitive Networking (TSN) and Deterministic Networking (DetNet) standards emerge to overcome the limitations of the predecessor technologies and are particularly attractive to realize the next generation of transport networks [3]–[5].

TSN and DetNet are gaining lots of momentum among the communities of researchers and industrials working on both wired and wireless networks. TSN is a set of standards specified by IEEE 802 aiming to define a converged layer 2 (L2) network technology that ensures the deterministic transport of the streams via IEEE 802 networks. On the other side, DetNet can be regarded as an extension of TSN to provide routes with deterministic QoS over Layer 3 (L3) routing segments. DetNet might rely on TSN to provide performance guarantees up to L2, though it can run over other underlying network technologies different from those based on Ethernet. The main objective of these technologies is to provide a flexible and dynamic network with end-to-end deterministic quality of service support in terms of latency, jitter, packet loss and reliability [1], [2].

TSN networks offer flexible and sophisticated flow control mechanisms in charge of handling the frames at each TSN bridge egress port to ensure the deterministic transport to the streams. Nonetheless, determinism and flexibility are at the expense of a combinatorial configuration complexity [6]–[8]. By way of illustration, in [7], Serna *et al.* report an experiment on the scheduling of 50 streams in a TSN network consisting of 10 synchronous switches with 32 temporal windows per port took longer than 40 hours (solver timeout). In [6], Specht and Samii show the scalability issues of the Satisfiability Modulo Theories (STM) solvers to find feasible configurations in asynchronous TSN networks. Although there are heuristics solutions proposed in the literature for configuring TSN networks to cope with the configuration problem complexity [6], the uncertainties (e.g., expected capacity demand per service or links time-to-failure) that might present some scenarios, which make the problem more challenging, have not been considered

so far in the literature. In this regard, historical data might be leveraged to adapt the network configuration to the changing network conditions.

We can distinguish between synchronous and asynchronous TSN. Synchronous TSN requires a precise and common time reference to synchronize the forwarding plane devices (TSN bridges), which might hinder the network scalability [2]. The TSN bridges use the time reference to schedule the transmission of the frame of the different flows over reserved time slots. This strategy results in poor capacity link utilization and is not suitable for sporadic traffic patterns. Considering the above, we argue asynchronous TSN can fit better the necessities of many medium and large size 5G Transport Networks (TNs) (e.g., those conveying the traffic of user-centric services) where deterministic aperiodic traffic (e.g., real-time media or smart grid) is predominant over the deterministic periodic one (e.g., closed-loop process control).

B. Context, objective and contribution

In this article, we address the flow allocation problem for asynchronous TSN 5G TNs by leveraging Data Analytics (DA). DA refers to the set of methods to extract valuable information from raw data. Most of the existing works addressing the flow allocation problem in TSN networks consider synchronous TSN in industrial scenarios [9]–[11]. Unlike in industrial scenarios, neither the number of flows nor their features can be known exactly beforehand in many mobile transport networks. For this reason, DA plays an essential role in our context. From Release 15, Third Generation Partnership Project (3GPP) standards define a DA framework to facilitate the data-gathering, analysis, and prediction within the 5G system by adding new functions to both the control and management planes. Another particularity of 5G TNs is the wide disparity and heterogeneity of the traffic they convey. Finding feasible configurations for a TSN network to simultaneously accommodate all the 5G traffic types is challenging. Moreover, this issue can be accentuated by the scale of the network. Last, some criteria or a procedure are required to cluster 5G streams into IEEE 802.1Q classes.

We focus on the Backhaul Network (BN), which is the segment of the TN connecting the radio base station sites to the core of the cellular network. We will consider IEEE 802.1Qcr Asynchronous Traffic Shaper (ATS) [12], which is based on the Urgency-Based Scheduler (UBS) [6], [13], to implement the Forwarding Plane (FP) of the BN. ATS is an asynchronous queuing algorithm that combines interleaved shaping and strict priority queues to realize per-flow deterministic QoS guarantees in a practical way. It is worth mentioning that the flow allocation problem in ATS-based networks has received little attention in the literature so far (please refer to Section II).

To solve the flow allocation problem in the scenario mentioned above, we propose a novel offline solution, dubbed “Next Generation Transport Network Optimizer” (NEPTUNO), which combines exact optimization methods and heuristics and leverages the 5G DA framework to compute a long-term configuration of the asynchronous TSN BN.

More precisely, NEPTUNO makes the following decisions: i) clustering of 5G streams into IEEE 802.1Q classes according to the 5G QoS Identifier (5QI), ii) flow-to-shaped buffer and flow-to-priority assignments at each ATS of the network, iii) paths selection to interconnect every source and destination, and iv) distribution of the end-to-end delay/jitter budget of the flows among the hops comprising each path in the network. The goal of NEPTUNO is to minimize the flow rejection ratio, while guaranteeing the QoS requirements (e.g., maximum delay budget and jitter, and minimum reliability) of the 5G flows.

We evaluate the performance of NEPTUNO in terms of the degree of optimality, runtime, and flow rejection ratio. Moreover, due to the scarce literature dealing with the flow allocation problem in asynchronous TSN BNs, we propose a novel online solution that also integrates exact methods, heuristics, and data analytics to solve the flow allocation problem. Unlike offline approaches like NEPTUNO, online solutions compute the flow’s allocation configuration right after the flow arrives at the network. Our results show that the flow rejection ratio offered by NEPTUNO is approximately 20% above the optimal one for low workloads and 10% above for high workloads for a TSN BN with 10 links and 4 delay-critical guaranteed bit rate 5QIs. Regarding the NEPTUNO’s computational complexity, our results suggest it scales linearly with the size of the network, i.e., the number of links or, equivalently, the number of ATSs. Last, the results highlight the importance of the DA to optimize the flow allocation problem in ATS-based BNs.

The remainder of the paper is organized as follows: Section II includes a brief revision of the related literature. Section III briefly describes the network architecture considered in this work and the operation of the ATS. Section IV includes a high-level formulation of the considered flow scheduling optimization problem for ATS-based networks. Section V covers the modeling of the key performance metrics considered in the problem. Section VI details the operation of NEPTUNO. Section VII describes the proposed online baseline solution. Section VIII includes the simulation results to assess the performance of NEPTUNO. Finally, Section IX concludes the paper.

II. BACKGROUND AND RELATED WORKS

This section provides some background and reviews the existing works related to data analytics for optimizing mobile networks and asynchronous TSN networks.

A. Data Analytics in 5G

Data Analytics (DA) is recognized as a key lever to unleash the full potential of 5G [14]. DA refers to the process of analyzing raw data to extract valuable information and conclusions. One of the primary applications of DA in mobile networks is to assist the management planes in optimizing the network operation to adapt it dynamically to the changing conditions. For instance, predictive analytics for forecasting traffic demands and services resources consumption can help configure the network proactively according to a given goal

TABLE I: Comparison between asynchronous TSN networks, synchronous TSN networks, and traditional asynchronous networks (strict priorities).

Network type	Advantages	Disadvantages
N1: Asynchronous TSN network	<ul style="list-style-type: none"> - Suitable for aperiodic (sporadic) deterministic traffic. - Higher scalability than N2. - Higher link utilization than N2. - Lower configuration complexity than N2. 	<ul style="list-style-type: none"> - Higher latency than N2. - More expensive than N3. - Lower link utilizations than N3. - Higher configuration complexity than N3. - Not suitable for periodic deterministic traffic.
N2: Synchronous TSN network	<ul style="list-style-type: none"> - Lowest latency. - Suitable for periodic deterministic traffic. 	<ul style="list-style-type: none"> - Not suitable for aperiodic deterministic traffic. - Most expensive technology. - Scalability issues due to time synchronization. - Most difficult to configure. - Lowest link utilization.
N3: Traditional asynchronous network	<ul style="list-style-type: none"> - Lowest configuration complexity. - Cheapest technology. - Highest link utilization. 	<ul style="list-style-type: none"> - Highest latency. - Issues to provide deterministic QoS guarantees. - Highest complexity to estimate the E2E delay bounds.

(e.g., maximizing the operator’s revenue). We refer the interested reader to [14] for a detailed explanation of a wide range of use cases in mobile networks that can benefit from DA.

Industry and researchers have widely known for a long time the importance of using DA and historical data of the users’ demands and network-related parameters to optimize the network operation. Nonetheless, 3GPP mobile networks built-in capabilities to support DA, i.e., data collection and processing functionality, have been recently adopted by including Network Data Analytics Function (NWDAF) and Management Data Analytics Function (MDAF) at the control and management planes, respectively. There is a vast literature previous to 3GPP DA framework definition that considers available predictive analytics as input to their optimization algorithms.

To mention some related works, Pateromichelakis *et al.* in [14] provide a technical overview of the DA support in 3GPP Release 15 and propose a novel integrated DA framework to overcome the limitations of the 3GPP standards. Furthermore, they showcase a particular implementation for application and Radio Access Network (RAN) analytics. In [15], Raza *et al.* present a proof-of-concept that consists of an Software-Defined Networking (SDN)/Network Functions Virtualisation (NFV)-based orchestrator that leverages Big DA to share infrastructure resources among multiple tenants to optimize the profit of a Infrastructure Provider (InP). A slice admission strategy based on Big DA for an InP is proposed in [16] by Raza *et al.*. The reported results show that using predictive DA for slices admission control yields up to 50.7% increase in InP’s profit. Afolabi *et al.* in [17] show the importance of the predictive DA to forecast the traffic load to carry out the proactive computing resource provisioning of Network Slicing Orchestrator System implemented as a softwarized network service [18].

B. Asynchronous TSN networks

One of the key ingredients to provide deterministic QoS is to implement mathematically analyzable queuing algorithms for handling the frames at the output ports of the FP bridges [1]. In other words, we can derive analytical expressions for the end-to-end (E2E) maximum delay, jitter, and frame loss, experienced by any critical flow given its characteristics, the current state of the network, and the resources allocated to it

[19]. Then, the flow allocation algorithms will rely on these expressions to decide in advance the setup and resources to be reserved for every flow in order to ensure its QoS requirements.

Many bridges implementations defined in TSN standards are synchronous, i.e., they need a precise and common time reference shared among all the TSN devices of the network. The TSN devices employ this time synchronization, for instance, to schedule the transmission of the traffic for the different flows over synchronized time slots (i.e., time-division multiplexing). IEEE 802.1AS standard specifies the way to provide synchronization for TSN by using a specific profile of the generic Precision Time Protocol [2]. Examples of TSN synchronous traffic scheduling algorithms are the IEEE 802.1Qch Cyclic Queuing and Forwarding (CQF) and IEEE 802.1Qbv Time-Aware Shaper (TAS).

The principal drawbacks of the synchronous traffic scheduling algorithms are the need for a network-wide coordinated time, which hinders the scalability of the network, and the use of reserved time slots for each flow or set of flows, which leads to a poor utilization of the link capacities [2]. TSN standards tackle these weaknesses by including the IEEE 802.1Qcr ATS, which removes the need for a common time reference among the ATS bridges and offers higher link utilization than synchronous schedulers by leveraging statistical multiplexing. It is nothing but fair, however, to note that synchronous schedulers exhibit lower delay than ATS.

In [20], Nasrallah *et al.* compare the performance offered by ATS and TAS for a typical industrial control network with a ring topology. Their results show that ATS slightly outperforms TAS for sporadic high-priority and best-effort (BE) traffic. However, ATS has problems to handle moderate and high loads of periodic deterministic traffic [20]. Despite this shortcoming and considering the expected dominance of mobile broadband traffic [21], the ATS is particularly attractive for 5G BNs due to its higher scalability as argued in [22].

Traditionally, asynchronous networks have relied on the use of traffic prioritization (strict priorities) at the bridges egress ports for QoS support (e.g., IEEE 802.1Q-2005 standard). This approach’s main drawback is that it results in arbitrarily large worst-case delays as the burstiness of the flows grows at every hop [23]. This issue can be solved by using per-hop shaping. However, the major obstacle for the practicability of per-hop

traffic regulation is that a physical queue per-flow is typically required, which is expensive. ATS overcomes this problem by introducing the interleaved shaping concept, which allows using a single queue for regulating a set of flows, each with its own shaping constraints, thus achieving a cost-effective per-hop traffic shaping.

Traditional asynchronous networks are cheap, scalable, and easy to configure as their functionality is reduced. Nonetheless, the estimation of tight delay bounds for the flows in traditional asynchronous networks is an NP-hard problem [24]. Conversely, the delay in TSN networks is more comfortable to compute, but their configuration is too complex due to their flexibility and sophisticated methods to provide deterministic QoS. Last, it shall be noted that ATS, as any other shaping mechanism, will reduce the multiplexing gains in the backhaul networks compared to bare IEEE 802.1Q.

Table I summarizes the benefits and drawbacks of the synchronous TSN, asynchronous TSN, and traditional asynchronous networks discussed above.

Several works have addressed the performance analysis of the ATS [13], [23], [25]. Specht and Samii propose the Urgency-Based Scheduler (UBS) in [13], which is the core of the ATS as a practical alternative to the existing asynchronous shapers that avoids per-flow queuing and offers a more scalable operation. Besides, they derive delay bounds for the UBS with a constant capacity link and considering both Length Rate Quotient (LRQ) and leaky bucket regulation constraints. In [23], Le Boudec formally proves the performance bounds of the UBS and introduces the concept of Minimal Interleaved Regulator that extends UBS to a broader set of regulation rules. Remarkably, they prove that any minimal interleaved regulator, placed after any arbitrary FIFO queue, does not increase the delay bound of the combination [23]. Mohammadpour *et al.* compute E2E bounds for a deterministic network based on Credit-Based Shaper (CBS) and ATS. They show that the derived bounds are tighter than those estimated as the sums of per-hop worst-case delay.

The performance evaluation of the ATS through simulation is tackled in [26]–[28]. Zhou *et al.* revisited the operation and bounds of the ATS in [26], [27]. Moreover, they assess the performance of the ATS. They conclude that ATS accomplishes effective traffic shaping and switching without requiring a global notion of time. Furthermore, they identify the strengths and weaknesses of the asynchronous algorithms UBS/Token Bucket Emulation (TBE), UBS/LRQ, and Paternoster scheduler. In [28], Grigorjew *et al.* introduce a novel simulation framework, featuring ATS and IEEE 802.1Qbu frame preemption. Their simulation results suggest a decrease in the jitter of the high priority flows thanks to frame preemption, though it negatively impacts on the latency of the low priority flows.

The flow allocation problem in asynchronous TSN networks is addressed in [6], [29]. In [6], Specht and Samii use two approaches to tackle the synthesis of shaped queues and priority assignment in UBS-based networks: a pure STM solver which always finds a feasible solution, and ii) a heuristic approach dubbed Topology Rank Solver (TRS) to cope with the high complexity exhibited by the pure STM solver. In addition, the authors explore several variants of the slack time

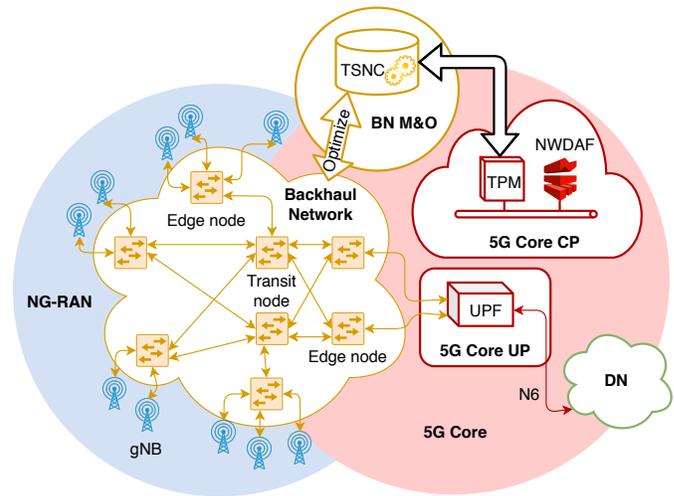


Fig. 1: 5G Network architecture.

maximization as optimization goals of the TRS. In contrast to this work, the number of flows to be allocated in the network and their characteristics are known beforehand in [6]. In our previous work [22], [29], we propose LEARNET, a deep reinforcement learning (DRL)-based online solution for the flow allocation problem in asynchronous TSN networks. The proposed solution leverages ATS analytical performance models to check the validity of the actions issued by the agent. In this way, the DRL-based solution becomes fully reliable. Please refer to [30], [31] for further benefits brought by the combined use of analytical and machine learning approaches. The solution proposed in [22], [29] is a first step on the application of DRL to solve the flow allocation problem in ATS-based networks. Nonetheless, further work is required to make the solution agnostic to the scenario (e.g., network topology).

III. 5G ATS-BASED BACKHAUL NETWORK

In this section, we briefly describe the assumed BN architecture upon which to build the envisioned flow scheduling solution.

A. Backhaul Network Architecture and Operation

The scenario considered in this work is similar to that described in [22]. It consists of an asynchronous TSN network to realize the 5G BN, i.e., to provide connectivity between the Next Generation NodeBs (gNBs) and the 5G Core, as shown in Fig. 1. The forwarding plane comprises a set of TSN bridges that support ATS. The TSN bridges include an ATS instance per egress port. Each ATS is in charge of scheduling the frames for a physical link that connects an egress port of one bridge with the other's ingress port. That is why, from now on, in formalism, we might use both terms interchangeably. To facilitate subsequent explanations, we consider two types of bridges: edge nodes and transit nodes. The edge nodes are directly connected with a 5G entity instance (e.g., gNBs, user plane function -UPF-). Besides, they include the TSN Frame Replication and Elimination for Reliability capability.

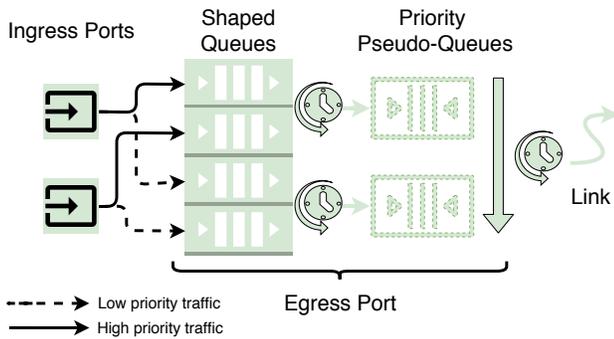


Fig. 2: ATS queuing model. The illustrated ATS has four shaped queues and two priority levels and receives traffic from two ingress ports.

To provide cross coordination between the 5G system and the transport network domain, we consider the IETF framework for Abstraction and Control of Traffic Engineered Networks [22], [32]. In this way, the 5G system can control the transport network slices through the traffic provisioning manager (TPM) in the 5G core control plane.

The primary entity at the BN management and control planes is a logically centralized TSN controller (TSNC). It is responsible for running algorithms to optimize the BN operation and configuring the bridges accordingly. These algorithms might leverage the 5G data analytics framework via the TPM. The following predictive analytics are particularly interesting to optimize the BN:

1. Temporal traffic demand per 5QI between every source and destination (traffic matrix).
2. Distribution of the guaranteed bit rate per 5QI.
3. The distribution of the flow lifetime per 5QI.

The algorithm might require further inputs, such as the network state and time-to-failure distribution for every link and device, directly from the telemetry and data analytics engine of the BN management plane.

B. ATS Operation

The ATS defines an asynchronous method for handling the frames at the egress ports of the TSN bridges [2], [33]. The ATS specified in TSN standards [33] is based on the UBS originally proposed by Specht and Samii in [13]. In fact, the ATS can be regarded as the practical implementation of the UBS in 802.1Q standards [33]. In this work, we adopt the nomenclature used in [13] for describing the ATS operation.

Figure 2 depicts the ATS queuing model. For the sake of simplicity, Fig. 2 shows only one egress port, but please note that there is an ATS instance per bridge egress port. The ATS consists of two stages of queuing: i) a set of shaped queues for interleaved shaping, and ii) a set of priority queues. All these queues follow a First Come, First Served (FCFS) discipline.

The interleaved shaping enables the use of a single queue (shaped queue) for realizing the traffic regulation of a set of flows, each with its own constraints. To that end, only the eligibility of the head-of-line (HOL) frame is checked, i.e., to examine whether the HOL frame is eligible for transmission according to the regulation constraints of its flow. If so, the

frame is released for transmission to the following queuing level. Interestingly, the interleaved shaping does not increase worst-case latency of the ATS [13], [23].

ATS supports a leaky bucket shaping constraints, i.e., it enforces an upper bound on the flows of the form $A_f(t) \leq r_f \cdot t + b_f$ [6], [13]. Where $A_f(t)$ is the accumulated amount of transmitted data until the instant t for the flow f , and r_f and b_f are respectively Committed Information Rate and Committed Burst Size in TSN standards [33]. For further details on the operation of the interleaved shaping please refer to [6], [13], [33].

The second stage in the queuing hierarchy includes one FCFS queue per priority level in the scheduler. Each queue merges the output of all shaped queues assigned to the same priority level. The transmission selection algorithm at this stage is strict priorities.

The allocation of a flow to a given ATS involves two main decisions, e.g., i) flow to shaping queue, and ii) flow to priority level assignments. These decisions are subject to the following rules: each shaped queue is associated with only one ingress port (QAR1 rule), one priority level in the previous hop (QAR2 rule), and one internal priority level (QAR3 rule). QAR2 and QAR3 rules are required to provide deterministic QoS, whereas QAR1 isolates the flows from different nodes, avoiding the propagation of non-conformant traffic overloads. These rules determine the required number of shaped queues Q to implement P priority levels. Let us assume an ATS in a bridge that aggregates traffic from L different input ports. Each input port $l \in [1, L]$ receives traffic from an ATS with P_l priority levels embedded in an adjacent node. Then, we will need $Q = P \cdot \sum_{l=1}^L P_l$ shaped buffers to realize P internal priority levels without any flow priority assignment restriction.

IV. FLOW ALLOCATION PROBLEM STATEMENT

This section covers the high-level and not a detailed description of the problem addressed together with two different approaches to solve it. But first, we start with the description of the abstract model of the network and main assumptions considered in this work.

A. Network Model

Let us consider an asynchronous TSN 5G BN as that described in Section III-A. The network consists of V asynchronous TSN bridges interconnected through E simplex links. There is a dedicated physical link that connects an egress port of one bridge with the ingress port of a different bridge. In this way, the communications are full-duplex. This network can be modeled as a digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where \mathcal{V} and \mathcal{E} denote the vertices (bridges) and the edges (links) of the graph, respectively. The weight C_e of each edge $e \in \mathcal{E}$ stands for the respective physical link capacity [6], [13], [23].

There is an ATS, whose operation is described in Section III-B, to schedule the packets at each output port of the TSN bridges. Observe, there is a one-to-one correspondence between links and ATSSs, as mentioned in Section III-A. Each ATS e is equipped with S_e shaped buffers to realize the interleaved shaping and P_e FCFS queues for traffic prioritization.

The flows arrivals to the network obey a Poisson process [34]. Each flow is upper constrained by the function $r_f \cdot t + b_f$. Each ATS enforces the corresponding traffic regulation on the flows passing through it [13]. The ATS might also enforce a maximum frame size l_f of the flow. We also assume that the flow lifetime is exponentially distributed.

The end-to-end (E2E) delay budgets of the flows are given by a set of predefined 5QIs [3], denoted as \mathcal{Q} , in the 5G system. A percentage of this E2E delay budget is spent in the BN. Besides the delay constraint, we consider the flows have jitter and reliability requirements. All these performance requisites are shared for all the flows with the same 5QI $q \in \mathcal{Q}$. Specifically, $D_q^{(max)}$, and $J_q^{(max)}$ are respectively the maximum delay and maximum jitter that a flow with 5QI $q \in \mathcal{Q}$ can experience when it is conveyed from its source to its destination in the transport network. In the same way, the BN has to ensure minimum reliability $R_q^{(min)}$ for the flows with 5QI q , i.e., probability of a seamless communication meeting all the flow's performance requisites during its lifetime.

B. Problem Formulation

Let us assume the network model described in the previous subsection and consider the following notation:

- λ_q is the arrival rate of flows with 5QI q to the network;
- $P_{B,q}$ is the blocking probability for flows with 5QI q ;
- α_q represents the cost associated with rejecting a flow with 5QI q ;
- D_f , J_f , and R_f stand for the delay, delay jitter, and reliability experienced by flow f , respectively;
- $D_q^{(max)}$, $J_q^{(max)}$, and $R_q^{(max)}$ are the E2E delay, jitter, and reliability budgets, respectively;
- \mathcal{F}_q is the set of accepted flows with 5QI q ;
- \mathcal{Q} is the set of 5QIs;
- r_f and C_e denote respectively the committed data rate of the flow f and capacity of the link e ;
- \mathcal{E} is the set of links or, equivalently, ATSs as there is a one-to-one correspondence between them;
- and \mathcal{F}_e is the set of flows passing through link e ;

Then, the flow scheduling problem addressed in this work is formulated below:

Objective :

$$\text{minimize } \left\{ \sum_{q \in \mathcal{Q}} \lambda_q \cdot \alpha_q \cdot P_{B,q} \right\} \quad (1a)$$

Constraints :

QoS assurance :

$$C1 : D_f \leq D_q^{(max)}, \quad \forall f \in \mathcal{F}_q \text{ and } q \in \mathcal{Q} \quad (1b)$$

$$C2 : J_f \leq J_q^{(max)}, \quad \forall f \in \mathcal{F}_q \text{ and } q \in \mathcal{Q} \quad (1c)$$

$$C3 : R_f \geq R_q^{(min)}, \quad \forall f \in \mathcal{F}_q \text{ and } q \in \mathcal{Q} \quad (1d)$$

Capacity constraints :

$$C4 : \sum_{f \in \mathcal{F}_e} r_f \leq C_e, \quad \forall e \in \mathcal{E} \quad (1e)$$

$$C5 : QAR1, QAR2, \text{ and } QAR3 \quad \forall e \in \mathcal{E} \quad (1f)$$

Stated briefly, the problem above aims at minimizing the flow rejection probability (when $\alpha_q = 1 \forall q \in \mathcal{Q}$) while ensuring the QoS constraints (e.g., delay, delay jitter, and reliability) for all the accepted flows and subject to technological constraints (e.g., link capacities, QAR rules, and shaping buffers sizes). The variables α_q serve to make the problem model more generic by enabling other optimization goals, such as the maximization of the operator's profit.

Solving the flow allocation problem in ATS-based networks entails the following decisions:

- i) the number of disjoint paths required to transport the flow in order to ensure its reliability requirement;
- ii) choosing the path(s) to convey the stream, i.e., the set of links to be traversed by the stream from its source to its destination;
- iii) the distribution of the flow's E2E delay and jitter budgets among the links of the path(s).
- iv) and selecting the primary configuration parameters at each involved ATS, e.g., flow-to-shaping buffer and the shaping buffer-to-priority levels assignments.

C. Approaches to Solve the Problem

We can distinguish two approaches for solving the flow allocation problem in an ATS-based 5G BN, namely, online and offline methods. On the one hand, online methods compute the flow's allocation configuration right after it arrives at the network. To that end, they might run an optimization algorithm to find, given a goal, the optimal allocation for the incoming flow. Conversely, offline methods compute a long-term configuration for the whole network considering the different types of traffic. Specifically, the flows are somehow clustered into classes, e.g., according to their 5QI or 5QI and slice (Single Network Slice Selection Assistance Information -S-NSSAI-), and the optimal allocation configuration is computed for each class. Observe that, in this case, the allocation configuration for each flow is predetermined, and the Access Control Mechanism (ACM) becomes a lightweight process that checks whether there are enough resources (links capacities and buffer space) for the incoming flow.

The right choice between the options mentioned above likely depends on the specific scenario. By way of illustration, if there are stringent requirements for the control plane (e.g., ultra-low response time for a packet data unit (PDU) session establishment), then the offline approach would be preferred as it allows for lighter ACMs as previously stated. On the other hand, the online approach could be more suitable for large scale networks as computing a global allocation configuration without sacrificing the degree of optimality might be too complex. Furthermore, online methods are more flexible as they enable flows to have a specific allocation configuration, whereas, in offline ones, the allocation configuration is shared among all the flows of the same class. This advantage of the online methods is at the expense of storing more state information. Specifically, they need to have an entry per ongoing flow with the whole allocation configuration for it. Instead, only the committed rate and burst size need to be stored with an offline approach.

V. PERFORMANCE MODEL FOR THE ATS-BASED BN

The objective here is to provide expressions to estimate the different E2E QoS metrics (e.g., flow rejection probability, maximum delay, jitter, and reliability) of a given flow passing through a set of deterministic switches in the BN. We will consider the UBS based switch operation as described in Section III-B.

A. Flow Rejection Probability

We assume that the flow interarrival times are exponentially distributed, whereas the flow lifetimes follow a general (arbitrary) distribution. Then, for each 5QI $q \in Q$ we can model the BN as an $M/G/N_q/N_q$ queue (Kendall's notation), where N_q stands for the number of flows with 5QI q that the BN can accept simultaneously. Please note that, in our case, the interarrivals and service processes (queuing theory terms) stand for the flow inter-arrival and flow lifetime processes, respectively. Therefore,

$$P_{B,q} \leq \frac{\lambda_q}{N_q \cdot \mu_q + \lambda_q} \quad (2)$$

is an upper bound of the flow rejection probability experienced by the flows of the 5QI q in the BN [35], [36]. Where λ_q and μ_q denote the flow arrival rate and the inverse of the mean flow lifetime, respectively, for 5QI q . Observe that the above expression is a convex function of N_q .

The key assumption here is the flow interarrival times follow an exponential distribution. In our previous work [34], we show empirically that the exponential distribution fits well the aggregated service request interarrival times (flow interarrival times) in a Fourth Generation (4G) mobile network (see [34, Fig. 9]).

Despite the fact mentioned above, we checked whether different flow arrival processes significantly impact our solution's performance, which uses (2). Specifically, we considered the exponential, Erlang-2 (Erl2), and two-stage hyperexponential (Hyp2) distributions to simulate the flow arrivals. The Erl2 and Hyp2 can respectively approximate any distribution with a coefficient of variation lower and greater than one [37]. The results showed a negligible performance degradation when the flow interarrivals times follow either the Erl2 or Hyp2 distributions (refer to Section VIII for further details).

B. Reliability

The Frame Replication and Elimination for Reliability (FRER) capability of TSN enables to forward several replicas of a flow f through disjoint paths in order to guarantee the flow's minimum reliability $R_f^{(min)}$. Here, we will consider the reliability as the probability of network success to carry out the communication of a given flow f during its entire lifetime τ_f . We suppose that every path consists of N_H independent links whose time-to-failure follows an exponential distribution with mean χ [38]. Then, the disjoint paths supporting the transmission of the replicas can be modeled as a reliability block diagram with a parallel-series configuration [38]. Under

these assumptions, the probability of service disruption $R_f^{(q)}$ for the flow f is given by:

$$R_f^{(q)} = 1 - \left(1 - e^{-N_H \cdot \frac{\tau_f}{\chi}}\right)^{N_P} \quad (3)$$

Thus, the required number of replicas for a given flow f and a minimum level of reliability $R_f^{(min)}$ can be estimated solving (3) for N_P :

$$N_P \geq \left\lceil \frac{\log\left(1 - R_f^{(min)}\right)}{\log\left(1 - e^{-N_H \cdot \frac{\tau_f}{\chi}}\right)} \right\rceil \quad (4)$$

C. E2E Maximum Delay and Jitter

The derivation of a per-hop delay bound of a given flow, hereinafter referred to as flow of interest (foi), is provided in [13], [33]. The maximum E2E delay experienced for any packet of a given flow when it traverses a given path \mathcal{P} is upper-bounded as:

$$D_f \leq \sum_{e \in \mathcal{E}_f} d_{f,e}^{(max)} = \sum_{e \in \mathcal{P}} \max_{f \in \mathcal{F}} \left\{ \frac{\hat{b}_H^{(e)} + \hat{b}_{SP}^{(e)} + \hat{l}_L^{(e)}}{C_e - \hat{r}_H^{(e)}} + \frac{\hat{l}_f}{C_e} \right\} \quad (5)$$

where $\hat{b}_H^{(e)} = \sum_{f \in \mathcal{F}_H} b_f$ and $\hat{r}_H^{(e)} = \sum_{f \in \mathcal{F}_{SP}} r_f$ respectively denote the aggregated burstiness and data rate generated by the set of flows \mathcal{F}_H with higher priority level than the foi at link $e \in \mathcal{P}$, $\hat{b}_{SP}^{(e)} = \sum_{f \in \mathcal{F}_{SP}} b_f$ is the burstiness of the set of flows \mathcal{F}_{SP} with the same priority level as foi at link $e \in \mathcal{P}$, $\hat{l}_L^{(e)}$ represent the maximum packet size for the set of flows with lower priority levels than foi, \hat{l}_f is the packet size of foi, and C_e is the transmission capacity at a given hop e .

The maximum packet jitter delay of the foi f at the path \mathcal{P} can be computed assuming a best-case delay per node of \hat{l}_f/C_e . Then,

$$J_f \leq \sum_{e \in \mathcal{P}} \max_{f \in \mathcal{F}} \left\{ \frac{\hat{b}_H^{(e)} + \hat{b}_{SP}^{(e)} + \hat{l}_L^{(e)}}{C_e - \hat{r}_H^{(e)}} \right\} \quad (6)$$

VI. OFFLINE SOLUTION

In this section, we detail the proposed offline solution (refer to Section IV-C), dubbed "Next Generation Transport Network Optimizer" (NEPTUNO), to compute a long-term configuration for allocating the flows in an Asynchronous Traffic Shaper (ATS)-based 5G Backhaul Network (BN).

A. Operation

NEPTUNO is an offline solution for flow allocation in ATS-based 5G BNs. It aims to find a long-term configuration of the ATS-based BN to maximize either the flow acceptance ratio or the operator's profit. The key idea behind offline methods is to exploit the fact that workload fluctuations typically occur at high time scales [39] to minimize the network configurations and allow for a lightweight ACM. Nonetheless, it does not preclude that the frequency to trigger the optimization process can be adapted according to the specific dynamics of the workload.

Figure 3 shows the main steps of NEPTUNO to make the decisions listed in Section IV-B. First, NEPTUNO collects

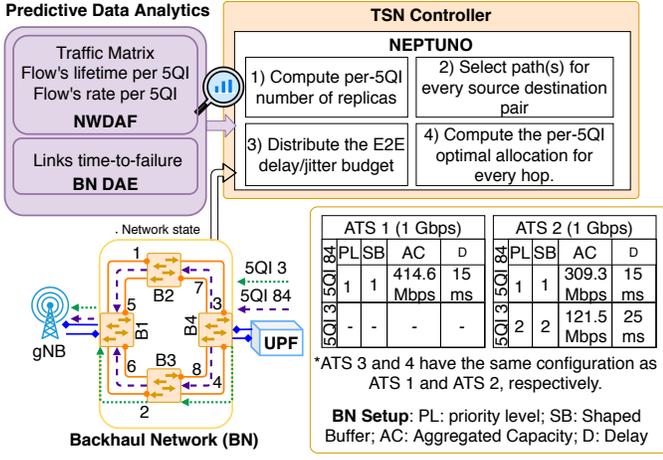


Fig. 3: Main stages of NEPTUNO for computing the optimal configuration of the network and example illustrating the primary configuration parameters for two 5QIs.

the data analytics of interest and network state information. Specifically, NEPTUNO uses the following analytics as inputs:

- A1) the mean flow lifetime duration τ_q for each 5QI q ,
- A2) the foreseen flow arrival rate per 5QI between every source-destination pair in the BN,
- A3) the link mean time-to-failure χ , and
- A4) the mean committed rate r_q for each 5QI q .

Besides these analytics, NEPTUNO relies on the knowledge about the BN topology $(\mathcal{V}, \mathcal{E})$, the ATS bridges characteristics (e.g., number of shaped queues and their sizes, and transmission rate at each link), and the characteristics of each 5QI (e.g., QoS requirements, maximum packet length, and flow burstiness) to compute the network configuration.

Next, NEPTUNO executes the optimization algorithm for finding the optimal configuration of the network. The first step of the optimization process is to compute the number of packet replicas required for each 5QI in order to assure its minimum reliability. The source node will send the different replicas of a packet through disjoint paths by using TSN FRER functionality [40]. The number of necessary packet replicas is estimated using (4). The subsequent stages of NEPTUNO use this information.

Secondly, NEPTUNO runs a path selection algorithm whose objective is to balance the workload through the different transit links of the BN. To that end, it uses the number of required flow replicas computed in the previous step, and data analytics A1 and A2 as input. The algorithm chooses the path(s) between every source-destination pair.

Thirdly, NEPTUNO distributes the delay/jitter budget among the hops of each path chosen in the previous step. We explored the following three different approaches to distribute the E2E delay/jitter budget Ψ_q^{E2E} for each 5QI $q \in \mathcal{Q}$ among the hops of each predefined path $\mathcal{P} \subseteq \mathcal{E}$:

- Based on the link capacities, i.e., the delay budget $\Psi_q^{(e)}$ at a given hop e is given by $\Psi_q^{(e)} = 1 / (C_e \cdot \sum_{h \in \mathcal{P}} 1/C_h) \cdot \Psi_q^{E2E} \forall e \in \mathcal{P}$.
- Based on the expected maximum link utilization, i.e., $\Psi_q^{(e)} = 1 / (\sum_{e \in \mathcal{P}} E[\hat{r}^{(e)}] / C_e) / \sum_{h \in \mathcal{P}} E[\hat{r}^{(h)}] / C_h$.

$\Psi_q^{E2E} \forall e \in \mathcal{P}$, where $\hat{r}^{(e)}$ is the foreseen aggregated traffic rate to be served by the hop e .

- Equal budget for each hop, i.e., $\Psi_q^{(e)} = 1/N_H \cdot \Psi_q^{E2E} \forall e \in \mathcal{P}$, where $N_H = |\mathcal{P}|$ is the number of hops of the path \mathcal{P} .

NEPTUNO implements the first approach, i.e., that based on the link capacities, because we observed it yields the best results in terms of flow acceptance ratio.

Fourthly, NEPTUNO computes the optimal configuration for each last hop by solving the Mixed Integer Linear Program (MILP) formulated in the next subsection. Specifically, it finds the 5QI-to-shapped buffers and 5QI-to-priority assignments, and the per-5QI maximum aggregated capacity to be allocated at the corresponding ATS (see Fig. 3).

Last, the optimization of the transit hops is performed by solving another MILP whose objective is given by (20). Furthermore, this MILP includes additional constraints compared to that for the last hops in order to ensure the QAR2 rule in the subsequent hops.

Once the algorithm is done, NEPTUNO disseminates the computed configuration to the ACM and the BN in order to apply it.

Figure 3 includes a TSN BN configuration example computed by NEPTUNO at the bottom. It considers two 5QIs: 3 and 84. The traffic of the 5QI 84 is forwarded through two disjoint paths to ensure its minimum reliability. The configuration of the involved ATSs is at the bottom right of Fig. 3. For instance, the maximum aggregated capacities reserved for 5QIs 84 and 3 are 309.3 Mbps and 121.5 Mbps in ATS 2, respectively.

B. Per-ATS Optimization

This subsection details the problem formulation to find the optimal flow allocation within a single ATS in the network. Linearization techniques are applied to the original program to convert it into an MILP in order to solve it efficiently. For simplicity, we omit here the index or any other reference to the specific ATS in the notation as we address the optimization of a single ATS.

To find the per-ATS (or per-link) optimal scheduling, NEPTUNO distinguishes two types of ATS: i) the last hop, i.e., the last ATS traversed by the flow before it leaves the TSN domain; and ii) the rest of ATSs, referred to as transit hops. First, the optimization goals for each of these types of ATS are different. Second, there are additional constraints for the transit hops to enforce the QAR2 rules (see Section III-B) in the next hops given the NEPTUNO's operation.

1) *Flow allocation optimization in the last hop*: Let us start with the problem formulation of the last hop. Let λ_q and μ_q denote the arrival rate and the inverse of the mean lifetime of the flows with 5QI $q \in \mathcal{Q}$, respectively. Each flow with 5QI q is characterized by its committed rate r_q and burst size b_q , and its maximum frame size l_q . Let Ψ_q denote the delay/jitter budget of the ATS for the 5QI q (computed in NEPTUNO's step 3). The ATS is characterized by its number of shaping buffers S , its maximum number of priority levels P , and the link capacity C . There is a cost α_q related to the rejection of

a flow with 5QI q . We initially define the following decision variables for the problem:

- N_q denotes the average number of flows with 5QI q that can be simultaneously served by the ATS.
- $N_{q,p}$ denotes the average number of flows with 5QI q that can be simultaneously served by the ATS at priority level p .
- $U_{q,p}$ is a binary variable indicating whether the flows with 5QI q are assigned to the priority level p ($=1$) or not ($=0$).

Formally, the flow allocation problem for the last hop can be formulated as follows:

$$\text{minimize } \left\{ \sum_{q \in \mathcal{Q}} \lambda_q \cdot \alpha_q \cdot \left(\frac{\lambda_q}{N_q \cdot \mu_q + \lambda_q} \right) \right\} \quad (7a)$$

s.t.:

$$\sum_{q \in \mathcal{Q}} N_q \cdot r_q \leq C \quad (7b)$$

$$0 \leq N_q \leq \sum_{p=1}^P N_{q,p} \quad \forall q \in \mathcal{Q} \quad (7c)$$

$$0 \leq N_{q,p} \leq N_q^U \quad \forall q \in \mathcal{Q}, p \in [1, P] \quad (7d)$$

$$N_{q,p} \cdot N_{q,k} \leq 0 \quad \forall q \in \mathcal{Q}; k \neq p \in [1, P] \quad (7e)$$

$$\forall p \in [1, P], q \in \mathcal{Q} \quad \sum_{q \in \mathcal{Q}} \sum_{k=1}^p N_{q,k} \cdot b_q + l_{max} \leq (U_{q,p} \cdot \Psi_q + (1 - U_{q,p}) \cdot \Psi^U) \cdot \left(C - \sum_{q \in \mathcal{Q}} \sum_{k=1}^{p-1} N_{q,k} \cdot r_q \right) \quad (7f)$$

$$\sum_{p=1}^P \bigvee_{q \in \mathcal{Q}} U_{q,p} \cdot I_q \leq S \quad (7g)$$

$$\sum_{p=1}^P U_{q,p} = 1 \quad (7h)$$

Objective (7a) aims at minimizing the flow rejection probability when $\alpha_q = 1$. As stated in Section IV-B, α_q variables serve to make the problem more generic by enabling the definition, for instance, of business goals like the maximization of the network operator's profit. The objective equation results from the combination of (1a) and (2). The latter is an upper bound of the flow rejection probability for $M/G/N_q/N_q$ queuing systems.

Constraint (7b) ensures that the capacity C of the respective link will not be exceeded. Constraints (7c) and (7d) fix the bounds of N_q and $N_{q,p}$, respectively. The upper bound N_q^U for $N_{q,p}$ could be computed as $N_q^U = C/r_q$. Alternatively, we could solve the problem (7a) s.t. (7b) for N_q , and set N_q^U to its solution N_q^* , i.e., $N_q^U = N_q^*$. These bounds are lower than C/r_q , thus reducing the complexity of the problem. Constraints (7e) enforce each 5QI is assigned to only one priority level. Constraints (7f) guarantee the delay and jitter budgets allocated to the ATS by NEPTUNO in step 3 (see Fig.

3) for every 5QI. Constraint (7g) assures QAR1 and QAR3 rules are met (refer to Section III-B). The QAR2 rule in the last hop will be enforced by including constraints in the transit hops, as we will see later. Last, constraints (7h) is equivalent to (7e). We will link these two set of constraints later.

The problem (7a) s.t. (7b)-(7h) is a non-convex mixed-integer nonlinear program as constraints (7e) and (7f) involves the product of several decision variables. Nonetheless, we can easily remove these products to convert it into a convex mixed-integer nonlinear program, which can be solved more efficiently.

Trivially, (7d) and (7e) can be rewritten as a set of linear constraints using $U_{q,p}$ decision variables as follows:

$$0 \leq N_{q,p} \leq U_{q,p} \cdot N_q^U \quad \forall q \in \mathcal{Q}, p \in [1, P] \quad (8a)$$

$$0 \leq N_{q,k} \leq (1 - U_{q,p}) \cdot N_q^U \quad \forall q \in \mathcal{Q}, k, p \in [1, P] : k \neq p \quad (8b)$$

For the linearization of (7f), first, we include new decision variables x_p that stand for the effective link capacity perceived by the priority level p :

$$x_p = C - \sum_{q \in \mathcal{Q}} \sum_{k=1}^{p-1} N_{q,k} \cdot r_q \quad \forall p \in [1, P] \quad (9)$$

Then, we define new decision variables $y_{q,p} = x_p \cdot U_{q,p}$. By substitution of x_p and $y_{q,p}$ in (7f) and after some simple algebraic manipulations, (7f) becomes:

$$\sum_{q \in \mathcal{Q}} \sum_{k=1}^p N_{q,k} \cdot b_q + l_{max} \leq y_p \cdot (\Psi_q - \Psi^U) + x_p \cdot \Psi^U \quad (10)$$

$$\forall p \in [1, P], q \in \mathcal{Q}$$

Also, the following set of linear constraints must be added:

$$x_p - (1 - U_{q,p}) \cdot M \leq y_{q,p} \leq x_p + (1 - U_{q,p}) \cdot M \quad (11)$$

$$\forall q \in \mathcal{Q}, p \in [1, P]$$

$$0 \leq y_{q,p} \leq U_{q,p} \cdot M \quad \forall q \in \mathcal{Q}, p \in [1, P] \quad (12)$$

Now, the problem (7a) s.t. (7b), (7c), (7g), (7h), (8a), (8b), (9), (10), (11), and (12) is a convex mixed-integer nonlinear program (CMINLP) and equivalent to the problem (7a) s.t. (7b)-(7h), i.e., they have the same solution, but it can be solved more efficiently.

The above problem can still be converted into a mixed-integer linear program (MILP), which can be typically solved much faster than a CMINLP. The previous statement is supported by our results. To that end, we have to linearize the cost function (7a) and the constraint (7g).

To linearize the objective, we define the variables $z_q = \lambda_q / (N_q \cdot \mu_q + \lambda_q) \forall q \in \mathcal{Q}$ and substitute them in (7a). Additionally, we include the following constraints:

$$w_q \cdot \mu_q + \lambda_q \geq \lambda_q \quad \forall q \in \mathcal{Q} \quad : w_q = N_q \cdot z_q \quad (13)$$

$$N_q = \sum_{i=0}^{n_q} \beta_{q,i} \cdot 2^i; \quad w_q = \sum_{i=0}^{n_q} \rho_{q,i} \cdot 2^i \quad (14)$$

$$\forall q \in \mathcal{Q} \quad : n_q = \min \{ n : 2^n \geq N_q^U \}$$

$$\begin{aligned} z_q - (1 - \beta_{q,i}) \cdot N_q^U &\leq \rho_{q,i} \leq z_q + (1 - \beta_{q,i}) \cdot N_q^U \\ 0 &\leq \rho_{q,i} \leq \beta_{q,i} \cdot N_q^U \quad \forall q \in \mathcal{Q}; \quad \forall i \in [1, n_q] \end{aligned} \quad (15)$$

Where $\beta_{q,i} \forall q \in \mathcal{Q}, i \in [1, n_q]$ are auxiliary binary decision variables to store the binary representation of N_q . On the other side, $\rho_{q,i} = \beta_{q,i} \cdot z_q \forall q \in \mathcal{Q}, i \in [1, n_q]$ are also decision variables.

Last, to remove the maximum operation, we substitute (7g) by the following set of constraints:

$$\sum_{p=1}^P \kappa_{|\mathcal{Q}|,p} \leq S \quad (16)$$

$$\begin{aligned} U_{q,p} \cdot I_q &\leq \kappa_{q,p} \leq U_{q,p} \cdot I_q + \phi_{q,p} \cdot I^U \\ \forall q &\in [2, |\mathcal{Q}|], \forall p \in [1, P] \end{aligned} \quad (17)$$

$$\begin{aligned} \kappa_{q-1,p} &\leq \kappa_{q,p} \leq \kappa_{q-1,p} + (1 - \phi_{q,p}) \cdot I^U \\ \forall q &\in [3, |\mathcal{Q}|], \forall p \in [1, P] \end{aligned} \quad (18)$$

$$U_{1,p} \cdot I_1 \leq \kappa_{2,p} \leq U_{1,p} \cdot I_1 + \phi_{2,p} \cdot I^U \quad \forall p \in [1, P] \quad (19)$$

The resulting optimization problem with objective $\text{minimize} \left\{ \sum_{q \in \mathcal{Q}} \lambda_q \cdot \alpha_q \cdot z_q \right\}$ and constraints (7b), (7c), (7h), (8a), (8b), (9), (10), (11), (12), (13), (14), (15), (16), (17), (18), and (19) is a MILP.

2) *Flow allocation optimization in the transit hops:* For the transit nodes, NEPTUNO considers the following optimization goal:

$$\text{minimize} \left\{ \sum_{q \in \mathcal{Q}} (N_q - N_q^T)^2 \right\} \quad (20)$$

Where N_q^T is the target mean number of flows with 5QI q to be allocated to the ATS. Let $\mathcal{E}_{agg} \subset \mathcal{E}$ be the set of links that the ATS aggregates. Then, $N_q^T = \sum_{e \in \mathcal{E}_{agg}} N_q^{(e)*}$, where $N_q^{(e)*}$ is the optimal mean number of flows with 5QI q previously computed for the ATS or link $e \in \mathcal{E}_{agg}$.

Objective (20) is equivalent to:

$$\text{minimize} \left\{ \sum_{q \in \mathcal{Q}} |N_q - N_q^T| \right\} \quad (21)$$

which can be linearized as follows:

$$\text{minimize} \left\{ \sum_{q \in \mathcal{Q}} \Omega_q \right\} \quad (22)$$

subject to:

$$\Omega_q \geq N_q - N_q^T, \quad \Omega_q \geq N_q^T - N_q \quad \forall q \in \mathcal{Q} \quad (23)$$

As previously mentioned, we have also to include additional constraints to enforce QAR2 rules in the next hops. Let $U_{q,p}^{(e)*}$ denote a binary variable indicating whether flows with 5QI q are assigned to the priority level p ($=1$) or not ($=0$), which was previously computed for the ATS $e \in \mathcal{E}_{agg}$. And let $S_{idle}^{(e)}$ the number of shaping buffers that have not been assigned to any 5QI at the ATS $e \in \mathcal{E}_{agg}$, i.e., $S_{idle}^{(e)} =$

$S^{(e)} - \sum_{p=1}^P \max_{q \in \mathcal{Q}} \left\{ U_{q,p}^{(e)*} \cdot I_q^{(e)} \right\}$. Then, the following set of constraints need to be added:

$$\begin{aligned} \sum_{m=1}^{P^{(e)}} \sum_{q=1}^{|\mathcal{Q}|-1} \sum_{k=q+1}^{|\mathcal{Q}|} U_{q,m}^{(e)*} \cdot U_{k,m}^{(e)*} \cdot \sum_{p=1}^P U_{q,p} \cdot (1 - U_{k,p}) &\leq S_{idle}^{(e)} \\ \forall e &\in \mathcal{E}_{agg} \end{aligned} \quad (24)$$

Again, constraints (24) include the products of two decision variables. Then, we include the decision binary variables $\delta_{q,k,p} = U_{q,p} \cdot (1 - U_{k,p})$, substitute them in (24) and add additional constraints to “linearize” the products as follows:

$$\begin{aligned} \sum_{m=1}^{P^{(e)}} \sum_{q=1}^{|\mathcal{Q}|-1} \sum_{k=q+1}^{|\mathcal{Q}|} U_{q,m}^{(e)*} \cdot U_{k,m}^{(e)*} \cdot \sum_{p=1}^P U_{q,p} - \delta_{q,k,p} &\leq S_{idle}^{(e)} \\ \forall e &\in \mathcal{E}_{agg} \end{aligned} \quad (25a)$$

$$U_{q,p} - (1 - U_{k,p}) \leq \delta_{q,k,p} \leq U_{q,p} + (1 - U_{k,p}) \quad (25b)$$

$$\delta_{q,k,p} \leq U_{k,p} \quad \forall p \in [1, P], q, k \in \mathcal{Q} : q \neq k \quad (25c)$$

In short, the flow allocation optimization problem in transit nodes is formulated as the CMINLP (20) s.t. (7b), (7c), (7g), (7h), (8a), (8b), (9), (10), (11), (12), (25a), (25b), and (25c); or as the MILP (21) s.t. (7b), (7c), (7h), (8a), (8b), (9), (10), (11), (12), (13), (14), (15), (16), (17), (18), (19), (22), (23), (25a), (25b), and (25c).

VII. ONLINE SOLUTION

In this section, we propose an online solution (refer to Section IV-C) for the flow allocation problem in ATS-based TSN networks. For the online approach, the flow allocation involves the selection of the path(s) and and the configuration of each involved ATS all along the chosen path(s) for every incoming flow, referred to as flow of interest (foi). These decisions are subject to the QoS constraints fulfillment of the foi and all the ongoing flows in the network.

First, the number of required disjoint paths I_{foi} for the foi is computed using (4) to ensure foi’s minimum reliability constraint. Then, our solution searches for the most offloaded path in the set of precomputed paths \mathcal{P}_d^s between foi’s source $s \in \mathcal{V}$ and foi’s destination $d \in \mathcal{V}$. In this way, similarly to NEPTUNO, the solution contributes to reduce the load imbalances through the network. If applicable, for the path selection of all the remaining replicas, we follow the same criterion. However, in order to assure we are selecting disjoint paths, the chosen paths for the previous replicas and those paths sharing any link with the selected ones are excluded from the search space.

For every selected path $\mathcal{P} \subseteq \mathcal{E}$, a MILP is solved to find the flow allocation configuration at every ATS along the path.

The decision variables of the MILP are $u_{e,p} \forall e \in \mathcal{P}, p \in [1, P_e]$ and $\Psi_{foi,e} \forall e \in \mathcal{P}$. The variables $u_{e,p}$ are binary and indicate whether the foi is assigned to priority level p at ATS $e \in \mathcal{P}$. The variables $\Psi_{foi,e}$ set the maximum delay/jitter the foi can experience at each traversed ATS e of the path \mathcal{P} . The use of $\Psi_{foi,e}$ reduces the required number of constraints to guarantee the ongoing flows’ QoS requisites are always met.

Let P_{HD} and P_{LD} be the probabilities that the next incoming flow has a higher and lower end-to-end delay/jitter budget than the foi. P_{HD} and P_{LD} are input variables of the MILP provided by predictive analytics. The foi is characterized by its committed data rate r_{foi} and burst size b_{foi} , and its maximum frame size l_{foi} . Its end-to-end delay and jitter constraints are denoted as D_{foi}^{max} and J_{foi}^{max} , respectively. The rest of notation is similar to that defined in Section V-C.

The MILP to find the optimal allocation for the foi is formulated as follows:

Online solution candidate objective 1:

$$\begin{aligned} \text{minimize} \left\{ \sum_{e \in \mathcal{P}} \sum_{p=1}^{P_e^{max}} u_{e,p} \cdot (P_{HD} \cdot p + P_{LD} \cdot (P_e^{max} - p)) \right. \\ \left. + \sum_{e \in \mathcal{P}} \left| \Psi_{foi,e} - \gamma_e \cdot \left(J_{foi}^{max} \wedge D_{foi}^{max} - \sum_{e \in \mathcal{P}} \frac{l_{foi}}{C_e} \right) \right| \right\} \quad (26a) \end{aligned}$$

Online solution candidate objective 2:

$$\text{minimize} \left\{ \sum_{e \in \mathcal{P}} \sum_{p=1}^{P_e^{max}} u_{e,p} \cdot \frac{\hat{b}_H^{(e)} + \hat{b}_S^{(e)} + \hat{l}_L}{C_e - \hat{r}_H^{(e)}} + \frac{l_f}{C_e} \right\} \quad (26b)$$

s.t. :

$$\sum_{p=1}^{P_e} \left(\hat{r}_{e,p}^{(S)} + u_{e,p} \cdot r_{foi} \right) \leq C_e \quad \forall e \in \mathcal{P} \quad (26c)$$

$$\begin{aligned} \sum_{k=1}^p \left(\hat{b}_{e,k}^{(s)} + u_{e,k} \cdot b_{foi} \right) + \hat{l}_L \leq \\ \Psi_{e,p}^{min} \cdot \left(C_e - \hat{r}_{e,p}^{(H)} - \sum_{k=1}^{p-1} u_{e,k} \cdot r_{foi} \right) \quad \forall p \in [1, P_e], e \in \mathcal{P} \quad (26d) \end{aligned}$$

$$\sum_{p=1}^{P_e} \frac{\hat{b}_{e,p}^{(H)} + \hat{b}_{e,p}^{(S)} + \hat{l}_{e,p}^{(L)}}{C_e - \hat{r}_{e,p}^{(H)}} \cdot u_{e,p} \leq \Psi_{foi,e} \quad \forall e \in \mathcal{P} \quad (26e)$$

$$\sum_{e \in \mathcal{P}} \Psi_{foi,e} = J_{foi}^{max} \wedge D_{foi}^{max} - \sum_{e \in \mathcal{P}} \frac{l_{foi}}{C_e} \quad (26f)$$

$$\begin{aligned} u_{e_h,p} + u_{e_{h-1},k} - 1 \leq u_{k,p}^{(e)}; \quad u_{e,p} \leq u_{e,p}^u \\ k \in [1, \dots, P_{e_{h-1}}] \& p \in [1, \dots, P_{e_h}] \& e \in \mathcal{P} \quad (26g) \end{aligned}$$

$$\sum_{p=1}^{P_e} u_{e,p} = 1 \quad \forall e \in \mathcal{P} \quad (26h)$$

We consider two different candidate objectives for the baseline online flow allocation solution. Objective (26a) aims to perform the foi priority assignment at each ATS e according to its delay/jitter constraints. Specifically, it helps to allocate the flows with the most stringent delay/jitter constraints at the highest priority levels. Observe that we are considering that lower indexes correspond to higher priority levels. Objective (26a) also aids to set the delay/jitter budget $\Psi_{foi,e}$ at each hop of the path according to a criterion given by the constants γ_e . Specifically, similarly to NEPTUNO, we set $\gamma_e = 1 / (C_e \cdot \sum_{h \in \mathcal{P}} 1/C_h)$. On the other hand, objective

(26b) aims to find the flow allocation configuration that minimizes the E2E delay of the foi at a given path \mathcal{P} . The reason behind this choice is that the SMT-assisted heuristic solution proposed in [6] for the flow allocation problem considers the maximization of the flow slack time as optimization goal.

Constraint (26c) ensures that the capacity of any ATS/link $e \in \mathcal{P}$ will not be exceeded after allocating the foi. Constraints (26d) and (26e) guarantee respectively that the E2E delay/jitter requirements of the ongoing flows and the foi will be met after allocating the foi. In (26d), $\Psi_{e,p}^{min}$ denotes the most stringent delay/jitter constraint at ATS e and priority level p , i.e., the minimum delay/jitter budget of the ongoing flows allocated at priority level p of the ATS e . Note that thanks to set $\Psi_{foi,e}$, we only need one constraint per active priority level (with ongoing flows) and ATS in the path \mathcal{P} instead of one constraint per ongoing flow affected by the foi allocation. Constraint (26f) ensures that the E2E delay or jitter budget (the most restrictive one) is distributed along the path. This constraint contributes to improve the acceptance ratio.

Constraints (26g) serve to enforce QAR1, QAR2, and QAR3 rules given the number of shaping buffers S_e at each ATS $e \in \mathcal{P}$. These constraints rely on an auxiliary algorithm that set the binary variables $u_{e,p}^u$, which equals 0 if the priority level at node e is unfeasible to allocate the foi, and $u_{k,p}^{(e)}$, which equals 0 if the foi cannot be simultaneously assigned to priority levels k and p of the adjacent nodes e and $e-1$ in the path \mathcal{P} . Basically, the algorithm checks for each ATS e_h if there is an active shaping buffer with enough space and assigned to the internal priority level p , the input port of the foi, and the priority level k in the previous ATS e_{h-1} in \mathcal{P} . If so, it sets $u_{k,p}^{(e)} = 1$ and records the id of the shaping buffer meeting the conditions to which allocate the foi in the case program (26a)-(26h) chooses the respective priority level p . If not, the algorithm checks whether there is any idle shaping buffer. If yes, it sets $u_{k,p}^{(e)} = 1$ and saves the id of any idle buffer. Otherwise, it sets $u_{k,p}^{(e)} = 0$. Similarly, when there is no idle shaping buffers or active shaping buffers assigned to the internal priority level p and the foi's input port at the ATS e , the algorithm sets $u_{e,p}^u = 0$.

Last, constraint (26h) imposes that only one priority level is assigned to the foi at each hop $e \in \mathcal{P}$.

VIII. RESULTS

A. Experimental Setup

We carried out the performance evaluation of NEPTUNO by using an event-driven simulator of a BN. We considered the network topologies shown in Figs. 4 and 5. Specifically, The topology depicted in Fig. 5 was used to evaluate the degree of optimality offered by NEPTUNO, and the three states Clos topology depicted in Fig. 4 to compare NEPTUNO with the proposed online solutions described in Section VII. Table II includes the 5QIs and characteristics used in our setup. The actual committed data rate of each flow was generated using a Gaussian distribution, whose mean is included in the third column of Table II and its standard deviation was set to 15% of the respective mean. Three types of time distributions were taken into account for the flow lifetime and

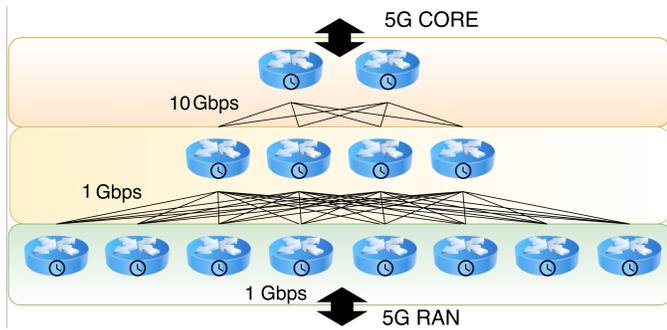


Fig. 4: Backhaul Network (BN) topology considered for comparing NEPTUNO (offline) with the online solutions presented in Section VII.

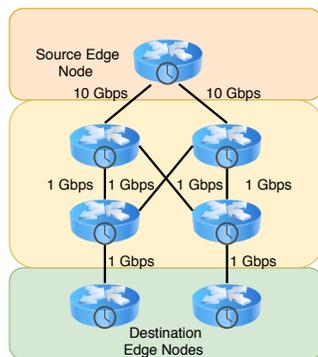


Fig. 5: Backhaul Network (BN) topology considered to evaluate the degree of optimality of NEPTUNO.

the flows inter-arrival times: i) exponential (Exp), ii) Erlang-2 (Erl2), and iii) two-stage hyperexponential (Hyp2). The Erl2 and Hyp2 distributions can respectively approximate any distribution with a coefficient of variation lower and greater than one [37]. In this way, we could assess the performance of NEPTUNO when the arrivals and flow holding processes are not Poissonian. Regarding the reliability, we set the mean time-to-failure for any link to 20 days. Then, only the packets belonging to the delay critical Guaranteed Bit Rate (GBR) flows (e.g., 5QIs 82, 83, 84, and 85) have to be triplicated according to (3) and the flow lifetimes included in Table II. Last, it shall be noted that we use CVX modeling system [41], [42] and Mosek to code and solve the CMINLPs, and Gurobi to solve the MILPs.

B. NEPTUNO Performance Evaluation

We carried out an extensive evaluation of the performance of NEPTUNO using simulation. In every simulation, we simulated the arrival and departure of 800000 different flows. We considered the same flow arrival rate for every 5QI as a simple criterion to generate the 5QI of each simulated flow. Moreover, discrete uniform distributions were used to choose the ingress (source) and egress (destination) nodes for each simulated flow.

1) *NEPTUNO's degree of optimality*: Figure 6 compares the flow rejection ratios offered by NEPTUNO (labeled as “Neptuno”) and by the optimal solution (labeled as “Optimal”) versus the flow arrival rate. We considered the critical

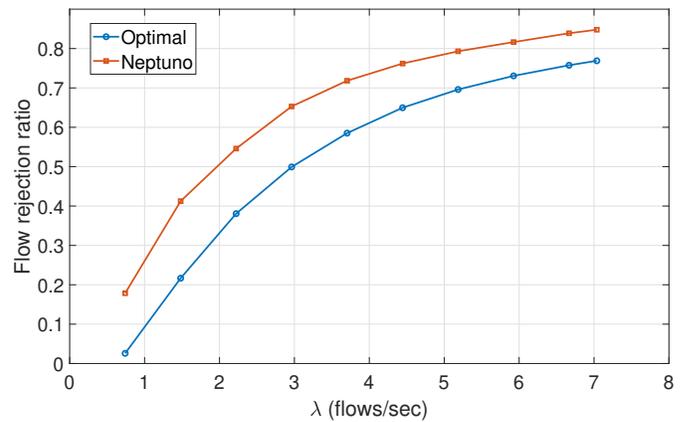


Fig. 6: Degree of optimality of NEPTUNO.

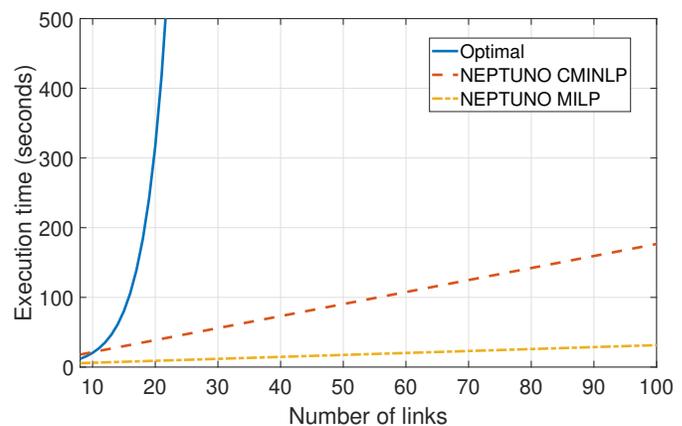


Fig. 7: Computational complexity versus the number of links.

guaranteed bit rate 5QIs (82, 83, 84, and 85) and the network topology shown in Fig 5. The number of shaped buffers and priority levels at each ATS were set to 4. As observed, the flow rejection ratio exhibited by NEPTUNO is roughly 20% above the optimal one for low workloads and 10% above for high workloads. That is because of NEPTUNO’s operation. More precisely, NEPTUNO configures the last hops to minimize the flow rejection probability, whereas the configuration of the transit ATSs is set in such a way that the per-5QI reserved capacity in the last hops can be accommodated. Thus, it seems reasonable that NEPTUNO performs better for high workloads where the configuration of the bottleneck link (last hop) becomes increasingly important.

2) *NEPTUNO's complexity*: We measured the computational complexity of NEPTUNO for both when the per-ATS optimization problem is modeled as a CMINLP (labeled as “NEPTUNO CMINLP” in the figure) and as a MILP (labeled as “NEPTUNO MILP”) as a function of the network scale (i.e., number of ATSs or links), and compared them with the computational complexity of finding the optimal solution (labeled as “optimal”). We used an adaptation of the formulation in [44] of the flow allocation problem to compute the optimal configuration of the whole TSN network as a CMINLP. Figure 7 shows the execution times of NEPTUNO and the optimal flow allocation versus the number of links

TABLE II: Flow types characteristics

5QI	Prio	Rate (Mbps)	Burstiness (bits)	Dmax (ms)	Rmin (%)	Avg. Dur. (s)	Lmax (bits)	Ex. service
1	20	0.064	6400	100	95	130	6400	Conv. voice
2	40	1.5	2250000	150	95	130	10832	Conv. video
3	30	0.1	5000	50	95	1200	5000	Real Time gaming
4	50	0.083	2500	300	95	231	2500	Non-conv. video
5	10	0.01	2040	100	95	130	2040	IMS signaling
7	70	2	10832	100	95	231	10832	Live video
65	7	0.064	6400	75	95	600	6400	MCPTT
67	15	0.5	10832	100	95	300	10832	MCV
69	5	0.01	2040	60	95	600	2040	MCPTT signalling
70	55	0.01	2040	200	95	600	2040	Mission Critical Data
79	65	0.0003	2040	50	95	300	2040	V2X messages
80	68	5	32496	10	95	600	10832	Augmented Reality
82	19	0.1	2040	10	99.999	1200	2040	Discrete Automation
83	22	0.2	10832	10	99.999	1200	10832	Discrete Automation
84	24	0.3	10832	30	99.999	1200	10832	Intelligent transport systems
85	21	0.3	2040	5	99.999	1200	2040	Electricity distribution HV

Internet Multimedia Subsystem (IMS); Mission Critical user plane Push To Talk voice (MCPTT); Mission Critical Video user plane (MCV); Vehicle-to-Everything (V2X); High Voltage (HV). Most of the data included in this table were extracted from [3] and the compound traffic models in [34], [43].

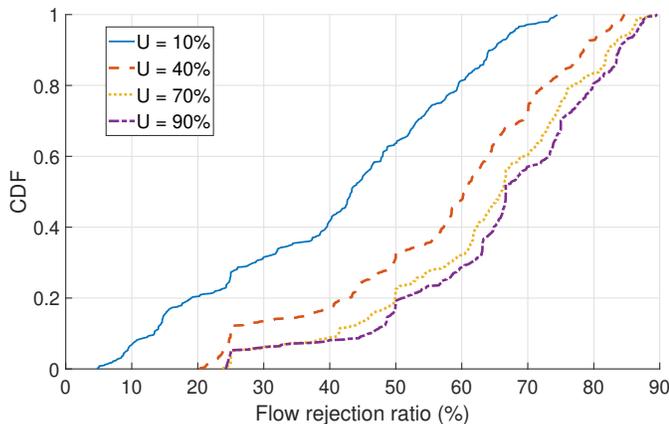


Fig. 8: Empirical cumulative distribution function of the flow rejection ratio exhibited by NEPTUNO considering 4 5QIs and different workload values.

or ATSS in the TSN network for four 5QIs. As observed, the time complexity of the optimal solution shoots up for 22 ATSSs, whereas the NEPTUNO one increases linearly with the number of ATSSs in the network. Last, the reduction in the execution time is remarkable when the per-ATS configuration process is converted to a linear integer model.

3) *5QI Impact On Flow Rejection*: Figure 8 depicts the empirical cumulative distribution functions of the flow rejection ratio offered by NETPUNO in the BN shown in Fig. 4 when there are 4 5QIs and for different workloads. Specifically, to generate them, we performed the randomized selection of 100 groups, each comprising 4 5QIs, and measured the NEPTUNO’s rejection ratio for each of them. As observed, the specific set of 5QIs considered has a great impact on the flow rejection ratio. In other words, the characteristics and performance requisites of the flows matter. For instance, for the same characteristics of the flows (e.g., committed rate and burst size), the rejection ratio increases when the flows’ constraints are more stringent, as can be deduced from (5).

4) *NEPTUNO (offline) Vs Online Baseline Approach*: We compared NEPTUNO with the online solution proposed in Section VII for both objectives (26a) (labeled in the figures as

“Online PD”) and (26b) (labeled in the figures as “Online DM”) in terms of flow rejection ratio. For this purpose, the TSN network topology shown in Fig. 4 was considered. Figures 9a, 9b, and 9c shows the flow rejection ratio offered for each solution as a function of the flow arrival rate for 8, 12, and 16 5QIs, respectively. As observed, NEPTUNO outperforms the online DM solution on almost all the simulated flow arrival rates. This fact is due to the online DM solution does not use DA. Besides, we have observed objective (26b) results in a flatter maximum delay distribution per 5QI. The latter suggests that DM might allocate flows with disparate latency constraints at the same priority level.

As observed, the online PD solution significantly reduces the flow rejection ratio compared to the DM one. These results highlight the importance of DA to maximize the utilization of the ATS-based BNs. Overall, the online PD solution performs better for low workloads, whereas NEPTUNO performs better for high ones for the same reason as previously discussed. The specific 5QI prioritization that results in the minimization of the flow rejection ratio depends on the per 5QI traffic demand characteristics (e.g., flow arrival rate, flow lifetime, average sustainable rate, and burstiness) and QoS requisites. For our setup, which considers a realistic characterization of the 5QIs included in Table II, the objective (26a), which prioritizes the traffic according to its delay/jitter constraint, leads to fair flow rejection ratios as observed in Figs. 9a, 9b, and 9c. Nonetheless, due to the higher flexibility of the online solutions for the flows prioritization, there is, therefore, a reasonable prospect that the online solution presented in Section VII can be enhanced through the inclusion of 5QI prioritization strategies that might outperform NEPTUNO in terms of flow rejection ratio even for high workloads.

In order to assess the robustness of NEPTUNO, we measured its flow rejection ratio considering different distributions to the exponential for the flow inter-arrival and holding times. Specifically, we repeated the set of simulations for the scenario with 16 5QIs but considering the Erl2 and the Hyp2 distributions for those times. The Erl2 distributions were configured to exhibit a coefficient of variation of $\sqrt{0.5}$, whereas the

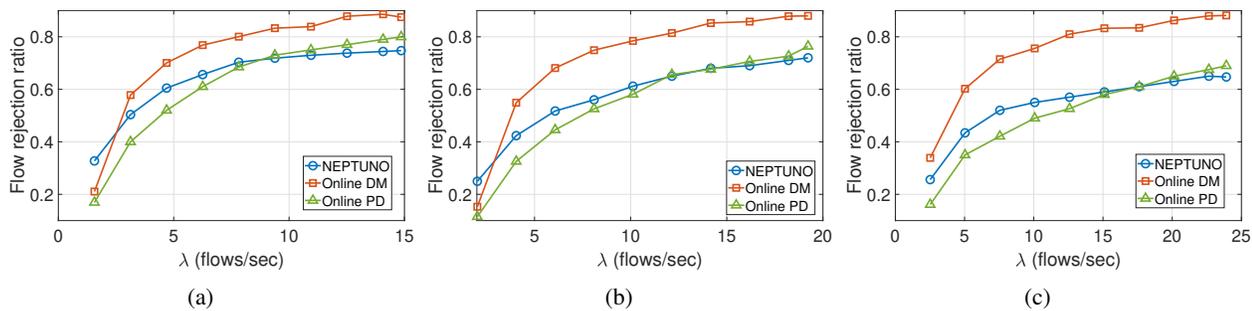


Fig. 9: Comparison between the different solutions in terms of the flow rejection probability for a) eight 5QIs (69, 70, 79, 80, 82, 83, 84, and 85), b) twelve 5QIs (5, 7, 65, 67, 69, 70, 79, 80, 82, 83, 84, and 85), and c) sixteen 5QIs (1, 2, 3, 4, 5, 7, 65, 67, 69, 70, 79, 80, 82, 83, 84, and 85) in the network.

TABLE III: Online solution’s execution times.

P	MILP			Auxiliary Computations		
	3 Hops	5 Hops	7 Hops	3 Hops	5 Hops	7 Hops
8	5.97 ms	5.88 ms	5.69 ms	0.86 ms	1.23 ms	0.91 ms
16	6.16 ms	6.27 ms	5.93 ms	0.88 ms	1.14 ms	0.94 ms
32	7.14 ms	7.02 ms	6.75 ms	1.13 ms	2.19 ms	1.30 ms
64	12.60 ms	11.5 ms	9.96 ms	2.02 ms	3.32 ms	5.59 ms
128	45.38 ms	44.78 ms	30.46 ms	3.12 ms	3.87 ms	4.51 ms
256	288.53 ms	294.25 ms	165.63 ms	18.5 ms	19.02 ms	18.44 ms

Hyp2 distributions were configured to exhibit a coefficient of variation of 1.5. The percentage of the maximum rejection ratio offered by NEPTUNO in both cases was practically overlapped with the results depicted in Fig. 9c. More precisely, the maximum reduction in the NEPTUNO attained profit was 2% and 2.6% for the Erl2 and Hyp2 cases, respectively. These results suggest that the flow arrival and lifetime processes’ impact on NEPTUNO performance is negligible.

Finally, we evaluated the practical aspects of the online solution and compared them with those of NEPTUNO. The key benefits of using offline approaches like NEPTUNO are they allow for lightweight and fast flow access control mechanisms and require less state information.

The first benefit mentioned above is because offline solutions compute a long-term configuration (planning) of the network and the acceptance of an incoming flow requires only to check whether there are enough resources (available link capacities and buffer space) to allocate it. On the contrary, online approaches run an optimization algorithm to decide the configuration of every incoming flow. Table III includes the execution times of the baseline online solution proposed in Section VII. The algorithm’s execution time consists of two parts: i) the auxiliary computations (e.g., compute feasible priorities at every hop and composition of the MILP matrices), and ii) the solving of the MILP (26a)-(26h) using Gurobi. We generated three synthetic Clos topologies with 3, 5, and 7 stages, each with three ATS-based bridges. We also swept the number of shaping buffers and priority levels per ATS, both set to the same value, from 8 to 256. Each runtime measurement included in Table III is the average of 1500 runs. The experimental results show the algorithm exhibits an exponential time complexity with the number of priority levels per egress port. Nonetheless, as observed, the algorithm scales well for up to 64 priority levels per ATS and path lengths of 7 hops. For its part, the measured maximum run time of the flow access control mechanism of NEPTUNO was 125 μ s.

Concerning the memory consumption, we observed that for the network topology depicted in Fig. 4, NEPTUNO needs to store roughly 28% of the online solution state information.

IX. CONCLUSION

This article has addressed the flow scheduling problem in asynchronous Time-Sensitive Networking (TSN) 5G backhaul networks. In this context, we have proposed a novel solution, which is dubbed “Next Generation Transport Network Optimizer” (NEPTUNO), that leverages 5G data analytics framework to maximize the flow acceptance ratio. The solution integrates exact methods and heuristics. We have evaluated the performance of NEPTUNO in terms of the degree of optimality, runtime, and flow rejection ratio. Additionally, we compare it with an online solution also proposed in this work. The online solution integrates analytical methods, heuristics, and data analytics too for optimizing flow allocation. The obtained results show that the flow rejection ratio offered by NEPTUNO is approximately 20% above the optimal one for low workloads and 10% above for high workloads for a TSN BN with 10 links and 4 delay-critical guaranteed bit rate 5QIs. Regarding the runtime, NEPTUNO exhibits a linear computational complexity with the number of links and Asynchronous Traffic Shapers (ATSS) in the network. Finally, the results for the flow rejection ratio evaluation suggest the importance of data analytics to drive the flow allocation optimization process.

ACKNOWLEDGMENT

This work is partially supported by the European Union’s Horizon 2020 research and innovation programme under the MonB5G and 5G-CLARITY projects with grant agreements No. 723172 and 871428, respectively. It is also partially supported by the Academy of Finland Project CSN - under Grant Agreement 311654 and the 6Genesis project under Grant No. 318927. It is also partially supported by the Spanish national research project TRUE5G: PID2019-108713RB-C53.

REFERENCES

- [1] N. Finn, “Time-sensitive and deterministic networking whitepaper,” Huawei Technologies Co. Ltd, Tech. Rep., July 2017.
- [2] A. Nasrallah, A. S. Thyagaturu, Z. Alharbi, C. Wang, X. Shao, M. Reisslein, and H. ElBakoury, “Ultra-low latency (ull) networks: The ieee tsn and ietf detnet standards and related 5g ull research,” *IEEE Communications Surveys Tutorials*, vol. 21, no. 1, pp. 88–145, Firstquarter 2019.

- [3] 3GPP TS23.501 V16.2.0. (2019) System Architecture for the 5G System (5GS) (Release 16).
- [4] E. Grossman, "Deterministic networking use cases," Informational, Internet Engineering Task Force (IETF), Internet-Draft draft-ietf-detnet-use-cases-20, Dec. 2018. [Online]. Available: <https://tools.ietf.org/html/draft-ietf-detnet-use-cases-20>
- [5] "Ieee standard for local and metropolitan area networks – time-sensitive networking for fronthaul," *IEEE Std 802.1CM-2018*, pp. 1–62, June 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)*, Dec 2017, pp. 178–187.
- [7] R. Serna Oliver, S. S. Craciunas, and W. Steiner, "Ieee 802.1qbv gate control list synthesis using array theory encoding," in *2018 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, 2018, pp. 13–24.
- [8] W. Steiner, S. S. Craciunas, and R. S. Oliver, "Traffic planning for time-sensitive communication," *IEEE Communications Standards Magazine*, vol. 2, no. 2, pp. 42–47, JUNE 2018.
- [9] S. S. Craciunas, R. S. Oliver, M. Chmelik, and W. Steiner, "Scheduling real-time communication in ieee 802.1qbv time sensitive networks," in *Proc. of the 24th Int. Conf. on Real-Time Networks and Systems*, ser. RTNS '16. New York, NY, USA: ACM, 2016, p. 183–192.
- [10] R. Serna Oliver, S. S. Craciunas, and W. Steiner, "Ieee 802.1qbv gate control list synthesis using array theory encoding," in *2018 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, 2018, pp. 13–24.
- [11] W. Steiner, S. S. Craciunas, and R. S. Oliver, "Traffic Planning for Time-Sensitive Communication," *IEEE Communications Standards Magazine*, vol. 2, no. 2, pp. 42–47, 2018.
- [12] J. Specht, "Ieee draft standard for local and metropolitan area networks—media access control (mac) bridges and virtual bridged local area networks amendment: Asynchronous traffic shaping," *IEEE P802.1Qcr D*, Nov. 2017.
- [13] 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.
- [14] E. Pateromichelakis *et al.*, "End-to-end data analytics framework for 5g architecture," *IEEE Access*, vol. 7, pp. 40 295–40 312, 2019.
- [15] M. R. Raza *et al.*, "Demonstration of resource orchestration using big data analytics for dynamic slicing in 5g networks," in *2018 Eur. Conf. on Opt. Communication (ECOC)*, 2018, pp. 1–3.
- [16] M. R. Raza, A. Rostami, L. Wosinska, and P. Monti, "A slice admission policy based on big data analytics for multi-tenant 5g networks," *J. of Lightw. Technol.*, vol. 37, no. 7, pp. 1690–1697, 2019.
- [17] I. Afolabi, J. Prados-Garzon, M. Bagaa, T. Taleb, and P. Ameigeiras, "Dynamic resource provisioning of a scalable e2e network slicing orchestration system," *IEEE Transactions on Mobile Computing*, vol. 19, no. 11, pp. 2594–2608, 2020.
- [18] J. Prados-Garzon, P. Ameigeiras, J. J. Ramos-Munoz, J. Navarro-Ortiz, P. Andres-Maldonado, and J. M. Lopez-Soler, "Performance modeling of software-defined network services based on queuing theory with experimental validation," *IEEE Transactions on Mobile Computing*, vol. 20, no. 4, pp. 1558–1573, 2021.
- [19] L. Maile, K.-S. Hielscher, and R. German, "Network calculus results for tsn: An introduction," in *2020 Information Communication Technologies Conference (ICTC)*, 2020, pp. 131–140.
- [20] A. Nasrallah, A. S. Thyagaturu, Z. Alharbi, C. Wang, X. Shao, M. Reisslein, and H. Elbakoury, "Performance comparison of ieee 802.1tsn time aware shaper (tas) and asynchronous traffic shaper (ats)," *IEEE Access*, vol. 7, pp. 44 165–44 181, 2019.
- [21] "Cisco visual networking index: Global mobile data traffic forecast update, 2017–2022," Cisco, Tech. Rep., February 2019, white Paper. [Online]. Available: <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-738429.pdf>
- [22] J. Prados-Garzon and T. Taleb, "Asynchronous time-sensitive networking for 5g backhauling," *IEEE Network*, vol. 35, no. 2, pp. 144–151, 2021.
- [23] 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. [Online]. Available: <https://doi.org/10.1109/TNET.2018.2875191>
- [24] A. Bouillard, L. Jouhet, and E. Thierry, "Tight performance bounds in the worst-case analysis of feed-forward networks," in *2010 Proc. IEEE INFOCOM*, 2010, pp. 1–9.
- [25] E. Mohammadpour, E. Stai, M. Mohiuddin, and J. Le Boudec, "Latency and backlog bounds in time-sensitive networking with credit based shapers and asynchronous traffic shaping," in *2018 30th International Teletraffic Congress (ITC 30)*, vol. 02, Sep. 2018, pp. 1–6.
- [26] Z. Zhou, Y. Yan, M. Berger, and S. Ruepp, "Analysis and modeling of asynchronous traffic shaping in time sensitive networks," in *2018 14th IEEE International Workshop on Factory Communication Systems (WFCS)*, June 2018, pp. 1–4.
- [27] Z. Zhou, M. S. Berger, S. R. Ruepp, and Y. Yan, "Insight into the IEEE 802.1 Qcr Asynchronous Traffic Shaping in Time Sensitive Network," *Advances in Science, Technology and Engineering Systems Journal*, vol. 4, no. 1, pp. 292–301, 2019.
- [28] A. Grigorjew, F. Metzger, T. Hoßfeld, and J. Specht, "A simulation of asynchronous traffic shapers in switched ethernet networks," in *2019 International Conference on Networked Systems (NetSys)*, March 2019, pp. 1–6.
- [29] J. Prados-Garzon, T. Taleb, and M. Bagaa, "Learnnet: Reinforcement learning based flow scheduling for asynchronous deterministic networks," in *ICC 2020 - 2020 IEEE Int. Conf. on Commun. (ICC)*, 2020, pp. 1–6.
- [30] A. Zappone, M. Di Renzo, and M. Debbah, "Wireless networks design in the era of deep learning: Model-based, ai-based, or both?" *IEEE Transactions on Communications*, vol. 67, no. 10, pp. 7331–7376, 2019.
- [31] "Ai and ml - enablers for beyond 5g networks," 5G PPP, Tech. Rep., May 2021, version 1.0 - White Paper. [Online]. Available: <https://5g-ppp.eu/wp-content/uploads/2021/05/AI-MLforNetworks-v1-0.pdf>
- [32] Y. Lee and J. Kaippallimalil. (2019, July) Applicability of ACTN to Support 5G Transport.
- [33] "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.
- [34] J. Prados-Garzon, A. Laghrissi, M. Bagaa, T. Taleb, and J. M. Lopez-Soler, "A complete lte mathematical framework for the network slice planning of the epc," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2019.
- [35] M. J. Sobel, "Simple inequalities for multiserver queues," *Management Science*, vol. 26, no. 9, pp. 951–956, 1980. [Online]. Available: <http://www.jstor.org/stable/2630440>
- [36] A. Harel, "Sharp bounds and simple approximations for the erlang delay and loss formulas," *Management Science*, vol. 34, no. 8, pp. 959–972, 1988. [Online]. Available: <http://www.jstor.org/stable/2632244>
- [37] D. R. Cox and H. D. Miller, *The theory of stochastic processes*. Routledge, 1977.
- [38] W. Ahmad, O. Hasan, U. Pervez, and J. Qadir, "Reliability modeling and analysis of communication networks," *Journal of Network and Computer Applications*, vol. 78, pp. 191 – 215, 2017.
- [39] H. Wang, J. Ding, Y. Li, P. Hui, J. Yuan, and D. Jin, "Characterizing the spatio-temporal inhomogeneity of mobile traffic in large-scale cellular data networks," in *Proceedings of the 7th International Workshop on Hot Topics in Planet-scale mObile Computing and Online Social neTworking*, ser. HOTPOST '15. New York, NY, USA: ACM, 2015, pp. 19–24. [Online]. Available: <http://doi.acm.org/10.1145/2757513.2757518>
- [40] "Ieee draft standard for local and metropolitan area networks — frame replication and elimination for reliability," *IEEE P802.1CB/D2.9, July 2017*, pp. 1–97, 2017.
- [41] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," <http://cvxr.com/cvx>, Mar. 2014.
- [42] —, "Graph implementations for nonsmooth convex programs," in *Recent Advances in Learning and Control*, ser. Lecture Notes in Control and Information Sciences, V. Blondel, S. Boyd, and H. Kimura, Eds. Springer-Verlag Limited, 2008, pp. 95–110.
- [43] J. Prados-Garzon, J. J. Ramos-Munoz, P. Ameigeiras, P. Andres-Maldonado, and J. M. Lopez-Soler, "Modeling and dimensioning of a virtualized mme for 5g mobile networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 5, pp. 4383–4395, 2017.
- [44] J. Prados-Garzon, L. Chinchilla-Romero, P. Ameigeiras, P. Muñoz, and J. M. Lopez-Soler, "Asynchronous time-sensitive networking for industrial networks," in *2021 European Conference on Networks and Communications (EuCNC)*, 2021, pp. 1–6.



Dr. Jonathan Prados-Garzon received his B.Sc., M.Sc., and Ph.D. degrees from the University of Granada (UGR), Granada, Spain, in 2011, 2012, and 2018, respectively. Currently, he is a post-doc researcher at MOSA!C Lab, headed by Prof. Tarik Taleb, and the Department of Communications and Networking of Aalto University (Finland). His research interests include Mobile Broadband Networks, Network Softwarization, and Network Performance Modeling.



Dr. Miloud Bagaa received his Engineer's, Master's, and Ph.D. degrees from the University of Science and Technology Houari Boumediene (USTHB), Algiers, Algeria, in 2005, 2008, and 2014, respectively. From 2009 to 2015, he was a researcher with the Research Center on Scientific and Technical Information (CERIST), Algiers. From 2015 to 2016, he was granted a postdoctoral fellowship from the European Research Consortium for Informatics and Mathematics, and worked at the Norwegian University of Science and Technology, Trondheim, Norway. Currently, he is a senior researcher in Aalto University. His research interests include wireless sensor networks, Internet of Things, 5G wireless communication, security, and networking modeling.



Prof. Tarik Taleb is currently Professor at the School of Electrical Engineering, Aalto University, Finland. He is the founder and director of the MOSA!C Lab (www.mosaic-lab.org). He is also working as part time professor at the Center of Wireless Communications, University of Oulu. Prior to his current academic position, he was working as Senior Researcher and 3GPP Standards Expert at NEC Europe Ltd, Heidelberg, Germany. He was then leading the NEC Europe Labs Team working on R&D projects on carrier cloud platforms, an

important vision of 5G systems. Before joining NEC and till Mar. 2009, he worked as assistant professor at the Graduate School of Information Sciences, Tohoku University, Japan, in a lab fully funded by KDDI. From Oct. 2005 till Mar. 2006, he worked as research fellow at the Intelligent Cosmos Research Institute, Sendai, Japan. He received his B. E degree in Information Engineering with distinction, M.Sc. and Ph.D. degrees in Information Sciences from Tohoku Univ., in 2001, 2003, and 2005, respectively.

Prof. Taleb's research interests lie in the field of architectural enhancements to mobile core networks (particularly 3GPP's), network softwarization & slicing, mobile cloud networking, network function virtualization, software defined networking, mobile multimedia streaming, inter-vehicular communications, and social media networking. Prof. Taleb has been also directly engaged in the development and standardization of the Evolved Packet System as a member of 3GPP's System Architecture working group. Prof. Taleb is a member of the IEEE Communications Society Standardization Program Development Board.

Prof. Taleb is/was on the editorial board of the IEEE Transactions on Wireless Communications, IEEE Wireless Communications Magazine, IEEE Journal on Internet of Things, IEEE Transactions on Vehicular Technology, IEEE Communications Surveys & Tutorials, and a number of Wiley journals.

Prof. Taleb is the recipient of the 2017 IEEE ComSoc Communications Software Technical Achievement Award (Dec. 2017) for his outstanding contributions to network softwarization. He is also the (co-) recipient of the 2017 IEEE Communications Society Fred W. Ellersick Prize (May 2017), the 2009 IEEE ComSoc Asia-Pacific Best Young Researcher award (Jun. 2009), the 2008 TELECOM System Technology Award from the Telecommunications Advancement Foundation (Mar. 2008), the 2007 Funai Foundation Science Promotion Award (Apr. 2007), the 2006 IEEE Computer Society Japan Chapter Young Author Award (Dec. 2006), the Niwa Yasujirou Memorial Award (Feb. 2005), and the Young Researcher's Encouragement Award from the Japan chapter of the IEEE Vehicular Technology Society (VTS) (Oct. 2003). Some of Prof. Taleb's research work have been also awarded best paper awards at prestigious IEEE-flagged conferences.