

Deterministic passive optical networking

KOSTAS CHRISTODOULOPOULOS,^{1,2,*} SARVESH BIDKAR,^{1,3} WOLFRAM LAUTENSCHLAEGE,^{1,4} 
THOMAS PFEIFFER,^{1,5}  AND RENE BONK¹

¹Nokia Bell Labs, Magirusstrasse 8, Stuttgart, 70469, Germany

²Current address: University of Athens, Athens, Greece

³Current address: Nokia Bell Labs, New Jersey, USA

⁴Current address: Baden-Wuerttemberg Cooperative State University (DHBW), Stuttgart, Germany

⁵Current address: Q.ANT, Stuttgart, Germany

*kchristodou@di.uoa.gr

Received 2 August 2023; revised 5 December 2023; accepted 13 December 2023; published 29 January 2024

A time-division-multiplexed (TDM) passive optical network (PON) that behaves like multiple point-to-point (P2P) links can support applications with stringent timing requirements while also maintaining the cost benefits of the point-to-multipoint (P2MP) topology. This requires the TDM-PON to provide deterministic performance: lossless transmission and low and constant (with low variations/jitter) latency to selected flows, while also serving other, e.g., best-effort, traffic flows. We demonstrate two approaches for providing deterministic performance over a TDM-PON. In the first approach, we operate the PON asynchronously to its clients and use scheduling that provides upstream bursts at a low and constant period. Jitter compensator gateways (JC GWs) are used to compensate for the jitter introduced by the non-matching PON bursts and the input traffic cycles. In the latter approach, we implement a deterministic dynamic bandwidth allocation (detDBA) process that controls the placement of bursts at each PON frame for selected transmission containers (T-CONTs) that correspond to flows with given traffic profiles. We synchronize and co-schedule the PON with the rest of the network and applications to provide deterministic performance to those flows. As a use case, we study the communication of industrial applications over the deterministic PON system and demonstrate latency below 100 μ s and sub- μ s packet latency jitter for the industrial flows, while at the same time serving best-effort traffic flows. © 2024 Optica Publishing Group

<https://doi.org/10.1364/JOCN.502227>

1. INTRODUCTION

Time-division-multiplexed (TDM) passive optical networks (PONs) are a mature, standardized, and low-cost technology to provide broadband access to residential users and serve fiber to the home (FTTH) applications in a point-to-multipoint (P2MP) topology. Recently, new use cases for PONs have been proposed and demonstrated, including the passive optical local area network (POL), the transport of 5G data in various functional-split configurations [1–4], tactile Internet [5], and communication in industrial applications [5–9].

Several of the emerging applications are *time-critical*: they require *low latency*, and some require low peak-to-peak (p2p) latency variation. Throughout this paper, we will use the term *jitter* to refer to p2p latency variation. The centralized radio access network (C-RAN) architecture of 5G/6G splits the radio signal processing between the radio unit (RU) and distributed unit (DU). The traffic between the RU and DU needs to be transported over a fronthaul interface with low latency (a few 100s of microseconds) and jitter requirements [10]. In industrial automation, printing, tooling, and packaging machines have control cycles on the order of milliseconds [11,12]. Note

that this refers to bidirectional communication and includes all network layers up to the application, the machine control. Thus, the single-direction transport latency requirement is tighter, while jitter must be substantially lower. According to [12], the most stringent class of industrial applications, the “isochronous real-time,” requires millisecond range of latency and a maximum of 1 μ s of jitter.

Many time-critical applications exhibit known traffic patterns. For example, industrial automation applications typically follow a cyclic communication pattern between a controller and several endpoints. Currently, such applications are served by field buses and networks (e.g., Profinet, Ethercat, Modbus) that use strict scheduling to cater to the cyclic traffic flows. However, such systems lack full interoperability and do not support conventional LAN traffic. The goal is to reduce the cost and improve the efficiency (e.g., through co-optimization) by converging multiple time-critical applications with best-effort (BE) applications onto the same network [13]. For a converged network, IEEE 802.1 Ethernet was extended with a set of time-sensitive networking (TSN) standards that provide various means for temporal control of the transported

flows [14]. These include traffic shaping/engineering, frame pre-emption, synchronization, and scheduling.

In the context of PON, making the P2MP PON behave like multiple point-to-point (P2P) links would support industrial, 5G/6G x-haul, as well as any time-critical application, while also maintaining the cost benefits of the P2MP architecture. To do so, the TDM-PON should provide *deterministic* performance: lossless transmission and low and constant (with low variations/jitter) latency to specific flows, while also serving other, e.g., BE, flows. This requirement applies to both communication directions. The upstream (US) direction, which requires coordination to avoid contentions, is more challenging and will be our focus. Note that WDM PONs offer deterministic performance by isolating time-critical flows in the frequency domain and, thus, creating point-to-point links. However, they are expensive and do not obtain statistical multiplexing gains as TDM-PON. Given that industrial applications typically operate at low rates (a range of tens of Mb/s, converged networks in the Gb/s range capacity), implementing such PON systems on industrial floors is cost-efficient. Thus, in this paper, we discuss the required extensions to TDM-PONs to function as multiple point-to-point links.

To achieve high upstream bandwidth (BW) efficiency and quality of service (QoS) differentiation for FTTH applications, the TDM-PON utilizes a dynamic bandwidth allocation (DBA) process at the optical line terminal (OLT). In upstream, applications are mapped to transmission containers (T-CONTs) with traffic descriptors defining their service requirements. The DBA allocates varying upstream BW to T-CONTs by considering both their traffic descriptors and real-time measurements during operation.

However, for an application with strict latency requirements, e.g., an industrial control process, we would avoid the dynamic nature of DBA to achieve better control over the application's latency. We would provision a T-CONT with a committed information rate (CIR) descriptor matching the application's BW requirement and a delay tolerance (T_{BDT}) descriptor (as defined in G989.3 [15]) matching the application's period, making the DBA operate statically for that

T-CONT. However, when the application period, and thus the configured T_{BDT} , is not a multiple or divisor of the PON frame duration (125 μ s), the arrival of packets would vary across the PON frames. In practice, the DBA does not guarantee the precise placement of US bursts in a frame, but the proper capacity and burst period on average [e.g., as depicted in Fig. 1(a), for a T_{BDT} of 200 μ s, the PON provides five bursts every 1 ms]. Moreover, the allocation of bursts depends on other provisioned T-CONTs and their priorities. So, the PON does not provide bursts exactly every T_{BDT} , which results in relatively high latency and jitter [as shown in Fig. 1(a) for a single T-CONT—without considering additional latency from other T-CONTs].

To reduce the latency and jitter in upstream, we can configure a smaller delay tolerance (T_{BDT}) parameter than the application period, and also keep the burst size, to accommodate the application's packet. In this way we offer additional opportunities to transmit the packet and reduce the latency, but sacrifice the BW efficiency [Fig. 1(b)]. If the burst period is a divisor of the PON frame duration, the starting points of the bursts are fixed in the US PON frames, and we have the advantage that we can repeat the same bandwidth map (BWmap) for each PON frame. We call this fixed bandwidth allocation (FBA), as opposed to DBA. Such an FBA approach with a burst period of a *quarter frame* was used to achieve low latency in [1,16] and discussed in [6]. Figure 2 shows the provisioned BW and the average queuing latency achieved for FBA and different burst periods with respect to a reference flow (1250 bytes every 200 μ s) served over an XGS-PON system. As we reduce the burst period, we achieve lower latency and jitter but increase the overprovisioning. Extension of the FBA solution to multiple flows is straightforward. In [17], Fig. 11 shows an equivalent to the FBA solution with a burst period of 1/8 PON frame for 16 flows (described as an “accelerated DBA”). In FBA, the jitter (p2p latency variation) equals the burst period minus the burst duration [16]. Thus, the maximum latency is lower than the average latency plus the jitter; for a jitter below 1 μ s, which we target, the maximum latency is very close to the average. For a flow of 1250 bytes every 200 μ s

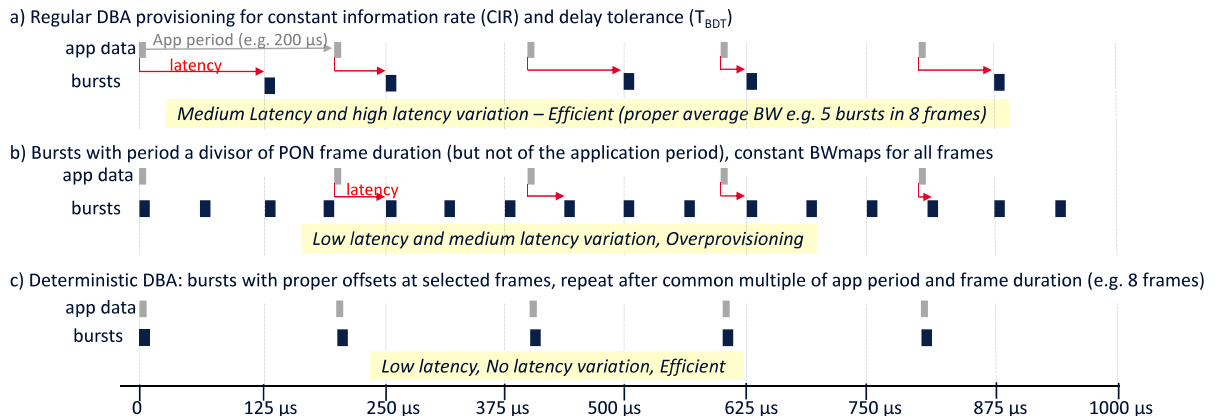


Fig. 1. Options to control the latency of a periodic flow (1250 bytes every 200 μ s in this example) over a TDM-PON: (a) Regular DBA provisioning for a matching CIR and delay tolerance (T_{BDT}); bursts at the start of the upstream PON frame. (b) Bursts with a period that is a divisor of the PON frame duration (half of a frame in this example), which differs from the application period; constant BWmaps for all frames. (c) Deterministic DBA: bursts with proper offsets at selected PON upstream frames; repeat after a common multiple of the application period and PON frame duration (e.g., eight frames in this example).

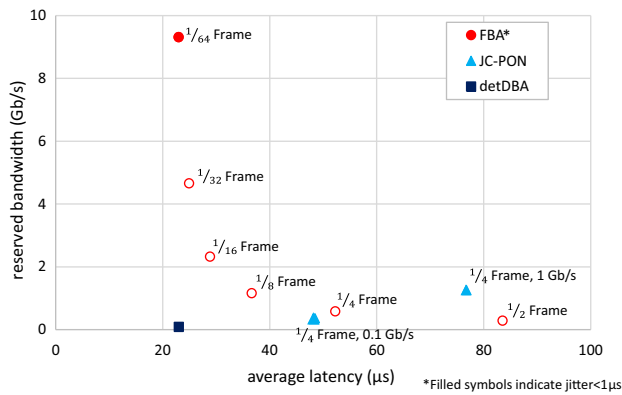


Fig. 2. Reserved bandwidth versus average queueing latency experienced by a periodic flow (1250 bytes every 200 μ s in this example), served with (i) FBA of 1250 bytes and different burst periods that are divisors of the PON frame duration, (ii) proposed JC-PON that utilizes bursts every quarter ($1/4$) of the PON frame, and (iii) proposed detDBA-PON that follows the traffic period. Filled symbols indicate the options that achieve jitter below 1 μ s. Results for option (i) are according to [16] and verified in our PoC demonstrator, while results of proposed solutions, options (ii) and (iii), are from PoC measurements.

(considered in Fig. 2), achieving jitter below 1 μ s requires provisioning for 64 bursts per PON frame, consuming almost 10 Gb/s for a 50 Mb/s flow. In Fig. 2 we depict with filled symbols the options that achieve such low jitter.

Since such BW inefficiency is unacceptable, we need to use fewer bursts per frame. However, due to the mismatch between the application packets and PON burst periods, the packets experience jitter, which can disrupt the time-critical applications. When such jitter is unacceptable, we can use jitter compensator gateways (JC GWs). This approach relies on FBA with the addition of JC GWs, which translate jitter to latency so that all packets experience the highest latency. Various mechanisms exist to create such JC GWs. The authors of [3] proposed a JC that relies on timestamps [at the optical network units (ONUs)], synchronization, and buffering at the OLT where packets are selectively delayed to achieve a constant latency.

We proposed in [8] (also detailed in Section 3) JC GWs that create a tunnel from an ONU user network interface (UNI) to an OLT service node interface (SNI). The tunnel provides known and constant latency [measured jitter below 200 ns in our proof-of-concept (PoC) demonstrator—see Section 5]. This jitter-compensated (JC)-PON does not require synchronization of the PON with the external network and the applications, and it allows the PON to have its own burst period (e.g., quarter frame). However, it overprovisions for the full line rate of the UNI—the client Ethernet port (e.g., 1 Gb/s)—and exhibits latency higher than the minimum (the increase is equal to the PON burst period). Figure 2 shows the provisioned BW and achieved average latency for the developed JC-PON solution and two provisioning options: 1 Gb/s (implemented) and 100 Mb/s.

An alternative solution is to follow the application traffic pattern with PON bursts [Fig. 1(c)]. In this direction, we proposed in [9] (also detailed in Section 4) a deterministic

DBA (detDBA) that controls the placement of bursts at each upstream frame for selected T-CONTs, providing them with deterministic performance. For a periodic application, we can synchronize the PON with the application and follow the application period. Thus, we can achieve the lowest possible (scheduling) latency and low jitter (measured below 230 ns in our PoC demonstrator) and consume only the required BW. Serving multiple time-critical applications requires synchronization and co-scheduling, which typically requires their convergence over, e.g., a TSN-IEEE 802.1Qbv-based network [14]. The detDBA can coexist with a regular DBA (regDBA) to serve both time-critical and best-effort traffic. Our detDBA draws ideas from the Co-DBA concept, which is standardized in [18], and the TSN-IEEE 802.1Qbv standard [14] to provide deterministic latency to multiple flows with different periods and packet sizes.

As a use case, we apply the two proposed solutions, the JC-PON and the detDBA-PON, to serve industrial flows from devices that are either directly connected to the PON or served over an Ethernet TSN IEEE 802.1Qbv network, in which case we interface the PON with the TSN switches. For the JC-PON solution, we demonstrate low latency and jitter, without the need of synchronizing and co-scheduling the PON with the TSN network and the applications. For the detDBA solution we make the PON part of the boundary clock distribution tree, and co-schedule it to demonstrate even lower latency (the lowest achievable in our setup), similar jitter level with the JC-PON, and high BW efficiency. In all cases, we simultaneously serve best-effort traffic flows without interfering with the industrial/time-critical flows.

This paper is an extended version of our previous works [8,9]. Compared to those, in this paper we provide a generalized explanation of the deterministic PON concept; we describe the two proposed solutions in more detail and discuss their advantages/disadvantages. We also present additional experimental results, evaluating the performance of these solutions independently as well as in comparison.

The rest of the paper is organized as follows. In Section 2 we discuss the related work. In Section 3 we describe the JC-PON, while in Section 4 we present the detDBA-PON. The detailed measurement setup and results from our proof-of-concept demonstrator are discussed in Section 5. Finally, Section 6 concludes this work.

2. RELATED WORK

There are several technologies to control the latency and jitter of the traffic flows. On one hand, we have specialized network technologies, referred to as field-buses and field-networks, developed for real-time control of machines and production, also referred to as operational technology (OT). Due to the wide variety of OT applications and varied requirements, several field-buses have appeared [19] (Profibus, Profinet, Ethercat, Industrial Protocol, etc). Different field-buses are generally not compatible with each other, as opposed to information technology (IT)-based networks relying on IP and Ethernet. To improve interoperability, field-buses are moving towards an Ethernet-based sublayer. However, their speed and interoperability are still limited.

On the other hand, in the IT world, several techniques have been developed that rely on traffic engineering to control the queueing jitter. In such approaches the control is done by admission control at the network borders, by traffic shaping, or by statistical calculations, using, e.g., network calculus [20]. However, the achieved latency and jitter bounds might turn out to be infeasible in real cases. For example, Ethernet IEEE 802.1Qcr [21] provides a formula that gives a 12 μ s collision duration as worst-case for a 1 Gb/s link, for any traffic load. However, this value is already higher than the synchronization and jitter requirement of certain time-critical applications.

Other TSN standards to control the latency and the jitter are the frame preemption (IEEE 802.1Qbu and 802.3br) and scheduled traffic (IEEE 802.1Qbv) [14]. Frame preemption enables a higher priority frame to interrupt an ongoing transmission of a lower priority frame (otherwise, it would wait for its full transmission). Pre-emption solves the collision problem of one high-priority class over all others, but not within the high-priority class itself or when several classes require low latency, as the use cases we target.

Scheduled traffic (IEEE 802.1Qbv) employs gates in front of each priority queue at the egress ports of the Ethernet switches that open and close according to a global schedule. Global scheduling requires synchronization, which is achieved through the IEEE 802.1AS [22] profile of the precision time protocol (PTP), which requires “any two time-aware devices separated by seven or fewer hops to be synchronized within 1 μ s peak-to-peak accuracy.” The schedule defines for each TSN Qbv switch a cyclically executed gate control list. The gate opening offset (w.r.t. the TSN switch cycle) is calculated by the transmission time at the previous switch, as well as the propagation and switching time of the considered switch. Flow isolation is achieved in time (delaying at a previous hop) and/or space (using different queues). The gate opening duration is calculated based on packet size, while synchronization and propagation errors are accounted for by adding a margin to that. Typically, one queue is allocated to BE/IT traffic, which is served during inactive/unscheduled periods of time-critical traffic. IEEE 802.1Qbv isolates the time-critical flows, and maintains their cycles, yielding constant latency and low jitter. However, it requires global synchronization and the diversity in paths and the variety of packet sizes, and cycles make the scheduling calculation very difficult (NP-hard [23,24]). Thus, although it provides the ultimate control over latency it is hard to scale to big networks and multiple applications.

Therefore, [25] proposed a backbone network for multiple independent TSN networks. The proposed backbone relaxes the key constraints of Qbv, i.e., it avoids global synchronization and scheduling, enabling the formation of TSN domains that stretch out over different locations and keep their heterogeneity (protocol, maintenance, etc.). The backbone is based on a new TDM architecture implemented as a shim layer in the Ethernet protocol stack. In this paper, we make use of that solution in the JC-PON described in Section 3.

Note that the above discussion pertains to layer 2 solutions (as is the PON) for deterministic latency. The IETF Deterministic Networking (DetNet) working group [26] is defining a layer 3 umbrella across layer 2 domains, excluding layer 2 modifications, which is our focus here.

In this paper, we consider TDM-PON, a telecom technology that has been massively deployed for serving residential users and FTTH applications. These are best-effort applications that do not have strict latency and jitter requirements, and as so, the PON was not developed with such a focus. However, the PON has the key features to grant low latency and low jitter. First of all, it is scheduled in the upstream direction, while in downstream there is a single outgoing port and known solutions are rightly applicable. Secondly, the PON can be synchronized to an external source and can provide synchronization as a service. Note that these two are the ingredients used in TSN Qbv and field-buses to achieve low latency and jitter. Moreover, PON is a point-to-multipoint system that is mature and cost-efficient. Most applications, even those requiring low latency and jitter follow a server-to-multiple-clients communication paradigm: industrial applications involve a controller and several end-devices, and x-haul communication is also point-to-multipoint, involving communication between a CU and multiple DUs and between a DU and multiple RUs. Converged industrial networks are envisioned to use Ethernet 802.1Qbv switches but such switches today have a low radix (four ports) and low rate (1 Gb/s). Thus, building a converged network requires a plethora of switches, which complicates scheduling and limits the scalability. Thus, PON has the key features and certain advantages over existing technologies (higher rate, high radix) and can replace or interface with existing network elements to implement a converged network. However, since the PON was designed for FTTH applications it requires extensions for deterministic performance. In this paper, we propose two such extensions.

Making use of the TDM-PON advantages, recent works propose it for various non-FTTH use cases [5,7] including the support of time-critical applications. The transport of various 5G functional-splits is considered in [1–5,16]. The works [1,5,16] rely on providing bursts at a constant burst period/FBA [Figs. 1(b) and 2(i)] that experience relatively high jitter, at an acceptable level for front- and mid-haul. The co-scheduling of the PON with the radio access network (RAN) according to the Co-DBA concept [18] is followed in [2,3]. The solution of [3] employs JC GWs based on time-stamping, an alternative technique to tunneling that we proposed in [8] and in Section 3 of this paper. An adaptive FBA based on feedback to identify the setup and tear-down of front-haul connections is proposed in [4]. A fast DBA based on P4 to quickly allocate US bursts to time-critical traffic was demonstrated in [27]. However, [27] relies on DBRu reports that in turn rely on previous BW allocations, which can cause high latency and jitter. Most of these aforementioned approaches neglect jitter, which is crucial for industrial applications [6,7,12]. The authors of [28] propose a time-aware DBA to achieve low latency and jitter, similar to the detDBA concept we proposed in [9] and in Section 4 of this paper. The focus in [28] is on the related scheduling algorithm and results come from simulations, while in this paper we describe a PoC demonstrator, where the PON is interfaced with TSN Qbv switches and we present experimental measurements.

3. JITTER-COMPENSATED PON

In this section, we describe the solution developed to achieve deterministic performance for upstream transmission in PON using jitter compensation (JC) techniques. In [1], a low-latency MAC was demonstrated, which served an upstream flow with multiple periodic bursts per PON frame. Through such a mechanism, the PON can be configured to have burst periods that are an integer multiple or divisor of the PON frame duration (i.e., 125 μ s). Targeting low latency, we would select a burst period that is a divisor of the PON frame duration, for example, in [1] a burst period of 31.25 μ s (quarter frame) was used, which results in four bursts per PON frame to grant low latency to CPRI-based fronthaul traffic. In such a case, where we use a burst period that is a divisor of the PON frame duration, a key advantage is that we can repeat the same BWmap at each PON frame for the time-critical flows. Assuming a periodic application traffic, with a period that is an integer multiple or divisor of the PON frame duration, the application can be supported with low latency and low jitter with such a scheme. However, for generic cases of a mismatch between the application and PON burst periods causes a jitter that could disrupt the time-critical application. Moreover, rate conversions, the PON operating at a higher rate (e.g., 10 Gb/s) compared to the UNI and/or SNI ports (e.g., 1 Gb/s), especially rate down-conversion, create self-induced collisions that alter the timing of the packets, thus creating jitter.

To solve these issues and reduce the jitter, we enhanced the PON with JC, extending the approach described in [25]. In the following we describe the proposed solution in the upstream direction and for our PoC demonstrator that consists of an XGS-PON and 10 GE (10 Gb/s) ONU UNI and OLT SNI ports. However, our solution is quite generic and can be extended for other TDM-PON generations and port rates.

We developed JC GWs in FPGA boards. The JC GWs have GE (1 Gb/s) ingress and egress ports, targeting to be interfaced with TSN switches and/or industrial applications at that port rate and to function as rate converters towards the XGS-PON. The JC GW at the ONU side periodically, every PON burst, i.e., every 31.25 μ s in the PoC discussed in the following, encapsulates the GE bitstream (including idle symbols) arrived during the previous period at the JC GW ingress port in a container Ethernet frame. This frame is then passed to the ONU and transmitted as a PON burst. The PON is configured with the low-latency MAC, i.e., four bursts every PON frame, which is with a burst period of 31.25 μ s (quarter frame). For

each T-CONT that has an actual maximum rate lower than the free (unreserved) slots. The wait we reserve a CIR fixed bandwidth of 1 Gb/s (actually slightly higher to accommodate the added Ethernet header). The burst reaches the OLT and then the encapsulated container Ethernet frame is forwarded to the egress JC GW, which decapsulates it and forwards the initial GE bitstream to the dedicated GE JC GW egress port. Figure 3 shows a schematic depicting the processing performed at each step of the JC-PON in the upstream transmission direction.

Under the hood, the JC GWs at the ONUs form a logical bus, so that we use the synchronous bus architecture that is described in [25]. The link between the JC GWs at the ONUs and the JC GW at the OLT is partitioned into slots that follow a cyclic schedule that can be reserved for connections between the client ports, according to the required granularity. For example, we can define a cycle of 100 slots on a 10 Gb/s link, to grant 100 Mb/s granularity. The bitstream of a T-CONT that requires deterministic performance is aggregated into containers that are forwarded solely to the corresponding reserved slots and are thus isolated from other traffic. The bitstream of a best-effort T-CONT is aggregated into containers that have opportunistic access to the free (unreserved) slots. The waiting time of the packets of a deterministic T-CONT equals the reservation cycle (the bitstream is aggregated from the reserved slot in one cycle until that slot in the next cycle), assuming a reservation of one slot per cycle. With reservations and periodic scheduling, the packets of the deterministic T-CONTs are isolated and experience constant latency that does not depend on the traffic of the other flows/slots.

In the JC-PON we also align (i) the JC GW at the ONUs and the OLT and (ii) the ingress and egress port clocks. These alignments are done inband with byte stuffing without using any external synchronization mechanism. Details about these alignment processes are given in [25]. In the PoC we implement a phase-locked loop (PLL) between the JC GWs and the PON to align the generation of the encapsulated Ethernet frames with the PON bursts. The JC GWs were implemented as external modules in FPGAs in our PoC demonstrator, but they are envisioned to be integrated with the ONU and OLT. Thus, such a mechanism for the alignment of the ONUs and OLT JC GWs will not be required in the integrated JC-PON solution.

The JC-PON solution does not require synchronization of the PON with the external network and the applications (the JC GWs even tunnel the PTP messages), and it allows

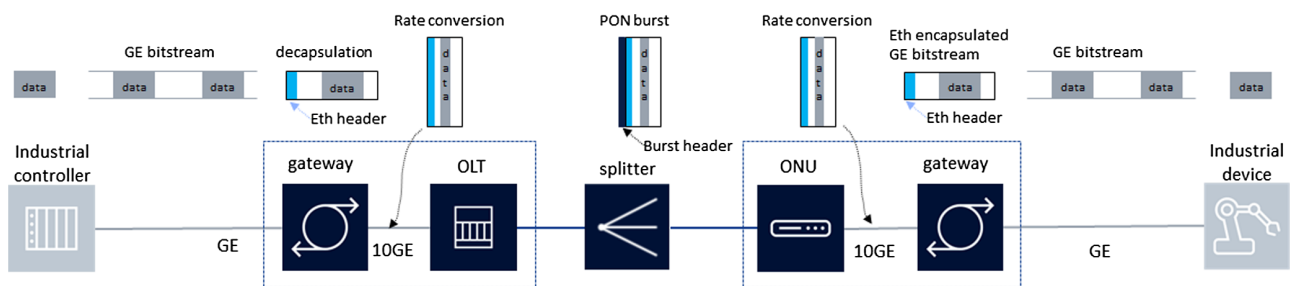


Fig. 3. Upstream transmission of a periodic flow over the JC-PON. The ONU JC GW encapsulates the GE bitstream arriving at the UNI port into a container Ethernet frame. Note that even the idle symbols are encapsulated. The container Ethernet frame is transported as a PON burst. The OLT decapsulates this Ethernet frame and forwards the initial Ethernet GE bitstream to the SNI port.

the PON to have its own schedule and burst cycle. The tunnel created between the JC GW ports transfers the Ethernet bitstream from the ingress to the egress port: all bitstream symbols, including symbols for the packets but also idles, are transferred *as is*. Figure 4 is an example that shows the timing for the JC-PON. Any input traffic, periodic and non-periodic or even random, maintains unaltered the timing between the packets. This creates a tunnel from an ONU UNI to an OLT SNI (assuming the JC-PON with integrated JC GW into the OLT and ONUs) and yields known latency. The latency is equal to the PON/JC GWs reservation cycle plus the PON propagation and is constant (measured jitter below 200 ns in our PoC demonstrator—see Subsection 5.B). Thus, the JC-PON behaves as if there was a “cable,” a P2P link, between the two ports.

The JC-PON that behaves as a P2P link can be easily interfaced with other networks or applications. For example, an IEEE 802.1Qbv TSN network is based on scheduling calculations that assume that the TSN switches are connected with links of known propagation delay (with known and small uncertainty). Moreover, a requirement for the TSN network is that the networking infrastructure (bridges or end-devices in PTP terminology) support PTP with the required accuracy of 1 μ s as defined in the 802.1AS-2011 [22]. The JC-PON does not affect the scheduling calculations of the TSN network since it behaves as a link of known propagation delay, while it also transparently transports the PTP messages to enable synchronization of the devices at its edges.

A similar solution with encapsulation and decapsulation at the JC GWs is applied in the downstream direction. The solution in downstream can be simpler as there is no need to match the time-critical flow and the PON burst cycles in that direction. Finally, note that in TDM-PON there is typically an upstream-downstream latency asymmetry. Thus, the JC-PON actually acts as an asymmetric link. The known latency and asymmetry can be accounted for in PTP configuration, in the TSN scheduling, and/or in the application configurations.

4. DETERMINISTIC DBA PON

The approach to deterministic PON described in Section 3 does not require any knowledge of the served traffic. However, in the detDBA-PON approach, we assume that the T-CONTs that require deterministic performance and relevant parameters about their traffic patterns can be identified in advance. For

example, for a periodic flow, we need to know its packet size, period, and starting time. Some of these parameters can be passed through the regular PON traffic descriptors during provisioning of the T-CONTs and/or require extensions of the traffic descriptions or a new interface, which are left outside of this work. An example of a real-time control plane interface for provisioning T-CONT parameters in the context of mobile fronthaul is the co-operative transport interface (CTI) [29] specified by O-RAN. Such input can be processed by a cooperative (Co-) DBA [18] in the OLT.

In this paper, we assume that a set of periodic flows requiring deterministic performance and their relevant parameters is available to the DBA process within the OLT. In the following explanation, D is the set of deterministic T-CONTs, T_d is the period, S_d is the packet size, and A_d is the arrival time of the deterministic T-CONT $d \in D$. The duration D_d of the bursts of T-CONT d is calculated based on the size of the packets S_d , assuming one burst per packet ($D_d = S_d/R$, where R is the PON upstream line rate).

The proposed detDBA-PON consists of a deterministic scheduler and an enhanced BWmapper, which complies with the Co-DBA concept. We will discuss the BWmapper at the end of this section. For the set of deterministic T-CONTs, the deterministic scheduler finds the starting times of the respective bursts by solving the *non-pre-emptive scheduling of concrete periodic tasks problem* [23,24,30]. The term *non-pre-emptive* indicates that we do not break a packet into multiple bursts, the term *idling* indicates that we do not need to transmit the packet immediately even if the PON channel is free but we can store it (at the ONU) until its burst is scheduled, and the term *concrete* indicates that we know the arrival times of the packets (they depend on the application configuration and any network element before reaching the ONU). We calculate the schedule for the *hyperperiod* of

$$T_{\text{detPON}} = \text{LCM}(\{T_d \text{ for all } d \in D, T_F\}),$$

where $\text{LCM}()$ is the least common multiple of the set $\{\}$ that consists of the periods T_d of the deterministic traffic T-CONTs $d \in D$ and of the PON frame duration ($T_F = 125 \mu$ s). The calculated schedule is repeated every T_{detPON} . We convert the packet arrival times of all deterministic T-CONTs with respect to a reference PON frame, with the following process: take the earliest packet arrival time across all deterministic T-CONTs ($A_{\text{ref}} = \min_{d \in D}(A_d)$) and find the PON frame that contains A_{ref} . The start of that PON frame T_{ref} is used as the reference

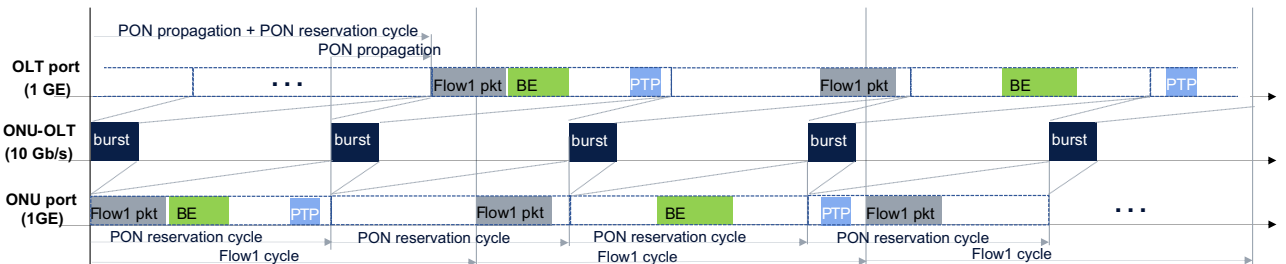


Fig. 4. Example that shows the timing for the JC-PON solution. The reservation at the PON is the full port ONU UNI rate (e.g., 1 GE in this example). All packets arriving at the ONU UNI port are tunneled to the OLT SNI port *as is*, without altering their timings. This solution can support applications with periods that do not match the PON reservation period, even aperiodic flows, best-effort traffic, PTP, or other control packets without interfering with them. It grants to all constant latency, so the PON behaves as a “cable,” a P2P link.

starting time. The (virtual) packet arrival times of the other deterministic T-CONTs are calculated by subtracting their period (T_d) from their arrival time (A_d) until we find the first (virtual) arrival of their packet after the reference PON frame T_{ref} . The schedule is calculated from T_{ref} for the calculated (virtual) starting times, for duration T_{detPON} , and repeated every T_{detPON} afterward. This process is required to avoid all time conflicts, since all possible conflicts will appear within that T_{detPON} . The actual reservations could be delayed until the respective traffic arrives.

For each deterministic T-CONT d , the schedule defines a burst offset o_d . The offset is applied at each period, defining thus the burst starting times over T_{detPON} . In particular, the bursts for T-CONT d start at $o_d + A_d + i \cdot T_d$, for $i = 0, 1, \dots, T_{\text{detPON}}/T_d$. The offsets are calculated to avoid collisions among bursts of different deterministic T-CONTs within the T_{detPON} . The duration between $o_d + A_d + i \cdot T_d$ and $o_d + A_d + i \cdot T_d + D_d$, for $i = 0, 1, \dots, T_{\text{detPON}}/T_d$ is reserved for bursts of T-CONT d exclusively. Figure 5 shows an example of the scheduling of two deterministic T-CONTs with periods 200 and 250 μs , over the hyperperiod T_{detPON} of 1 ms.

The offset o_d is equivalent to the scheduling (queueing) latency of the packet of T-CONT d at the ONU buffer until it is served by its allocated burst. Since a single offset is calculated per T-CONT and applied at the T-CONT period, all T-CONT packets exhibit fixed scheduling latency (zero scheduling jitter). This holds for flow periods that are multiples of the PON word duration (12.86 ns in XGS-PON), which is the lowest PON scheduling granularity. Otherwise, if the period is not a multiple of the PON word duration, there will be a very low scheduling jitter (< 1 PON word duration).

The scheduling problem for the detDBA-PON, described above, is complex (NP-hard) even if it is for a single hop [24]. It cannot be solved optimally in a reasonable time for a large number of deterministic flows. However, for a low number (e.g., tens to hundreds) of flows, the shortest period first [which

maps to the earliest deadline first (EDF) assuming that the deadline is the period] heuristic achieves good performance. It is worth noting that our detDBA proposal makes the PON equivalent to a switch in a TSN Qbv-based network, which requires such scheduling calculations [23]. Moreover, we envision using such a system quasi-statically, where the addition of new flows is not done dynamically and remains for long periods, as would be the case on an industrial floor or a 5G/6G functional pipeline. The scheduler will run when a deterministic T-CONT is provisioned, which could be done well ahead of the first packet arrival. Finally, if the start of the application can be controlled and we can schedule all network elements within the path, then the problem becomes that of scheduling *non-concrete* tasks, which can be solved optimally with the EDF heuristic [30]. Evaluation of the scheduling performance of the detDBA-PON solution requires detailed simulations and is part of our future plans.

Depending on the other network elements connected to the PON and the applications served, the PON needs to be synchronized and co-scheduled to yield deterministic performance. To be more specific, assuming multiple applications, they should be globally synchronized to be served over the detDBA-PON, as would be the case if they were served over a converged TSN IEEE 802.1 Qbv-based network. In such a case, the TSN switches would be synchronized and distribute the clock by forming a boundary clock distribution tree using the IEEE 802.1AS PTP profile [22]. Since commercial PON systems support PTP, the PON can be part of the same clock distribution tree. The proposed detDBA-PON is equivalent to one TSN switch with ingress ports (UNIs) at the ONUs and egress ports (SNIs) at the OLT, thus enabling it to be a part of the TSN network scheduling.

In the proposed solution, the detDBA and the regDBA processes coexist at the OLT (Fig. 6). The scheduler calculates the schedule (starting times and durations of the bursts) for the deterministic T-CONTs at provisioning time. The regDBA works in cycles of duration C . The schedule duration

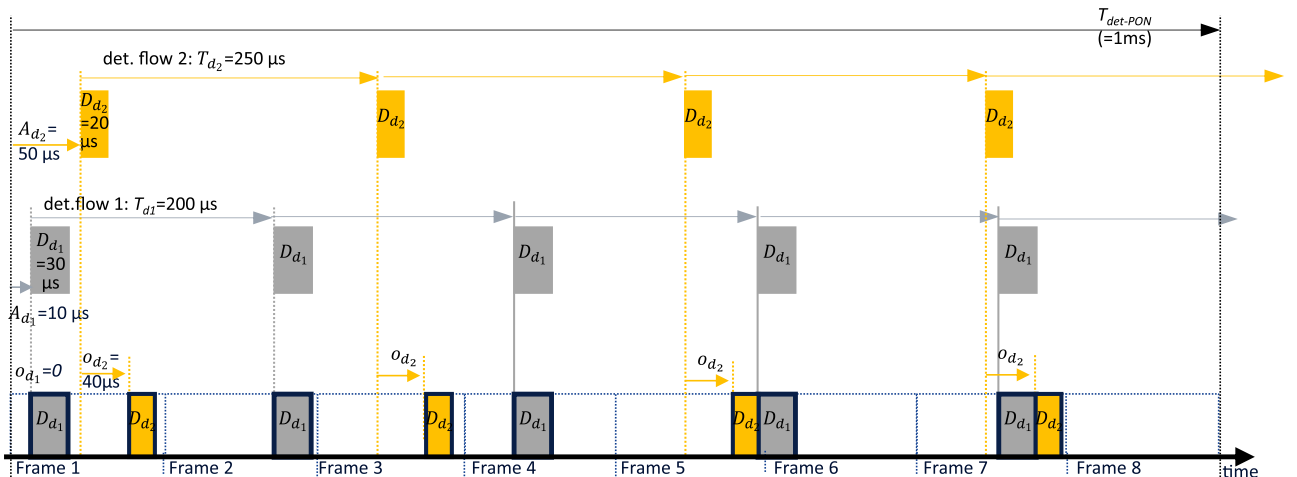


Fig. 5. Scheduling two flows d_1 and d_2 . Flow d_1 has period $T_{d_1} = 200 \mu\text{s}$, duration (in the PON) $D_{d_1} = 30 \mu\text{s}$, and arrival time $A_{d_1} = 10 \mu\text{s}$ with respect to the start of the first PON frame. Flow d_2 has period $T_{d_2} = 250 \mu\text{s}$, duration (in the PON) $D_{d_2} = 20 \mu\text{s}$, and arrival time $A_{d_2} = 50 \mu\text{s}$ with respect to the start of the first PON frame. The schedule is calculated for hyperperiod T_{detPON} , the LCM of the PON frame duration $T_F = 125 \mu\text{s}$, and of the periods of the two flows, that is, for $T_{\text{detPON}} = 1 \text{ ms}$. The schedule identifies the offsets $o_{d_1} = 0$ and $o_{d_2} = 40 \mu\text{s}$ so that bursts do not collide in the hyperperiod T_{detPON} . This schedule is repeated every hyperperiod T_{detPON} .

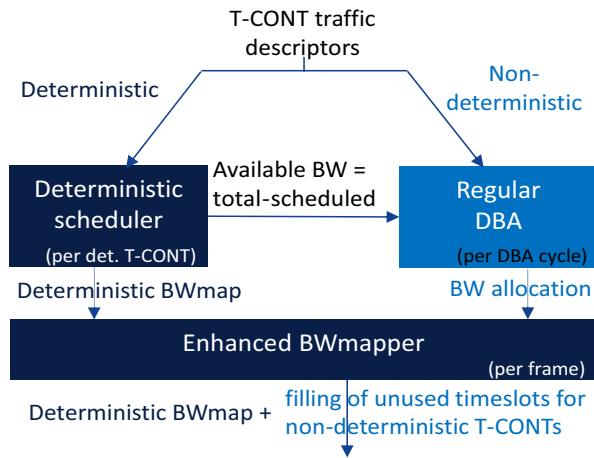


Fig. 6. Schematic of the detDBA process, including the deterministic scheduler, the regular DBA, and the enhanced BWmapper.

T_{detPON} can be larger or smaller than C . So at each start of a regDBA cycle, the scheduler informs the regDBA of the BW it has allocated in that cycle so that the regDBA can allocate the remaining BW to the non-deterministic T-CONTs. The scheduler transfers the schedule (in the form of deterministic BWmaps) to the enhanced BWmapper, which runs per PON frame. The enhanced BWmapper applies the deterministic BWmap part for the frame and fills the unused timeslots with bursts for the non-deterministic T-CONTs according to the calculations of the regDBA.

Considering the previous example with the scheduling of two flows depicted in Fig. 5, and assuming a regDBA with a cycle of $C = 2$ ms, the detDBA would inform regDBA that it has allocated $(5 \cdot D_{d1} + 4 \cdot D_{d2}) / R$ bandwidth to the deterministic T-CONTs, where R is the upstream PON line rate. The regDBA could allocate the rest of the R to non-deterministic T-CONTs. The enhanced BWmapper would get the deterministic schedule for each frame (or for the whole 1 ms cycle as shown in Fig. 5 and would apply it per frame) while allocating the free timeslots (white parts of the frames in Fig. 5) to non-deterministic T-CONTs based on the BW allocation provided by regDBA. Note that the above exemplified calculations exclude FEC and burst overheads.

The detDBA solution applies specifically in the upstream direction and the TDM-PON transmission convergence (TC) layer that controls the optical transmission between ONUs and the OLT. Typically, several OLTs are integrated into a *multi-OLT system* that serves several PONs and includes one or more layers of (electrical) packet switches. Even the ONUs can have multiple UNIs and a small aggregation switch before the optical PON interface. Therefore, the latency performance of the complete PON system, including the electrical packet aggregation switches at the OLT or ONU, depends on the determinism of those switches. The same applies in the downstream direction where there is no PON scheduling and packet multiplexing is performed according to the traffic management capabilities of the packet switches in the multi-OLT system. Thus, to achieve deterministic performance across the whole PON system, in addition to the detDBA solution outlined above, the packet switches should comply with TSN IEEE

802.1Qbv or any TSN flavor that satisfies the latency and jitter requirements of the applications.

5. DETERMINISTIC PON PROOF-OF-CONCEPT DEMONSTRATION

To demonstrate the deterministic performance of the two proposed PON solutions, we implemented a converged industrial network testbed. Our testbed consisted of a Nokia XGS-PON (ISAM 7360) system implementing the OLT, $2 \times$ Nokia XGS-PON ONUs, $2 \times$ TSN IEEE 802.Qbv switches (Hirschmann RSPE35), and an Ethernet traffic generator and analyzer (IXIA Novus One). In the case of the JC-PON, we used $3 \times$ Xilinx Virtex-7 FPGA boards to implement the JC GWs. Figure 7 shows a photo of our testbed.

In our performance evaluation, we first characterized the TSN network (Subsection 5.A) and then the two deterministic PON solutions (JC and detDBA) independently without connecting the TSN switches (Subsections 5.B and 5.C, respectively). Then we measured the performance of the full network with the TSN switches connected using the JC-PON or the detDBA-PON solutions (Subsection 5.D).

A. TSN Network Characterization

For reference, we initially measured the latency and peak-to-peak variation (jitter) for a network consisting of two TSN switches alone.

Figure 8 shows the logical representation of the setup. The IEEE 802.1AS PTP profile was used with the industrial controller being the server while the TSN switches participated in the boundary clock distribution tree as shown in the figure. We synthesized two industrial flows: flow 1 was configured to generate 1250 byte packets with a period of $200 \mu\text{s}$, and flow 2 was configured to generate 625 byte packets with a period of $250 \mu\text{s}$. Note that we intentionally created industrial flows with different periods, to validate the convergence capability of IEEE 802.Qbv TSN technology to support flows with different periods over the same network. We also created a BE traffic flow with randomly sized packets (64–1512 bytes) and a 300 Mb/s (average) rate. Traffic was bidirectional and measurements include both the US and downstream (DS) directions.

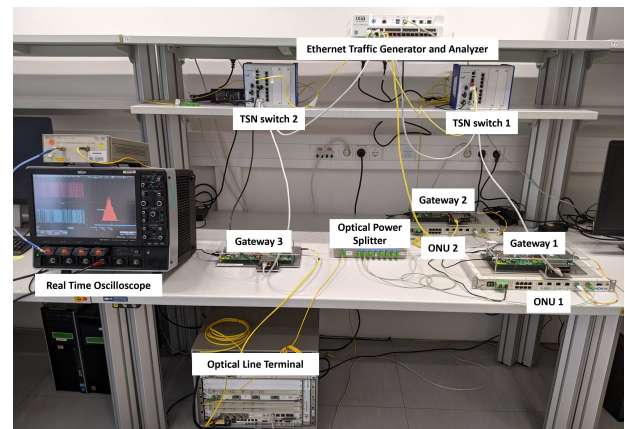


Fig. 7. PoC converged industrial network testbed.

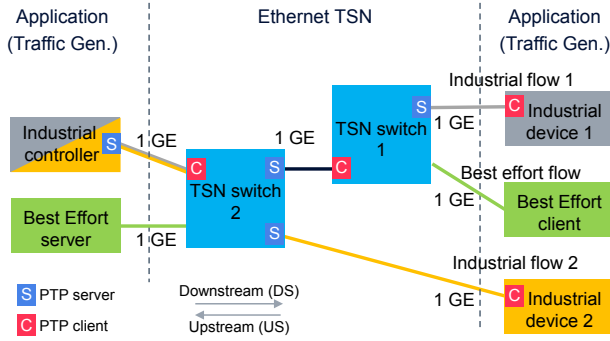


Fig. 8. Setup used to measure the TSN network.

The TSN switches were scheduled to isolate and follow the industrial flow cycles and to minimize the latency of the packets, avoiding their queuing in the TSN switches. In particular, for the upstream transmission, industrial flow 1 was configured to start at 0 μ s, and industrial flow 2 at 32 μ s. TSN switch 1 was configured with a 200 μ s cycle and to open gate 0 for flow 1 from 13 μ s (10 μ s for the reception + 2 μ s for switching to the output gate + 1 μ s of margin) and for 13 μ s duration (10 μ s for the transmission + 2 μ s for the opening/closing of the gate + 1 μ s for PTP inaccuracy margin). TSN switch 2 was configured with a 1 ms cycle (= LCM of the periods of the two industrial flows), to open the gate 0 for flow 1 from $26 + i \cdot 200 \mu$ s with $i = 0, 1, \dots, 4$, for 13 μ s, and to open the gate 1 for flow 2 from $39 + i \cdot 250 \mu$ s with $i = 0, 1, \dots, 3$, for 7 μ s. The remaining timeslots were configured to open the default gate that served the BE flow. Note that the above calculations were based on the delay (reception and transmission) specifications and margins of the TSN switches (Hirschmann RSPE35), p. 1360 [31].

Table 1 shows the measured packet latencies for the different flows. We can observe that the average packet latencies of the industrial flows were quite low, and their jitter was also low, below 1 μ s, which is the stringent requirement of industrial applications. The lower latency of flow 2 was because it crossed only one TSN switch and had a smaller packet size. Note that measuring the jitter of the scheduled flows indirectly measures the PTP synchronization accuracy. A packet arrives, according to the timing/gate opening of the previous hop, and waits for the gate opening of the current hop. We open the gate longer by a margin to account for the synchronization inaccuracy between the two hops. Thus, the measurement reflects the synchronization inaccuracy of the last two hops. We have accounted for such inaccuracies with a 1 μ s margin in all hops. So we indirectly know that PTP functioned properly across the whole path and according to IEEE 802.1AS standard, which requires synchronization within 1 μ s accuracy. We also observed that the latency was quite symmetric for US and DS for both industrial flows. Also, the measured latencies were equal to the configured offsets of the TSN switches minus their initial offsets. This indicates that the schedule was *tight* and the packets of both flows did not wait (apart from the applied margins) in the queues of TSN switches.

The jitter of the BE flow was quite higher (>70 μ s) than that of the industrial flows (<1 μ s). Note that the jitter of the BE flow depends on its load and on the other traffic. In Table 1

Table 1. TSN Network Latency Performance

Flow	Avg Latency (μ s)	Min Latency (μ s)	Max Latency (μ s)	Jitter (μ s)
Industrial flow 1 - DS (1250 bytes, 200 μ s)	26.083	25.392	26.120	0.728
Industrial flow 2 - DS (625 bytes, 250 μ s)	7.615	7.265	7.657	0.392
BE flow - DS (64–1512 bytes, 300 Mb/s)	32.582	6.538	80.072	73.534
Industrial flow 1 - US (1250 bytes, 200 μ s)	26.090	25.407	26.112	0.705
Industrial flow 2 - US (625 bytes, 250 μ s)	7.602	7.478	7.731	0.253
BE flow - US (64–1512 bytes, 300 Mb/s)	32.691	6.567	80.137	73.570

measurements, the load of BE was set to 300 Mb/s. We also increased the BE load to 900 Mb/s and we observed losses in the BE packets, and very high latency and jitter. Since the industrial flows were isolated through TSN IEEE 802.Qbv, they were unaffected by any other traffic flow in the network and their average latencies and jitter remained the same as shown in Table 1.

B. JC-PON Characterization

Figure 9 shows the logical representation of the setup we used to characterize the performance of the JC-PON (Section 3). Since the JC GWs function as tunnels from an ONU UNI port to an OLT SNI port, we required three SNI ports to serve the traffic. Due to the isolation among the tunnels, there is no need for a common time base. Consequently, we installed two independent PTP servers on the two ports that were tunneled to the two industrial devices.

We synthesized the same traffic flows as before: industrial flow 1 was 1250 bytes every 200 μ s, industrial flow 2 was 625 bytes every 250 μ s, and the BE flow was a random size of 64–1512 bytes and 300 Mb/s. For the JC GW configuration, we used a schedule of nine slots of 4032 Ethernet payload bytes (excluding preamble, start frame delimiter, and frame check sequence), 40 bytes interframe gap between the slots, plus 2454 bytes of varying gaps after the ninth slot for clocks alignment. The period of the schedule was set to 31.25 μ s (to match the PON burst period). Each slot corresponded to a BW of 1.032 Gb/s. In the schedule, we reserved the first slot

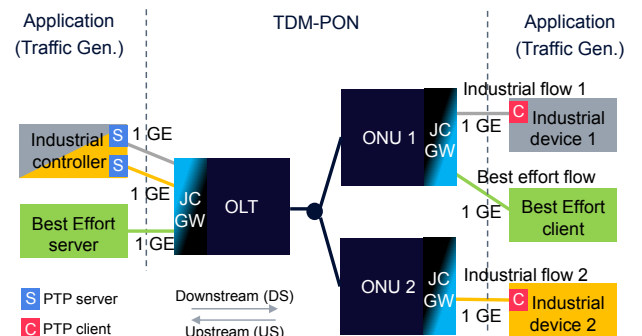


Fig. 9. Setup used to measure the JC-PON.

for ONU 1—industrial flow 1, the second slot for ONU 2—industrial flow 2, and the third slot for ONU 1—BE flow. Note that we intentionally reserved a slot for the BE flow to showcase the capabilities of the JC GWs; in normal operation the BE flows would have opportunistic access to any free slot. The PON was configured with a low-latency MAC with four bursts every PON frame, that is, a burst period of 31.25 μ s, and 1.050 Gb/s CIR for each of the three T-CONTs. The encapsulated Ethernet container frames from the JC GWs were aligned with the bursts with a custom PLL mechanism outlined in Section 3.

Table 2 presents the measured packet latencies for the different flows. We can observe the very low jitter for all the flows, including the BE random flow. The reason for this is that the JC GWs created tunnels and isolated each flow with a separate slot reservation. Even randomly arriving packets of the BE flow were transferred as they arrived; they experienced constant latency as if they were transferred over multiple P2P links. The JC-PON offered constant latency for any upstream flow (with a reservation), which was measured equal to 90.5 μ s and is the sum of the TDM-PON and JC GW cycle = 31.25 μ s (the waiting time at the JC GW until the bits are encapsulated to form the container) plus the transmission and propagation time for the 4032 byte container plus the JC GW processing time (FPGA implementation).

Note that the JC GWs were optimized for upstream transmission, taking into account the inter-burst gaps when aligning the JC GW frames and the PON bursts. The same alignment was used for downstream, which resulted in slightly varying latencies. This was an implementation choice and would be optimized in an integrated solution, where the JC GWs would be integrated into the ONUs and the OLT. It is also worth noting that US and DS latencies were asymmetric, and in particular for flow 1 the asymmetry was measured equal to 11.84 μ s. Such asymmetry should be and was accounted for when configuring PTP.

As stated above, the JC GW slots were 1.032 Gb/s and the PON reservation was 1.050 Gb/s adding small overheads to the intended reservation of 1 Gb/s. However, the intended reservation results in high overprovisioning (industrial flow 1 was 50 Mb/s and flow 2 was 20 Mb/s). This provisioning to the client port line rate plays a key role in the JC GW function (transferring the whole bitstream, including idle symbols) for achieving low jitter.

Table 2. JC-PON Latency Performance

Flow	Avg Latency (μ s)	Min Latency (μ s)	Max Latency (μ s)	Jitter (μ s)
Industrial flow 1 - DS	78.631	78.552	78.810	0.258
Industrial flow 2 - DS	79.046	78.960	79.215	0.255
BE flow - DS	79.452	79.360	79.614	0.254
Industrial flow 1 - US	90.471	90.385	90.562	0.177
Industrial flow 2 - US	90.514	90.417	90.610	0.193
BE flow - US	90.503	90.400	90.580	0.180

C. detDBA-PON Characterization

Figure 10 shows the logical representation of the setup we used to characterize the performance of the detDBA-PON. The PON was a part of the PTP boundary clock distribution tree. We synthesized traffic flows as before. Note that industrial flow 1 period (200 μ s) was not a multiple or divisor of the PON frame duration (125 μ s), and thus it was non-trivial to be served with a regular DBA [see Fig. 1(a)] and to exhibit low latency and jitter. We calculated the schedule and loaded the BWmaps of $T_{\text{detPON}} = 1$ ms duration (eight PON frames = LCM of industrial flow periods and of PON frame duration) to the OLT and repeated them every 1 ms. The bursts for industrial flow 1 were of 1280 bytes every 200 μ s and for flow 2 were 680 bytes every 250 μ s (set in equivalent PON words [15]). BE traffic was served with a 300 Mb/s T-CONT and a delay tolerance traffic descriptor of 250 μ s (two frames). The ranging process was disabled after the ONUs were operational on the PON. If ranging is required during PON operation, solutions such as whisper ranging should be applied [32] to avoid interfering with the detDBA operation.

The detDBA-PON setup was measured only in the upstream direction for which the detDBA mechanism was applied. At the OLT the two industrial flows were aggregated on the same SNI port and forwarded to the industrial controller, while the BE flow was forwarded through another port to the BE server. All three flows were interleaved without contentions on the detDBA-PON and they exited the PON with almost constant latency (low jitter). Then they went through the backplane path and crossed a switching element of the Nokia ISAM multi-OLT platform until they exited from the SNI port of the network (NT) card. Note that the Ethernet switch used in the ISAM platform did not implement TSN features. For that reason, we isolated the time-critical industrial flows from BE in the spatial domain, using different backplane paths and SNI ports in the ISAM platform (while in the PON we isolated them in the time domain with the detDBA).

Table 3 shows the measured packet latencies for the different flows (only for upstream). We measured an average latency of 36.12 μ s for industrial flow 1 and 28.38 μ s for flow 2. The latency jitter was below 0.220 μ s, showcasing the proper isolation of the flows. For comparison, the latency jitter for the BE flow was measured to be 223 μ s, which was in accordance with the delay tolerance descriptor configured for the T-CONT of that flow. The addition of other traffic flows would make the jitter of the BE flow much higher.

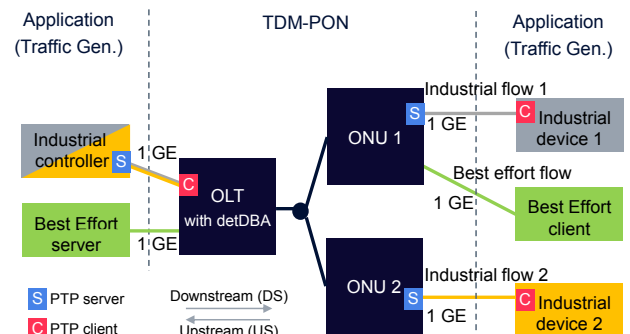


Fig. 10. Setup used to measure the detDBA-PON.

Table 3. detDBA-PON Latency Performance (Only Upstream)

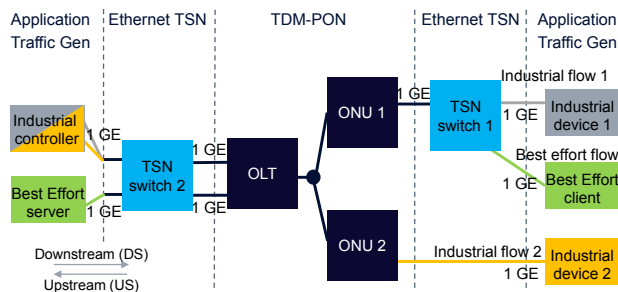
Flow	Avg Latency (μ s)	Min Latency (μ s)	Max Latency (μ s)	Jitter (μ s)
Industrial flow 1 - US	36.122	36.097	36.317	0.220
Industrial flow 2 - US	28.389	28.310	28.510	0.200
BE flow - US	124.711	21.860	245.012	223.152

D. TSN and PON PoC Experiments

Finally, we performed experiments with the complete TSN and PON network and for the two proposed deterministic PON approaches, the JC-PON and detDBA-PON. Figure 11 shows a generic representation of this setup, without indicating which deterministic PON approach is used.

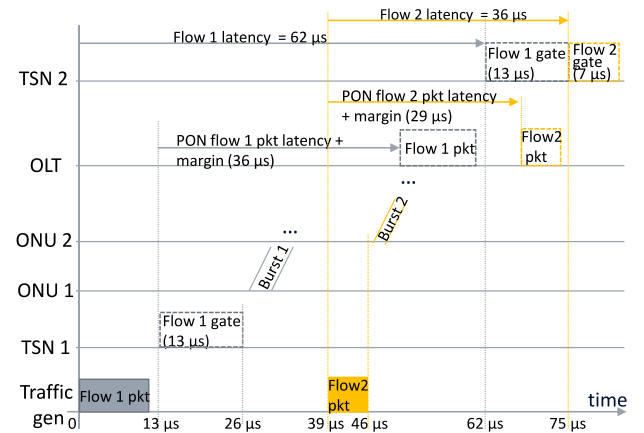
In the case of JC-PON, we included the JC GWs at the two ONUs and at the OLT. As discussed in Subsection 5.B the JC-PON did not play an active role in the PTP clock distribution, was not synchronized with the TSN switches nor with applications, and kept its own schedule. Its schedule was similar to that described in Subsection 5.B; the only difference was that we configured two T-CONTs and two tunnels instead of three, since the T-CONT of ONU1 carried both industrial flow 1 and BE flow mixed in TSN switch 1. The upstream/downstream asymmetry was known and accounted for in the scheduling and PTP configurations of the TSN switches. The TSN switch schedule was similar to that described in Subsection 5.A with the only difference that the gate opening times of TSN switch 2 were moved by 90.5 μ s (= latency of the JC-PON “cable”).

In the case of detDBA-PON, the PON became part of the PTP boundary clock distribution tree. The synchronization hierarchy was a mix of Figs. 8 and 10. In addition, the detDBA-PON and the TSN switches were co-scheduled: the detDBA-PON became one additional TSN switch in the network. In this experimental setup, we could set the starting time of the industrial flows. We calculated the schedule and the flow starting times to obtain the minimum latencies for both industrial flows so that their packets were not queued at any point, not in the TSN switches nor in the ONUs. In TSN scheduling each switch is represented with a constant and a size-dependent latency part; the TSN switch manual includes such specifications [31], while for the detDBA-PON the respective latencies for the 1250 byte and 625 byte frames were measured in the lab (Table 3). Taking those into account we calculated the following schedule for US. The industrial flow 1 was configured to

**Fig. 11.** Setup used to measure the combined TSN and deterministic (two approaches) PON network.

start at 0 μ s, and industrial flow 2 at 39 μ s. TSN switch 1 was configured with a 200 μ s cycle and to open gate 0 for flow 1 from 13 μ s and for 13 μ s duration. The PON was configured with a cycle of 1 ms (= LCM of two industrial flow periods and of PON frame duration), the bursts for T-CONT 1 (industrial flow 1) to start at $26 + i \cdot 200 \mu$ s with $i = 0, 1, \dots, 4$, for 1280 bytes, and the bursts for T-CONT 2 (industrial flow 2) to start at $46 + i \cdot 250 \mu$ s with $i = 0, 1, \dots, 3$, for 640 bytes. TSN switch 2 was configured with a 1 ms cycle, to open the gate 0 for flow 1 from $62 + i \cdot 200 \mu$ s with $i = 0, 1, \dots, 4$, for 13 μ s, and to open the gate 1 for flow 2 from $75 + i \cdot 250 \mu$ s with $i = 0, 1, \dots, 3$, for 7 μ s. The remaining timeslots were configured to open the default gate that served the BE flow. The calculated schedule for the first 82 μ s is shown in Fig. 12. For the downstream direction, where detDBA is not applicable, we used the TSN switch 2 to properly shape the traffic, creating protective timeslots of 15 μ s before each time-critical flow (industrial flows 1 and 2).

Figure 13 shows the latency and jitter measurements for the three synthesized flows in upstream. For both PON solutions, we measured jitter below 1 μ s for the industrial flows, which was our target, and a much higher jitter for BE, which was not controlled and would be higher if more traffic was mixed in the network. For both PON solutions, the average latencies of the industrial flows are equal to the sum of the latencies of the two individual systems (TSN and JC-PON and TSN and detDBA-PON), considering the added 1 μ s margins. This indicates that packets were not additionally queued at any point in the network. Thus, the scheduling calculations were properly performed: (i) the JC-PON tunneled the traffic and acted as a “cable” and was accounted for as the propagation delay in TSN scheduling, and (ii) the detDBA-PON yielded size-dependent known latency and was accounted for as an additional switch in TSN scheduling. The latency and jitter of the BE flow were better in the JC-PON than in the detDBA-PON. The JC-PON transports the industrial flow 1 and BE flow traffic as mixed in TSN switch 1 over the same tunnel and thus without adding jitter, while the detDBA-PON contributes varying latency to the BE. In both cases increasing the BE traffic load

**Fig. 12.** TSN switches and detDBA-PON schedule, targeting the minimization of latencies of the two industrial flows. Flow 1 packets are stored and forwarded at TSN 1, ONU 1, the OLT, and TSN 2, and flow 2 packets are stored and forwarded at ONU 2, the OLT, and TSN 2. Margins are used in timing calculations (duration of TSN gates and PON bursts).

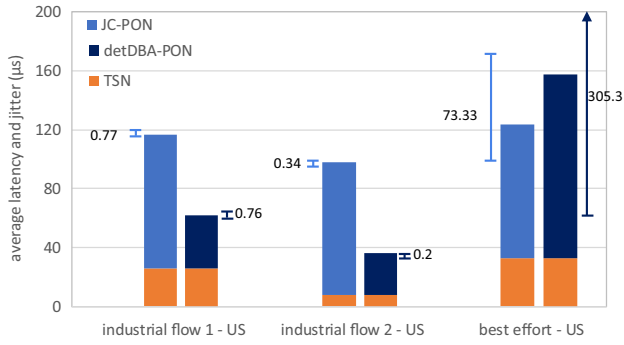


Fig. 13. Measured average end-to-end latency and jitter (shown in the sides of the bars) in μs for the three flows in US.

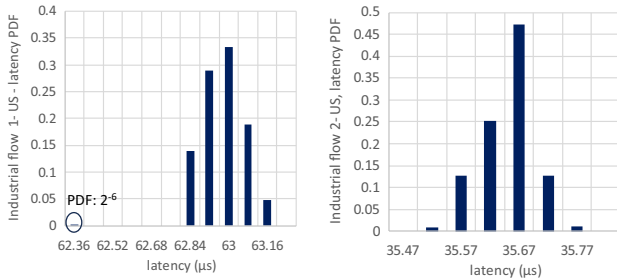


Fig. 14. Histogram of measured latency for the detDBA-PON industrial flows 1 and 2 in the upstream direction.

higher than 90% resulted in drops, higher latency, and jitter for the BE packets, while the industrial flows were unaffected. Figure 14 illustrates the histograms of measured latency for industrial flows 1 and 2 in upstream for the detDBA-PON. The latency histogram of industrial flow 2 has a Gaussian shape. Notably, for industrial flow 1, the majority of latency measurements cluster within 250 ns around the mean, with very few being approximately 500 ns lower than the mean. These occurrences are likely attributed to the synchronization process involving TSN switches and the PON. The findings suggest that the TSN switch 1, exclusively traversed by flow 1 and not flow 2, possesses a less stable clock source prone to faster drift compared to the others. Despite this, all network elements maintained synchronization, and the measured peak-to-peak latency variation (jitter) remained below 1 μs , within 801.AS operation bounds, in all tested scenarios. Due to space limitations, we do not include more histograms; information about the latency distribution of JC GWs is found in [25] and references within.

Figure 15 shows the measured latency and jitter in downstream. As discussed above, for the detDBA-PON, we shaped the traffic in TSN switch 2 to isolate the two industrial flows. We observe similar performances as in upstream: low latency and jitter below 1 μs for industrial flows, and high jitter for the BE flow that is not isolated.

E. Scalability of Proposed Solutions

The JC-PON operates without requiring synchronization of the PON with any client port. Consequently, it can cater to different applications that do not share a common clock reference and can interface with distinct TSN domains and/or domains

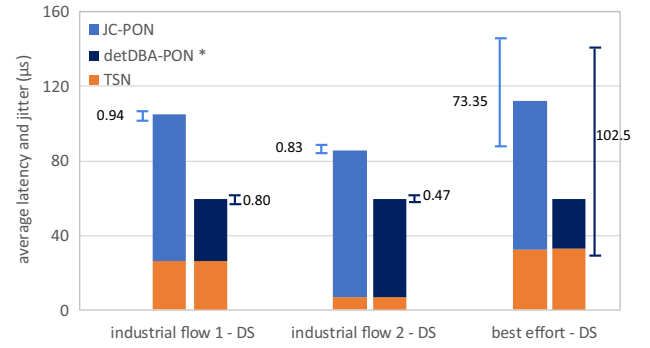


Fig. 15. Measured average latency and jitter (shown in the sides of the bars) in μs for the three flows in DS. DetDBA-PON* is not applied in DS (for this reason it is indicated with *), and industrial flows are isolated with margins in TSN switch 2.

utilizing different TSN flavors. The JC-GWs perform the crucial task of translating jitter to latency, resulting in latencies higher than the minimum (increased by the PON burst period, set at 31.25 μs in our PoC setup). The primary limitation of JC-PON stems from overprovisioning for the full client rate (1 Gb/s in our PoC setup, supporting up to 10 time-critical client ports in an XGS-PON). In such a configuration, it is better suited as a backbone to handle aggregated traffic from multiple industrial and BE flows, as demonstrated in ONU 1 where TSN switch 1 has aggregated the traffic. This limitation can be significantly alleviated by extending the JC-PON to support various client rates, with the most crucial to increase JC-PON efficiency and support industrial applications being those of 100 Mb/s.

On the other hand, the detDBA-PON delivers lower latency for industrial flows (the minimum achievable in our PoC) and jitter on the same order as the JC-PON, and reserves only the required bandwidth, thereby substantially improving PON bandwidth efficiency. However, detDBA is more intricate, demanding knowledge of the application traffic profile, global synchronization (across all served applications and interfaced networks), and co-scheduling (refer to Fig. 12). Note that these requirements mirror those of a converged state-of-the-art IEEE TSN Qbv-based network. The detDBA-PON finds its niche in directly serving low-rate time-critical flows, exemplified in ONU 2, which is directly connected to the end-device. Finally, it is worth noting that a hybrid approach, where both proposed solutions operate simultaneously, is also a viable option. This could combine the benefits of both approaches, according to the requirements of the clients.

6. CONCLUSIONS

We proposed and demonstrated two approaches for providing deterministic performance over a TDM-PON. In the JC-PON we operated the PON asynchronously to its clients, scheduled it to provide upstream bursts at a low and constant period, and used JC GWs to compensate for the jitter introduced by the non-matching PON schedule and the input traffic cycles. In the detDBA-PON, we implemented a DBA that controls the placement of bursts at each PON frame for selected T-CONTs that require deterministic performance and have known traffic cycles. In this case, we synchronized and co-scheduled the

detDBA-PON with the rest of the network and the applications. As a use case, we applied the two proposed solutions, the JC-PON and the detDBA-PON, to serve industrial flows from devices that are either directly connected to the PON or interfaced through TSN switches. We demonstrated latency below 100 μ s and sub- μ s packet latency variation (jitter) for the industrial flows over short distances, while also serving best-effort traffic flows.

Funding. Bundesministerium für Bildung und Forschung; Hellenic Foundation for Research and Innovation.

Acknowledgment. The authors would like to thank R. Sundaralal, J. Poffyn, and W. V. Hoof for their support in the measurements.

This work was partially supported by the German BMBF with the EU CELTIC project AI-NET ANTILLAS and the project 6G-ANNA and by the Greek HFRI through the project DetermIneD.

REFERENCES

1. S. Bidkar, J. Galaro, and T. Pfeiffer, "First demonstration of an ultra-low-latency fronthaul transport over a commercial TDM-PON platform," in *Optical Fiber Communication Conference (OFC)* (2018), paper Tu2K.3.
2. H. Nomura, H. Ujikawa, H. Uzawa, H. Nakamura, and J. Terada, "Novel DBA scheme integrated with SR- and CO-DBA for multi-service accommodation towards 5G beyond," in *European Conference on Optical Communication (ECOC)* (2019).
3. K. Takahashi, H. Nakamura, H. Uzawa, K. Miyamoto, Y. Nakayama, T. Shimada, J. Terada, and A. Otaka, "NG-PON2 demonstration with small delay variation and low latency for 5G mobile fronthaul," in *European Conference on Optical Communication (ECOC)* (2017).
4. A. Zaouga, A. F. de Sousa, M. Najjar, and P. P. Monteiro, "Self-adjusting DBA algorithm for next generation PONs (NG-PONs) to support 5G fronthaul and data services," *J. Lightwave Technol.* **39**, 1913–1924 (2021).
5. H. Chung, H. H. Lee, K. O. Kim, K.-H. Doo, Y. Ra, and C. Park, "TDM-PON-based optical access network for Tactile Internet, 5G, and beyond," *IEEE Netw.* **36**, 76–81 (2022).
6. M. Jiang, "Advances in industrial PON," in *Mo1F Workshop, ECOC* (2021).
7. T. Pfeiffer, P. Dom, S. Bidkar, F. Fredricx, K. Christodouloupoulos, and R. Bonk, "PON going beyond FTTH," *J. Opt. Commun. Netw.* **14**, A31–A40 (2022).
8. K. Christodouloupoulos, S. Bidkar, W. Lautenschlaeger, T. Pfeiffer, and R. Bonk, "Demonstration of industrial-grade passive optical network," in *Optical Fiber Communication Conference (OFC)* (2022), paper Tu2G.6.
9. K. Christodouloupoulos, S. Bidkar, T. Pfeiffer, and R. Bonk, "Deterministically scheduled PON for industrial applications," in *Optical Fiber Communication Conference (OFC)* (2023), paper Tu3F.5.
10. O-RAN, "Control, user and synchronization plane specification," O-RAN-WG4.CUS.0-v08.00 (O-RAN Alliance, 2022).
11. 5G-ACIA, "Industrial 5G edge computing—use cases, architecture and deployment," white paper.
12. S. Dietrich, G. May, O. Wetter, H. Heeren, and G. Fohler, "Performance indicators and use case analysis for wireless networks in factory automation," in *Emerging Technologies and Factory Automation (ETFA)*, Limassol, Cyprus, 2017.
13. "OPC UA: industrial interoperability standard from sensor to cloud (and back)," Industry 4.0 OPC Foundation Webinar, 1 December 2020.
14. IEEE, "Time-sensitive networking (TSN) task group," <http://1.ieee802.org/tsn/>.
15. "40-gigabit-capable passive optical networks (NG-PON2): transmission convergence layer specification," ITU-T Recommendation G.989.3 (2021).
16. S. Bidkar, R. Bonk, and T. Pfeiffer, "Low latency TDM-PON for 5G xhaul," in *International Conference on Transparent Optical Networks (ICTON)* (2020).
17. X. Liu, "Enabling optical network technologies for 5G and beyond," *J. Lightwave Technol.* **40**, 358–367 (2022).
18. "Optical line termination capabilities for supporting cooperative dynamic bandwidth assignment," ITU-T G Suppl. 71 (2021).
19. M. Rostan, "Industrial Ethernet technologies: overview," 2014, https://www.etherncat.org/download/documents/Industrial_Ethernet_Technologies.pdf.
20. J. Y. Le Boudec and P. Thiran, *Network Calculus: A Theory of Deterministic Queuing Systems for the Internet*, Vol. 2050 of Lecture Notes in Computer Science (Springer, 2001).
21. "Asynchronous traffic shaping," IEEE Std. 802.1Qcr-2020 (2020).
22. "IEEE standard for local and metropolitan area networks," IEEE Std 802.1AS-2011 (2011).
23. S. S. Craciunas, R. S. Oliver, M. Chmelik, and W. Steiner, "Scheduling real-time communication in IEEE 802.1Qbv time sensitive networks," in *International Conference on Real-Time Networks and Systems (RTNS)* (ACM, 2016).
24. K. Jeffay, D. F. Stanat, and C. U. Martel, "On non-preemptive scheduling of period and sporadic tasks," in *Real-Time Systems Symposium* (1991).
25. W. Lautenschlaeger, F. Frick, K. Christodouloupoulos, and T. Henke, "A scalable factory backbone for multiple independent time-sensitive networks," *J. Syst. Archit.* **119**, 102277 (2021).
26. Datatracker, "Deterministic networking (detnet) charter," <https://datatracker.ietf.org/wg/detnet/>.
27. D. Mafioletti, F. Slyne, R. Giller, M. O'Hanlon, D. Coyle, B. Ryan, and M. Ruffini, "Demonstration of a low latency bandwidth allocation mechanism for mission critical applications in virtual PONs with P4 programmable hardware," in *Optical Fiber Communication Conference (OFC)* (2022), paper M3Z.7.
28. C. Su, J. Zhang, and Y. Ji, "Time-aware deterministic bandwidth allocation scheme in TDM-PON for time-sensitive industrial flows," *J. Opt. Commun. Netw.* **15**, 255–267 (2023).
29. O-RAN Alliance, "Cooperative transport interface transport control plane specification," O-RAN.WG4.CTI-TCP.0-v01.00 (2020).
30. L. George, N. Riviere, and M. Spuri, "Preemptive and non-preemptive real-time uniprocessor scheduling," Research report RR-2966 (INRIA, 1996).
31. Hirschmann, "RSPE HiOS-2A Rel. 07300 Reference manual."
32. R. Bonk, R. Borkowski, M. Straub, H. Schmuck, and T. Pfeiffer, "Demonstration of ONU activation for in-service TDM-PON allowing uninterrupted low-latency transport links," in *Optical Fiber Communication Conference (OFC)* (2019), paper W3J.4.