Dynamic Deterministic Digital Infrastructure for Time-Sensitive Applications in Factory Floors

Sébastien Bigo, *Fellow, IEEE*, Nihel Benzaoui, Konstantinos Christodoulopoulos, Ray Miller, Wolfram Lautenschlaeger, and Florian Frick

(Invited Paper)

***Abstract*—By massively digitalizing production processes, the industry expects an efficient and secure cooperation of a large number of machines, as well as an increasing integration of IT-systems for control and services, for instance using edge clouds. Their vision calls for a distributed, converged datacom and telecom infrastructure where performance, can be deterministic per application, i.e. strictly guaranteed, end-to-end, across technologies and across layers. Its success will largely depend on how dynamically it can reuse resources and how cost-effectively it can host mainstream best-effort applications. In this paper, we review the key performance indicators for such an infrastructure while confronting them to thirty use cases reported by the industry. We then examine the requirements for the digital infrastructure. We benchmark them against the capabilities of relevant network technologies, while pointing to meaningful enhancements. We particularly discuss Time-Sensitive Networking (TSN), a set of standards extending IEEE 802.1 Ethernet with some deterministic functions, which is considered to be the most promising enabling technology for future industrial networks. We propose to augment its scalability with a novel optical backbone which can interconnect islands of TSN networks while preserving timing.**

***Index Terms*—Industry 4.0, deterministic network, edge cloud, 5G, Time Sensitive Networking, industrial Ethernet, low latency, jitter, end-to-end performance, quality of service.**

I. Introduction

IN THE past decade, the internet, intranets and data communication networks have merged into a single, global digital infrastructure capable of a myriad of applications leveraging connections of every “thing” (terminal, robot, server, object) to any other “thing”. While there is no denying of the immense opportunities, some applications had to be discarded or delayed because performance expectations are not met. When legacy technologies with various performance characteristics are concatenated, the digital infrastructure fails to meet the most demanding requirements of time-sensitive applications end-to-end (E2E). Applications are generally completed faster than a target time, but this cannot be guaranteed. The probability of exceeding that target time can be reduced by tweaking the systems at the expense of additional cost but can hardly be guaranteed end-to-end.

As the digital infrastructure expands, in particular into manufacturing shop floors [1] [2], the connected “things” are increasingly expected to massively cooperate [3]. In such scenarios, guaranteeing a deterministic transmission is essential, which is significantly harder to achieve. Perfectly matching the target time requires that the deviation of latency (referred to as jitter) is ideally zero [4], [5]. All those challenges explain why the cloud has largely been restricted to best effort applications so far. By contrast, the most demanding applications would require performance (i.e. latency, jitter, and reliability) to be strictly deterministic [6] i.e. guaranteed across the entire end-to-end digital infrastructure [7], [8]. Deterministic performance is standard practice in some networking technologies, e.g. in long-haul optics [9]. Those technologies can maintain determinism over data streams from thousands or millions of applications but would not be economically viable if expanded to the edge of the infrastructure, where each and every application needs to be discriminated with its own level of determinism. The challenge there would be twofold: applications flows at the edge are multiple orders of magnitude smaller than aggregated data streams in long-haul optics, but they also change much more frequently [10] (at sub-second speed to compare with minutes or hours in long-haul optics at best), as extrapolated from today’s centralized data centers [11], [12]. Therefore, the time to turn up/tear down a flow can become a significant portion of the overall flow completion time and therefore becomes a concern.

Therefore, the requirements on dynamic operation and on deterministic performance are two sides of the same coin, and they need to be translated into two requirements across the end-to-end infrastructure. They have been optimized separately in the past and fundamentally pull in opposite directions. The key challenge of a future dynamic deterministic network (DDN) is to manage them holistically and to provide an innovative game-changing mix for determinism and dynamics.

Manuscript received February 25, 2021; revised June 7, 2021; accepted June 22, 2021. Date of publication June 29, 2021; date of current version July 13, 2021. *(Corresponding author: Sébastien Bigo.)*

Sébastien Bigo and Nihel Benzaoui are with Nokia Bell Labs, 91620 Nozay, France (e-mail: sebastien.bigo@nokia-bell-labs.com; nihel_djoher.benzaoui@nokia-bell-labs.com).

Konstantinos Christodoulopoulos and Wolfram Lautenschlaeger are with Nokia Bell Labs, Stuttgart 70435, Germany (e-mail: konstantinos.1.christodoulopoulos@nokia-bell-labs.com; wolfram.lautenschlaeger@nokia-bell-labs.com).

Ray Miller is with Nokia Bell Labs, Murray Hill, NJ 07974 USA (e-mail: ray.miller@nokia-bell-labs.com).

Florian Frick is with ISW, University of Stuttgart, 70174 Stuttgart, Germany (e-mail: florian.frick@isw.uni-stuttgart.de).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/JSTQE.2021.3093281.

Digital Object Identifier 10.1109/JSTQE.2021.3093281

In the first part of this paper, we discuss the key indicators which should be used to specify dynamic deterministic networking. Then we review thirty use cases, with a particular focus on industrial applications, and discuss the target values for these key indicators. We later discuss the readiness of available technologies for DDN, while highlighting which improvements or hardenings would be required to make them compliant with the most demanding time-sensitive use cases. Time-Sensitive Networking (TSN), as standardized by an IEEE 802.1 working group, deserves a particular focus as promising platform to support DDN. We discuss scalability challenges of TSN in a factory floors and show results of an experiment, where we alleviate those challenges with an innovative optical backbone (Section V). Apart from IEEE 802.1 TSN, the work of this paper was also partly inspired by the IETF DetNet workgroup. The most original content (e.g., in Section V) has not been promoted in any of those forums yet but will require a broad agreement from industry to gain market acceptance.

II. Key Performance Indicators for Dynamic Deterministic Networks

Organizing the massive cooperation of connected “things” is not a recent problem. It started with personal computers. Twenty-five years ago, after acknowledging the limitations of the digital infrastructure at the time, L. Peter Deutsch popularized a set of false assumptions that junior programmers invariably make when programming distributed applications. These false assumptions are now widely known as the eight fallacies of distributed computing [13]. When the connected “things” become robots in a manufacturing shop floor or servers in an edge cloud, overturning the eight fallacies becomes our guiding principle for the design of an optimal digital infrastructure. We use this principle for the definition of the key performance indicators [14].

A. A Converged Platform

Because of massive success of legacy networking equipment to carry data with best-effort technologies, we believe that DDN should not support all applications as if they were time-sensitive, but allow for the coexistence of premium time-sensitive applications with non-time-sensitive, best-effort applications in a converged infrastructure, as schematized in Fig. 1. Mainstream traffic (e.g. web browsing, video) should continue to travel across the digital infrastructure, with unchanged performance requirements with respect to today’s networks.

Even if the fine dimensioning of network resources can be left to each and every use case, we assume that real-world scenarios should not involve significantly more than 10% premium traffic with deterministic requirements. The review of use cases further down in this paper can help estimate the maximum premium traffic ratio, and there is obvious room for this ratio to be tuned later as market needs refinement. The 10% limit is to be understood as average value that does not preclude close to 100% premium occupation on a selected instance or time. The limitation is of economic nature as determinism comes at the cost of exclusive resource allocation that restricts statistical multiplexing and which is prone to admission blocking when free resources are insufficient. Hence, selecting (restricting) the ratio of premium traffic is an essential assumption for the success of any DDN approach. One-size-fits-all combinations are unlikely. Optimal combinations will depend on use cases and should ideally allow for programmable levels of determinism (as-a-service determinism). However, the largest benefits for DDN are expected where potential jitter is at risk of exceeding one tenth or larger than the (average) latency, i.e. where the applications are performed over relatively short distances, namely at the edges of the network.

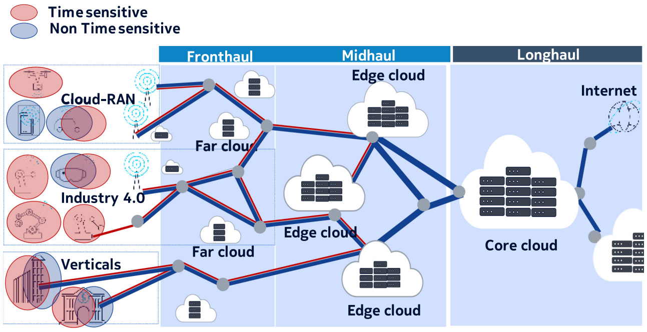


Fig. 1. Schematic of a DDN network. The most time-sensitive applications need to be processed at the edge, while mainstream applications will continue to propagate over longer distances into the core of the networks. RAN stands for radio Access network.

In particular, industries planning for the digital transformation of their manufacturing processes have foreseen opportunities for augmenting the control of their factory floor through advanced 5G wireless, transport and cloud, thereby attracting attention of service providers, eager to serve them adequately. However, these enterprises have made clear that they cannot give up on deterministic performance (i.e. negligible jitter and ultralow data loss) across the wireless access, the fiber transport network, and edge data center (DC) altogether, i.e. end-to-end, which raises important challenges. All scenarios where time-sensitiveness matters, e.g. ultra-reliable low latency communications (URLLC) or high-throughput low latency vehicle-to-anything (V2X) communications share similar constraints [1], but we will use the Industry 4.0 factory floor as our reference.

B. Reliability

Reliability at the physical layer is a critical metric for the new suite of industrial applications and services. Without a reliable network, metrics such as latency and jitter lose their value. Zero packet loss ($<<10^{-10}$ loss ratio) is expected for the most time-sensitive services, well below losses of IT services in today’s data centers (typ. 10^{-3} loss ratio), which protocols like TCP correct today, at the expense of prohibitive end-to-end latency and jitter. Therefore, for the most critical applications, redundant end-to-end connections within the access, transport, IP, and DC domains will be the primary method for establishing enhanced reliability. Enhanced domain network control at <50 msec speeds will be essential to support this new level of transport reliability, particularly in access and edge networks.

C. Latency

Many industrial applications require a very fast response time and necessitate a low network latency. Because no one can beat the speed of light, data cannot be exchanged over too long distances, and can only travel to a nearby data center. Network slicing technologies, popular in virtual radio access networks (vRAN), can help discriminate between applications requiring processing in an edge (close) data center or a core (remote) data center. When applications are more sensitive to jitter than latency, longer buffers can be used to equalize arrival times and compensate for jitter, but at the expense of increased latency and cost [15], as high-speed buffers are required and are generally quite expensive.

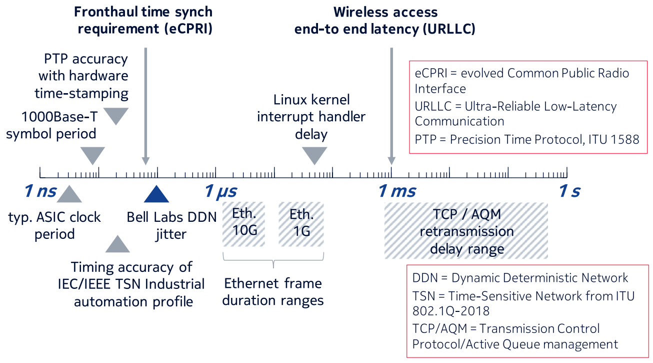

Fronthaul time synch requirement (eCPRI)
Wireless access end-to end latency (URLLC)
PTP accuracy with hardware time-stamping
1000Base-T symbol period
Linux kernel interrupt handler delay
eCPRI = evolved Common Public Radio Interface
URLLC = Ultra-Reliable Low-Latency Communication
PTP = Precision Time Protocol, ITU 1588
1 ns
1 μs
1 ms
1 s
typ. ASIC clock period
Bell Labs DDN jitter
Eth. 10G
Eth. 1G
TCP / AQM retransmission delay range
Timing accuracy of IEC/IEEE TSN Industrial automation profile
Ethernet frame duration ranges
DDN = Dynamic Deterministic Network
TSN = Time-Sensitive Network from ITU 802.1Q-2018
TCP/AQM = Transmission Control Protocol/Active Queue management

Fig. 2. Time scale showing operation range of commonplace technologies (triangles) and requirements of network services (arrows).

D. Jitter

In packet networks, jitter (defined as excursion of latency) stems from two origins: (1) the path varies during the transmission of the packet flow; (2) the waiting time in the queues varies as a result of congestion, when other flows compete for the same output.

Forcing all data from the same flow to travel across the same path is the first mechanism to contain jitter. In case protection is activated, then the length of the protection path should be pre-equalized with respect to length of the nominal path. Such pre-equalization is already common practice in legacy field buses for factory floors but could have to be generalized for all network segments hosting premium time-sensitive traffic.

Over-provisioning of capacity is standard practice today for containing the impact of congestion. However, unless prohibitively expensive, it is rarely stretched to the point where deterministic performance can be guaranteed for the most demanding applications. Competition between no more than two dozen flows into the same switch port can increase the latency by more than 100x when unfortunate combinations take place. Owing to the low probability of such heavy congestion events, high-speed buffers are not dimensioned large enough to compensate for jitter during those events, hence strict performance guarantees are precluded. Therefore, over-provisioning hardly scales to keep flow competition under control at all times, especially for time-sensitive applications running in the cloud.

The control of jitter, as well as all other forms of timing control, relies on a family of technical approaches which are compared in Fig. 2, from Integrated circuits to network hardware and to protocols, over nine orders of magnitude. Jitter control requires to share, distribute, or carry a time reference from source to destination and can only be as good as the precision of that time reference. The requirements for Industry 4.0 applications are discussed in the use case review section later in this paper. They can be benchmarked along the time scale of Fig. 2, where the timing precision requirement for the evolved Common Public Radio Interface of 5G radio and eCPRI and of Ultra-Reliable Low latency communication are also shown. We also reported in Fig. 2 relevant durations, as reference points for comparisons.

Timing and jitter for network communication play a large role in ensuring the reliable performance which most industrial applications need. Even for relaxed industrial use cases which would regularly send "heartbeat" signal every 4 msec, jitter must be much less than 4 msec in order to avoid failure of the application. Most industrial applications have much stricter requirements on jitter, down to the nanosecond range. Jitter requirements of 100 msec or less largely prevent the use of TCP for retransmission and its use for addressing the reliability requirements.

In a typical end-to-end scenario involving cloud computing and Operation Technology (OT) built with off-the-shelf legacy technologies, timing precision would typically span across the entire range of Fig. 2, and therefore be limited by technologies with the weakest timing control. However, industry standards require to operate in the left-most part, where the TSN standards lie, building upon legacy field bus technologies. This suggests that edge clouds performing some forms of industry control tasks will have to fully mutate with technologies with similar requirements as e-CPRI and TSN, into a form of quasi-synchronous data center. By contrast, applications running at a lower level of determinism over a Linux kernel jitter cannot expect timing control better than a few 10μs. Fig. 2 recalls that while virtualized Ethernet switches have come true and are gaining in popularity, virtualized (software-defined) deterministic switches are largely out of reach and would require breakthroughs in digital platforms, in deterministic operating systems and in deterministic server architectures.

In all scenarios, jitter compensation mechanisms shall be used, where best fitted, i.e. wired in hardware (switches) in order to comply with time sensitive industry standards, or by software at the protocol and application layers to comply with lower determinism levels. [15]

Deterministic communications with low jitter can also improve software performance by allowing for better determinism in operation in the server, freeing up resources by reducing buffering time.

Less time waiting for information allows for more efficient processing. Deterministic communications can turn data centers into (quasi) synchronous. First simulations by Bell Labs predict 70% compute time gain by joint optimization of network and compute resources for distributed application in deterministic-data center [16]. Even some papers discuss the feasibility of full synchronous data centers [17], we favor here a scenario with hybrid best-effort/synchronous operation for all reasons discussed above.

E. Dynamics

In the scenario that we consider as most likely, time sensitive applications represent a relatively small fraction of the traffic hosted in the converged infrastructure. Customers of these applications will accept to pay premium prices and therefore utilize network resources for a (moderately) longer duration than the duration of the application. There is little doubt that the success of deterministic networking will largely depend on the connectivity dynamics and the ability for resources to be quickly reused, as exemplified by the economics of the cloud.

We postulate that the best place to seize the extent of the expected revolution ahead of us is to extrapolate from where machine-to-machine communications already dominate today, namely inside big data centers. There, any server can turn-up an IT service with any other server, even located at the other end of the data center, to exchange a flow for a duration that can be orders of magnitude shorter than the smallest service duration in today's optical networks.

The extreme variability of flows called for networks where services are forced to compete for shared resources, away from peak traffic conditions, away from the dependability of optical networks, and multiplexed statistically. For example, the amount of flows competing in a single switch of a data center can range by hundreds or thousands, producing sizeable jitter and ultimately prohibitive packet loss. When those traffic conditions gradually apply to the future edge cloud, then we argue that whatever possible combinations of legacy optical transport and switching technologies, end-to-end service turn-up time and jitter could soon approach or exceed two boundaries, as indicators of inefficient networking:

1) when service setup or tear-down takes as much time as service duration.
2) when jitter is as large as latency.

Obviously, a good design of the future digital infrastructure should stay away from those two boundaries to be cost-competitive.

III. A Review of Time-Sensitive Applications

Based on current technological trends, we conjecture that the digital transformation of manufacturing and consequently the related applications, platforms and converged communication infrastructure will evolve along a similar path of transformations as IT has been evolving. Eventually, purely digital and real-world systems, like servers and robots, will become connected "things", as shown in Fig. 3. The share and value of software will take precedence over hardware in the entire value chain. Thereby hardware will increasingly be generic and configured by software [18], thereby enabling a form of software-defined manufacturing. In the past, classic automation would consist of mechanical system, some electronics and a purpose-built controller, designed primarily for greater reliability. The demand for flexible, more cost-effective production called for modular, easier-to-exchange hardware, resulting in flexible mechatronics systems. Over time, stand-alone machines were turned into connected things, while controllers were kept local.

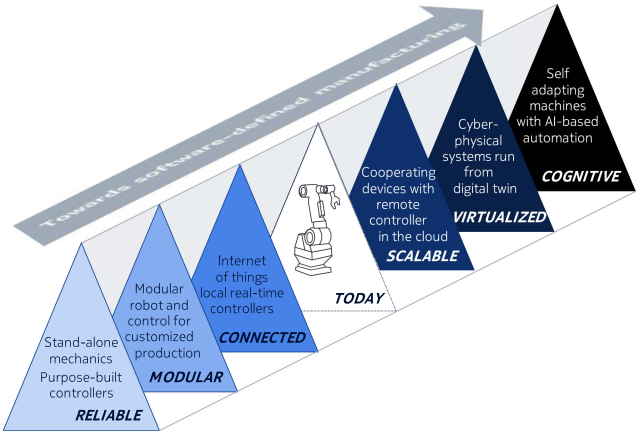

Towards software-defined manufacturing
Stand-alone mechanics Purpose-built controllers
RELIABLE
Modular robot and control for customized production
MODULAR
Internet of things local real-time controllers
CONNECTED
TODAY
Cooperating devices with remote controller in the cloud
SCALABLE
Cyber-physical systems run from digital twin
VIRTUALIZED
Self adapting machines with AI-based automation
COGNITIVE

Fig. 3. A roadmap towards software-defined manufacturing.

The on-going digital transformation, also referred to as Industry 4.0, will bring further flexibility through a closer control of IT and OT systems, allowing for example remote control of machines from a pool of centralized servers, inside an edge cloud. The added value of introducing a cloud in factory automation will increase when additional services like digital twins of the factory floor are added, allowing decisions for large scale cyber-physical systems. Virtualization could be the next phase of factory transformation whereby the product is increasingly defined by the software and less by the production tool itself, while orchestration will allow for interworking across a multiplicity of hardware, software and communication technologies. Ultimately, cloud-based machine learning will suggest expert decisions and ease self-adaptation of factory floors.

A. Overview of Data Communication in the Industry 4.0

This section gives an overview of time-sensitive applications for Industry 4.0 and how their requirements affect network design. These applications have a need to communicate data end-to-end and consistently, where consistency is assessed in terms of hard-bounded guarantees, for one or more performance metrics.

In the upcoming transformation in the context of Industry 4.0, most applications require a cooperation between machines, and hence have to exchange data reliably across the factory floor network but also the edge DC, as schematized in Fig. 4. Instead of classifying applications per groups of use cases, we searched for common denominators in their data exchange patterns. In the following we differentiate between machines in the factory floor (robots, engines, automated guided vehicles (AGV), user terminals, etc.) and servers in the edge data center. We identified five classes of machine to machine, machine to server and server to server communications:

i) Control loops where control data are exchanged between a controller, and the sensors and actuators of the machine, and looped back to the controller. Loops require ultralow

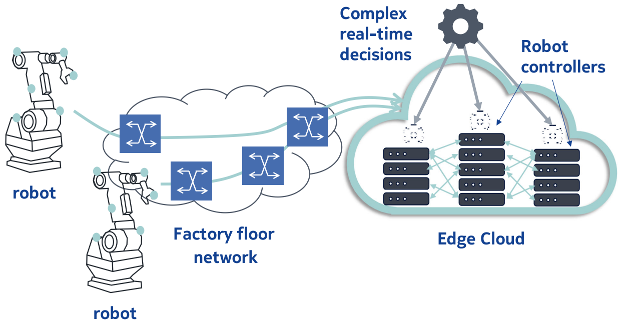


Fig. 4. Typical layout of cognitive Industry 4.0.

and bounded latency. When the controller is hosted in a server within the edge DC, the communication between the machine and its controller necessitates performance determinism on both the factory floor network and the network segment connecting to the server inside the edge DC.

ii) Cooperative control loops refer to the control of coordinated actions between two or several machines. This coordination takes place inside the edge DC where controllers need to exchange specific data for the consistent coordination of machines, before each controller sends the command back to its machine. The ultralow latency requirements expand into the edge DC network. On top of latency, a precise synchronization is required.

iii) Data stream for factory floor operation where machines send a periodic constant data stream from sensors or cameras to servers for applications like digital twining, or collision detection. Because real time decisions are taken based on the collected data from the machines, these data need to be as up to date as possible. This sets constraints on latency on factory floor networks and intra edge DC interconnects.

iv) Data streams for services where data are shared between machines and multiuser servers for specific application involving the creation of a shared media environment such as virtual meetings. These applications are resilient to packet losses and absolute latency but require a very low jitter to synchronize the experience for all service tenants. Jitter needs to be contained end-to-end across the factory floor and the edge DC

v) Parallelized processing for compute-hungry real time applications; e.g., a simulation based on digital twins to improve time-critical decision making. Data processing needs to be parallelized to reduce the overall time execution. Latency and reliability set therefore paramount requirements, while reduced jitter will make it possible to run the edge data center in a synchronous manner, thereby decreasing further the execution time [16].

B. Requirements for Time Sensitive Applications

An examination of literature provided by various standards bodies and industrial forums provides a reasonably large set of applications and use cases which are time sensitive and therefore relevant to the discussion on determinism.

TABLE I
CONTROL MESSAGES REQUIREMENTS

Applications	*Latency*	*Jitter*	*Data rate*	*Reliable (%)*
UNIDIRECTIONAL CONTROL				
Periodic smart grid distance protection and intra-substation process bus communication [5]	5ms	NA	64kbps	99.999
periodic smart grid inter-substation protection signaling [5]	5ms	NA	64kbps	99.999
smart ambulance -data [19]	5ms	NA	1Mbps	99.999
periodic wind turbine monitoring and control [5]	16ms	NA	75kbps	99.99
critical short-range road safety [20]	20ms	NA	60kbps	99.999
aperiodic tactile interaction [21]	0.5ms to 1ms	NA	100kbps	NA
aperiodic smart grid inter-trip protection [5]	5ms	NA	NA	99.9999
smart factory massive wireless sensor networks (for safety) [22]	5ms to 10ms	NA	NA	up to 99.99999
building monitor and control – alarms, controls [5]	< 10ms to 100ms	< 1ms	NA	NA
BIDIRECTIONAL CONTROL				
Critical smart factory control/data streams (e.g., safety heartbeat) [5]	100µs to 50ms	NA	NA	99.999
periodic smart factory motion control [22][23]	< 500µs to < 2ms	= one-way latency	10kbps	99.999999
periodic smart factory control to control [22]	4ms	2ms	1kbps	up to 99.99999
periodic smart factory mobile control panels with safety functions [22][23]	4ms to 8ms	<½ latency	NA	up to 99.999999
periodic smart grid current differential protection [5]	5ms	<250µs	64kbps	99.9999
aperiodic smart grid tele-protection [5]	4ms to 8ms	<250µs	NA	99.9999
aperiodic smart factory control to control [22]	<10ms	<10ms	<1kbps /control loop	up to 99.999999
aperiodic smart factory mobile control panels with safety functions [22][23]	<30ms	15ms	>5Mbps	NA
UNIDIRECTIONAL MULTI FLOW				
cooperative driving [24]	10ms-25ms	NA	65Mbps	up to 99.99

(NA stands for not available)

While the applications and use cases span across numerous industrial segments, there are structural similarities regarding the behaviors of the services. They differ primarily by the target values of the service parameters: latency, jitter, data rate, and reliability, as reported in the tables below. All examined time-sensitive applications fall into one of the following categories:

1) Control messages, used in communication classes i) and ii) need the most stringent latency and reliability guarantees (Table I). Control messages may be periodic time-triggered, i.e. continuously generated, or aperiodic

TABLE II
REAL-TIME MEDIA REQUIREMENTS

Applications	*Latency*	*Jitter*	*Data rate*	*Reliable (%)*
RATE ADAPTIVE UNIDIRECTIONAL				
automated guided vehicles [22][23]	1ms to 500ms	<½ latency	10Mbps	99.9999
video from drones and other machines/robots [25]	1ms	10s of Mbps	NA	100 (utopian in ref [25])
smart ambulance – media [19]	< 20ms	NA	200Mbps to Gbps	99.999
tele-operated driving [20]	20ms	high	25Mbps-40Mbps	NA
real time video surveillance [22]	< 500ms		80Mbps	99.99
RATE-ADAPTIVE BIDIRECTIONAL				
augmented reality [22]	<50ms		2-5Mbps	99.9
CONSTANT BIT-RATE UNIDIRECTIONAL				
remote surgery [28]	1ms		NA	NA
low latency audio streaming [22]	4ms		<4Mbps	99.9999

(NA stands for not available)

event-triggered, i.e., sporadically generated. They may be unidirectional, where messages do not strictly depend on reverse-path feedback, or bidirectional, where messages do depend on reverse-path feedback. Systems requiring bidirectional periodic communication must receive replies before generating the next message. Alternatively, systems requiring bidirectional aperiodic control messages use a strict bound on round-trip time. Control messages can also be unidirectional multi-flow cooperative, where one node exchanges control data with many other nodes in close proximity.

2) Real-time media, used in communications classes iii) and iv) has stringent latency and reliability, but also high data rate requirements (Table II). In real-time media applications, frames must be delivered at a constant rate. A delayed frame is a lost one. Real-time media may be
   - rate adaptive unidirectional with the encoding rate adapting to non-guaranteed data path conditions, and interactivity through low-bandwidth control traffic on the reverse path; requiring tens of Mbps throughput, single digit ms latencies, and even less jitter
   - rate adaptive bidirectional with the encoding rate also adapting to non-guaranteed data path conditions, but in this case the source and receiver are co-located, and the network provides intermediate processing - delay constraint applies to the round-trip time (RTT), not just unidirectional delay; may require Mbps throughput and tens of ms round trip latencies
   - constant bit rate unidirectional where the encoding rate is fixed and does not adapt to data path conditions. Data path conditions are guaranteed, but synchronization might be achieved by complementary, lower-bandwidth streams (e.g., audio).

TABLE III
RAN XHAUL REQUIREMENTS

Applications	*Latency*	*Jitter*	*Data rate*	*Reliable (%)*
ultra-low-latency, non-rate-adaptive (e.g. CPRI and eCPRI RAN Fronthaul) [5][29]	45-100us	65ns, 130ns, 260ns	NA	NA
low-latency, rate-adaptive (e.g. RAN Midhaul) [5]	1-10 ms	NA	NA	NA
low-latency, non-rate-adaptive (e.g. wireless road-side infrastructure backhaul) [30]	30ms	NA	NA	99.9999

(NA stands for not available)

3) Radio Access Network (RAN) Xhaul, used in communication class iii) is a category where all uses cases need high throughput, typically in Gbps range (Table III).

Many of these diverse, high demanding services and increasingly demanding traffic flow requirements will need to be simultaneously supported on a single infrastructure. However, fulfilling one requirement can prevent the network from meeting another. Can URLLC be fulfilled in the presence of high endpoint/flow density and/or large service/coverage areas, e.g. safety heartbeat? Can high data rates with low latency be met at high speeds, video-assisted remote drone control? Can ultra-low latency be maintained in the presence of high data rates, e.g., haptic feedback for remote surgery?

IV. TECHNOLOGIES FOR A DETERMINISTIC DIGITAL INFRASTRUCTURE

This section does not address all aspects of deterministic performance but focuses on strict end-to-end real-time guarantees, requiring hardware-based time synchronization as specified by Industry 4.0 standards and shown in the leftmost part of Fig. 2. Such guarantees need to be discriminated down to every application flow which requires it, but they concern a relatively small portion (typ. $<10\%$ of total traffic) of the flows. Alternative levels of less stringent determinism are not tackled in this paragraph. These still provide determinism in a sense of guaranteed packet delivery and bounded latency, but with at least two orders of magnitude larger jitter margins. As they stay in scope of regular store-and-forward and queuing operation, we subsume them in the following under the best effort class, well accepting the multifold shades of determinism in it. Future networks will likely need to be a combination of real or pseudo circuit switched connections and packet switched connections through the IP and optical end-to-end transport. For seamless interworking of the different forwarding principles and degrees of determinism we assume an elaborated orchestration and control at layer 3 and above. Here we refer the reader to the work done in scope of the IETF DetNet workgroup [7].

However, no congestion nor packet-to-packet change of path should be allowed for the flows with highest priority traffic. Congestion and change of path create jitter, and ultimately packet loss. Hence, today's split of technologies, namely circuit switching for transport and packet switching for statistical multiplexing

TABLE IV
KEY DESCRIPTORS OF REFERENCE SCENARIO

	Specifications	
(a)	Support of both time-sensitiveness and statistical mux	yes
(b)	Ratio of time-sensitive versus best effort traffic load	~10%
(c)	End-to-end latency	~100 µs
(d)	End-to-end jitter	1-10 µs
(e)	Resource reconfiguration time	< 1ms
(f)	Number of competing time-sensitive flows per switching node	>100
(g)	Number of servers in each edge cloud	200-1000
(h)	End-to-end propagation distance	~50km (creating 250µs latency)
(i)	End-to-end packet loss rate	$<<10^{-9}$

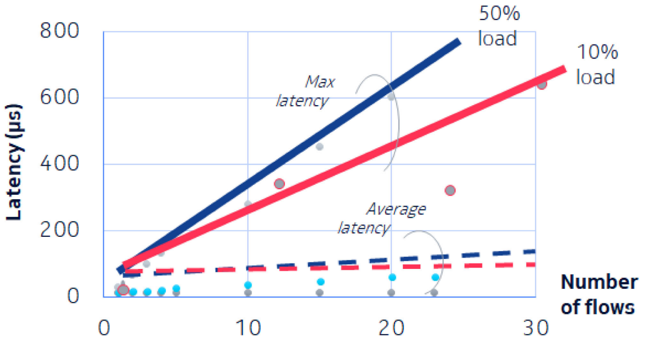


Fig. 5. Measured average and peak latency of 2.5 million packets from 5 concatenated Ethernet switches for data-centers, versus the number of competing input flows, for two loads per output port of 10% and 50%. All flows are assumed of the same class (e.g. time-sensitive). The dots are actual measurements, while the lines are best fits.

and aggregation at the edges (including inside data centers) is being challenged. Over-dimensioning switching capacity, i.e. performing statistical multiplexing at low load, as exemplified in today's central data centers, alleviates the probability of congestion and decreases jitter but cannot contain it when the number of priority flows competing for the same output exceeds a small amount, e.g. a few tens. Reserving bandwidth over some circuit container, as in today's optical transport networks, seems like a natural work-around to process priority flows but unlike today's optical networks, it must be end-to-end and for any input to any output of the network and discriminated per application flow. This requirement will drive the cost very high unless the circuit containers are reconfigured nearly as quickly as flows come and go. Overall, the optimal digital infrastructure based on a distributed edge cloud network should provide the jitter of circuit switching with the dynamics of statistical multiplexing.

To discuss the best ways to approach this goal, we use a reference industrial scenario based on the expectations drawn in the use cases of the previous section. Table IV summarizes the expected criteria for this scenario

Table V benchmarks existing layer 1-Layer 2 technologies against the requirements of Table IV, pointing to their strengths and main limitations. Interestingly, DDN does not have to rely on a single transport and switching technology across all segments but the hard challenge is to have the end-to-end concatenation of all technologies meet the criteria of Table IV.

Ethernet is undoubtedly the most successful implementation but suffers from unbounded jitter. Fig. 5 shows the measured average and peak latency of 2.5 million packets from 5 concatenated Ethernet switches for data centers, versus the number of competing input flows, totalizing 10% load or 50% load. At constant load, average latency increases weakly when the number of competing flows increases, but peak latency (hence, jitter) grows rapidly. Large queueing delays may be rare events (outliers) but they will happen owing to statistical multiplexing, irrespective of network load and the target packet loss rate (i) of Table IV cannot be met.

Circuit switching, e.g. OTN in its most successful form, whether paired with FlexE or not, has prevailed over years as natural tool to allow for deterministic performance, especially in long haul networks. Each data stream needs to be allocated a dedicated, reserved set of network resources, even when the stream is interrupted, but can be easily sliced from the other stream with any risk of mutual influence. It requires relatively heavy signaling for service turn-up/tear down, which results in service turn-up times often well above the 1s time scale. They are not compliant with our scenario for DDN at the edge where the "service" becomes the application itself, failing criterion (e) of Table IV. However, where long reconfigurations times are acceptable, e.g. to interconnect edge clouds over long distances, OTN has the advantage of preserving end-to-end determinism.

When reviewing all the options on the table, we drew the conclusion that the common denominator for the new distributed edge cloud network should be *scheduled, slotted* and *synchronized/isochronous* (3S), across all network segments, e.g. access, wireless, optical (edge) transport and DC. Slotted networks use time division multiplexing (TDM) to temporally interleave data containers of fixed duration (slots), thereby allowing for synchronized or isochronous ultra-reliable delivery of time-critical applications. Trains of slots are like temporarily allocated circuits which can be reworked to host best effort (e.g. video) traffic opportunistically as easily as in today's IP/Ethernet. All the technologies in Table II except standard Ethernet are actually 3S, but they would need significant reworking to comply with the expectations of Table IV end-to-end. Some segments are already compliant, but generally require update or hardening up to the control servers themselves, while the overall combination of them should support end-to-end scheduling at scale.

Radio is a 3S technology and should be considered in end-to-end DDN networks, but in addition deserves renewed efforts to drive latency and jitter down (criteria (c) and (d)), while ensuring lower packet loss (criterion (i)).

Work-around approaches against the limitations of standard Ethernet have been implemented for supporting the determinism of time-sensitive traffic. They all rely on the introduction of time slots of fixed duration. For example, PONs could deliver deterministic performance in fixed bandwidth allocation (FBA) mode, once connectivity is established, but cannot guarantee when connectivity is granted, should multiple flows compete with simultaneous requests and connectivity is static. Hence,

TABLE V
CANDIDATE TECHNOLOGIES FOR THE FUTURE DYNAMIC DETERMINISTIC DIGITAL INFRASTRUCTURE

	Network segment				Total Switching capacity	Latency (excl. propagation)	Jitter	Resource re-configuration time	Hard Slicing	Guaranteed delivery?	Main limitations in Table 1.
	Acc	FH	FF	DC							
5G new Radio	✓	✗	✓	✗	~100 Gb/s	~1 ms (target)	~100 ns	Fast (ms-time scale)	Yes	No	*(c), (d), (i)*
Conventional Ethernet	✗	✓	✗	✓	Pb/s	~µs per hop	Unbounded	N/A	No	No	*(a), (d), (f), (i)*
Industrial Ethernet	✓	✗	✓	✗	Typ. Gb/s	0.1-100 ms long schedule; sub µs per hop	Tens of ns to µs time scale	Static	Yes	Yes, depending on class of service	*(a), (e), (f), (h)*
TSN Ethernet	✓	✓	✓	✓	Gb/s	Similar to standard Ethernet	ns time scale over a few flows	Fast (ms-time scale) but not deterministic	Yes	Yes, depending on class of service	*(e), (f), (i)*
PON (in FBA mode)	✓	✓	✗	✗	~100 Gb/s	~100 µs	Not applicable	Static (FBA)	Yes (per flow time-wavelength slots allocation)	Yes	*(a), (e)*
OTN / FlexE	✗	✓	✗	✗	Tb/s	Dominated by propagation	Jitterless (circuit switching)	Slow (second)	Yes (frame allocation, coarse multi-Gb/s granularity)	Yes	*(a), (e)*
Bell Labs DDN [8]	✗	✓	✓	✓	Pb/s	1-100 µs timescale	<100 ns	Fast (ms-time scale)	Yes (per flow time-wavelength slots allocation)	Yes	*Maturity*

Network segment = Access (Acc), Front-Haul (FH), Factory Floor (FF) or Data Center (DC)
TSN = Time Sensitive Networking according to IEEE 802.1; OTN = Optical Transport Network ITU G.709; FlexE = Flexible Ethernet from OIF; PON FBA = Passive Optical Network Fixed Bandwidth Allocation [31]

today's PONs fail criterion (e) for now. Industrial Ethernet technologies, referring to different field buses utilizing Ethernet to various degrees, were specifically designed for time-sensitive industrial applications. Their ability to be dynamically reconfigured is very limited and they can only sustain a few flows over kilometer distances, therefore failing criteria (e), (f) and (h). Even though, many of the approaches support best effort traffic along time-sensitive traffic over the same infrastructure, they fail to be interoperable with other best effort networks (criterion (a)).

It should be emphasized that the originality of DDN is to stretch the optimization of determinism and dynamics altogether. One of the oldest 3S technology, namely ATM, had been quite successful for performance and efficiency but failed on the dynamics of connectivity, whereas Ethernet/IP largely prevailed. Our DDN paradigm of a hybrid converged infrastructure supporting best-effort and dynamic determinism is a very new challenge.

TSN was invented for that purpose and is certainly the most advanced networking technology for DDN. TSN leverages time slots which may be preempted or reserved per class of service if time sensitive. It extends the Ethernet standard (IEEE 802.1) with real-time capabilities [26]. Key functions are time synchronization, traffic shaping including exclusive gating, prioritization, preemption, and configuration mechanisms for strict resource planning of all involved devices.

Unfortunately, TSN in its current form has some limitations and already a limited amount of time-sensitive flows might exceed its capabilities (failing criterion (f) and, a consequence, criterion (i) too), as discussed further in the next section, and reconfiguration is not guaranteed within a deterministic time (criterion (e)). These limitations stem from the IEEE 802.1Qbv/bu discrimination into 8 classes of services and from the global and tight synchronization requirements of all switches and connected devices. While standards like 802.1Qci (addressing per stream filtering and policing) and 802.1Qch (addressing cyclic queueing and forwarding) have been proposed for partly removing these shortcomings, the implementation could be complex.

As it is unlikely that TSN will spread across all segments of the digital infrastructure, any coexisting network technology needs interwork with TSN and preserve performance guarantees end-to-end, while end-to-end scheduling and control should be generic and open enough to support multiple vendors and technologies. All network segments should support time-sensitiveness with some 3S technology. When transiting from one segment to the next, gateways should use jitter compensation mechanisms and flow priority queueing, to contain any additional latency and jitter. The control plane of the end-to-end network should be preferably designed as hierarchical, with controllers per segment for local slot scheduling and an orchestrator. The orchestrator coordinates the local schedulers to provision the end-to-end paths. Detailed considerations on end-to-end control are addressed in the DetNet workgroup.

Other, more disruptive 3S networking approaches have been reported to cope with low-latency applications. They should also be considered against the challenges of deterministic networking. For example, optical slot switching was investigated as 3S technology to perform traffic engineering and prioritize time-sensitive traffic in a data center, [32]–[34]. It can produce high throughput over a very small form factor but requires very high speed opto-electronics, and therefore has not yet reached the level of maturity of electronics.

Another 3S network concept is Fusion, which implements a strict priority scheduling policy to enable a hybrid network where some traffic (requiring determinism) is circuit-switched and the remaining traffic (best effort) is packet-switched [35]. However, Fusion is not natively designed to provide determinism for low data-rate applications and is therefore constrained by the relatively small number of time-critical flows it can support, therefore failing criterion (f) in Table IV. The concept of time-shared optical network (TSON) shares similar advantages and drawbacks with Fusion [36].

While popular software-based slicing and protocol enhancements have their role to play, they cannot fulfill the most demanding cloud-based industry requirements. In contrast to with some popular trends in telecommunications, DDN needs a new split between the functions which can be advantageously centralized/virtualized and the functions which must stay local, acting directly on bits and packets. Timing management and jitter compensation will have to be mostly performed by the physical layer, while reservation of resources for scheduling should be largely off-loaded to local agents on firmware.

V. Overcoming TSN Scalability Limitations

TSN is among the most promising enabling technologies for the convergence of information technology (IT) and operation technology (OT) into a single infrastructure across manufacturing shop floors [37]. Some of the first released TSN switches use prioritization and preemption for protection of a class of flows against greedy (best-effort) IT traffic as recommended in IEEE 802.1Qbu, but they do not avoid collisions within that critical traffic class itself, and they only offer this service to a single class. Supporting more classes and achieving a reliable isolation of time-sensitive flows among themselves and others, can only be achieved through reservations and cyclic scheduled gating (IEEE 802.1Qbv). A network utilizing IEEE 802.1Qbv is rather flat, requires a global synchronization and schedule (Fig. 6, top). However, the interconnection of thousands of devices and multiple applications with strict and diverse (e.g. scaling over several orders of magnitude) timing requirements leads to a deterioration of the synchronization, unsolvable scheduling, and bandwidth issues on critical links.

Recently, we considerably alleviated these limitations by proposing a new architecture [27]. We proposed to partition the TSN network into domains and create a hierarchy with an optical backbone to interconnect these domains for a converged industrial network. The optical backbone which we refer to as Industrial Optical Ethernet (IOE) implements a novel shim layer in the Ethernet stack. As explained later, the backbone runs asynchronously to the clients/TSN input ports and can tunnel the traffic of multiple independent TSN networks together along with possibly heavy IT traffic. A TSN domain preferably covers all devices and controllers which can spread across locations that require strict scheduling and a common clock base by their combined application. For example, a domain could be a machine or a set of machines (e.g. a production belt) which collaborate and/or have their control functions virtualized in the Enterprise Datacenter. The optical backbone interconnects the multiple location of each TSN domain while removing the need for synchronization among the domains, among logically unrelated devices/applications. So, in the above example, the machine (or set of machines) does not have to be synchronized and commonly scheduled with all the other machines that it does not collaborate with and still shares the common network infrastructure with them. A TSN domain covers at least all devices and controllers that require strict scheduling by their combined application. Multiple cooperating end-devices sitting nearby each other forming a so-called TSN island can be connected to other islands at different locations by isochronous IOE paths (Fig. 6, bottom). DetNet workgroup proposed to interconnect these TSN islands by asynchronous traffic shaping. In contrast, we make sure that connected islands belong to the same synchronization domain whereby timing of inter-island flows is equally precise as for intra-island flows.

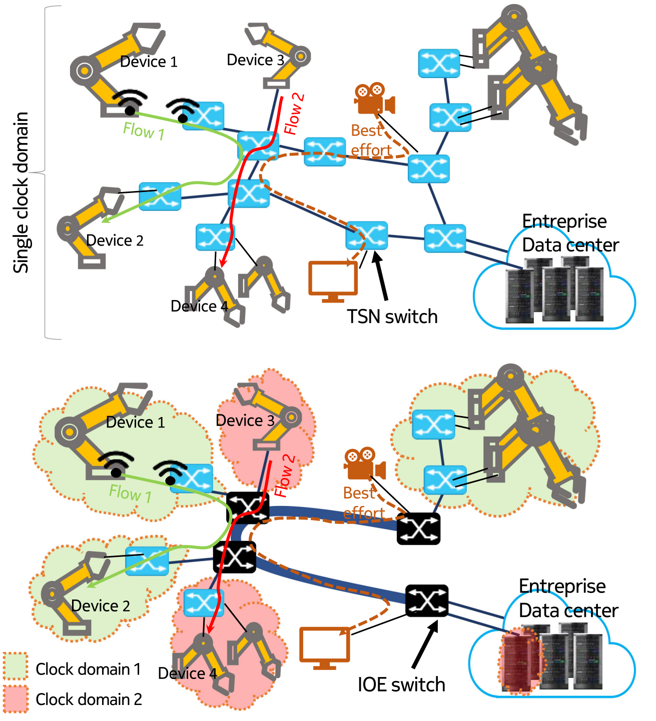


Fig. 6. Reference scenario of flat TSN and proposed hierarchical architecture.

We show the improved scalability by applying the proposed solution and an adequate scheduling algorithm we developed to big factory floors through network simulations. We also prove experimentally the feasibility of the proposed solution, while significantly improving our previous results [27]: we demonstrate the independent time synchronization of several TSN domains across the shared, yet asynchronous backbone, and report on the transparent transmission of tightly scheduled traffic over a tunnel.

A. *Fundamental Limits of Flow Gating*

Gated network operation requires a precise common network time base. The gates of each device are operated according to a globally coordinated schedule [38]. Synchronization is

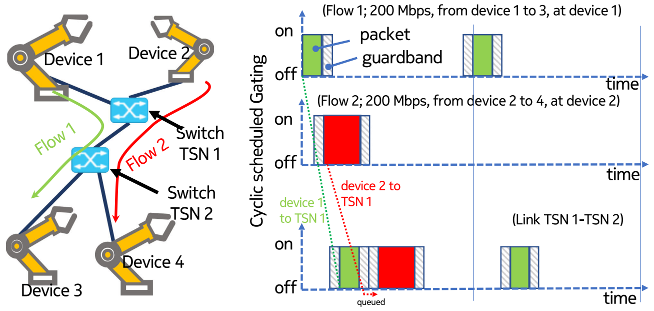


Fig. 7. Cyclic scheduling of 2 flows in a flat TSN-Qbv illustrating build-up of fragmentation and timing mismatch (requiring guard bands), hence limiting efficiency and network scalability.

achieved by means of appropriate protocols e.g. with IEEE 802.1AS standard [39] which is a variation of the Precision Time Protocol (PTP) for the ‘Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks’. Still, the imperfections of the clock oscillators, of the synchronization protocol, as well as the measurement inaccuracies (e.g. due to asymmetries) of the propagation times leave a certain timing mismatch. This forces the use of guard bands so that packets do not miss their allocated gates along their path, but at the expense of efficiency, as depicted in Fig. 7. As the network size and path lengths increase, timing mismatch increases [40]–[42], longer guard bands are needed and network efficiency drops. Moreover, higher data rates result in equivalently smaller packets. Thus, they require tighter gate timing and equivalent improvement in synchronization to achieve the same efficiency. Finally, cyclic scheduling of periodic flows is computationally intractable, that is NP-complete [38], [43]; serving flows with diverging size and cycles results in time fragmentation and requires to search among an exponential number of flow arrangements to be efficient.

B. Enhancing TSN Scalability by Creating Scheduling Domains

The proposed backbone is represented here by Industrial Optical Ethernet (IOE), which is an upgraded version of [44] for industrial networks. The IOE is a bus network but extendable to meshed topologies through cross-connects. Each IOE bit stream over one carrier wavelength is partitioned into container slots. While best effort client packets are aggregated into containers that have opportunistic access to free slots, time-sensitive client packets are served into reserved time slots over *isochronous* paths forming virtual circuits. For that, IOE defines a cycle of, e.g. 1000 slots on a 100 Gb/s link, and reserves as many slots as the ratio between client data rate and the slot capacity. Bookkeeping of reservations and admission control and dynamic establishment of new reservations are supported. For each time slot, a header is added to the data which contains control information (e.g., the reservation id that maps to routing and quality of service parameters common to all client packets carried in the corresponding time slot). A key difference between IOE and classical time division multiplexing (TDM) is the opportunistic use of time slots. IOE reserves slots for the deterministic flows, but also enables any node that comes later than the reservation source node to claim any reserved and empty slot to insert its own best effort traffic. Note that in IOE the transit traffic has strict priority over the inserted traffic at intermediate nodes.

The ingress flows are not synchronized to the IOE slots. The inevitable jitter at ingress to access the reserved slots is compensated at egress with an appropriate jitter compensation buffer. Note that DDN, and in particular IOE in this case, can encapsulate any upper layer protocol. To optimize throughput-efficiency during client data encapsulation into time slots, a segmentation and reassembly (SAR) mechanism is used together with the jitter compensation module. Isochronous reservations provide guaranteed packet delivery and deterministic latency with negligible additional jitter (ns range) [44]. Single packets, cyclic packet flows at any frequency and mix, or even aperiodic packet flows are carried “as is”, provided they do not exceed the reserved capacity. That is, the packets are tunneled and maintain their distances, with negligible jitter, as indicated in our results. The absence of timing restrictions for the incoming traffic implies that the timing on parallel reserved paths is independent of each other.

Following the above, we propose to use IOE’s isochronous reserved paths as tunnels for TSN traffic, including management and time synchronization (PTP) messages in order to create independent TSN domains. IOE proposal is an ultra-low-latency data plane that sits below existing standards (such as TSN and DetNet) in the layer stack, even below the Ethernet MAC. Our transport is agnostic of the higher layers which will make the integration in a multi-layered network particularly convenient and can take advantage of the latest advances reported in standards.

The proposed architecture implements the backbone for the TSN islands, forming a TSN/IOE hierarchy. With the proposed architecture, the network-wide schedule is broken down into independent schedules per domain, alleviating the bottleneck and enabling the strict time management of many more devices.

We next provide a quantitative evaluation of the benefits on the network scalability. We benchmark a (heuristic) scheduler for the flat TSN network and an end-to-end scheduler that we developed for the TSN/IOE network (consisting of an IOE scheduler and independent TSN heuristic schedulers per domain). The scheduler orders the flows in increasing period order, serves them one by one, keeping track of the buffer slot utilization per gate and searching for the first free allocation for the flow at hand.

We consider an industrial network with an increasing number of domains, each domain assumed to have 20 devices and a maximum network diameter of 5 hops. In each domain, we assume the uplink TSN switch of the domain to be connected to an IOE backbone switch in the case of TSN/IOE network, or to another TSN switch which acts as an aggregator in the reference scenario of the flat TSN network. We assume the domain controllers to be virtualized in a co-located enterprise data center (DC). For each domain we randomly select a cycle period and packet time from the set: {0.1ms/250Bytes, 1ms/1250Bytes, 10ms/5000Bytes}. We create flows so that devices communicate with their virtual

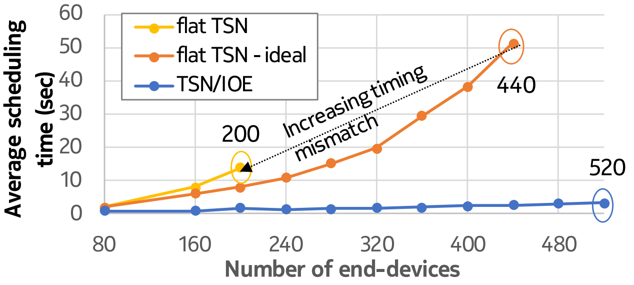


Fig. 8. Average scheduling time and maximum number of time-critical end-devices over a flat TSN and TSN/IOE networks.

controller and back and simulated 50 traffic instances for each set of parameters.

Regarding the timing mismatch / synchronization error, we take into account the IEEE 802.1AS standard for the "Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks" [39]. In IEEE 802.1AS any two time-aware devices with seven or fewer hops are required to be synchronized within 1μs. In the IEEE 802.1AS 2020 the synchronization bound of 1 μs is extended to more devices. A typical number for industrial networks is a chain of 50 devices, resulting in a per hop error of 20 ns. Ref. [42] studied the IEEE 802.1AS protocol behavior in large-scale networks while considering critical implementation details and concluded the feasibility for a chain of 50 hops to achieve synchronization within 1 μs.

Based on these we assume a synchronization error bounded within 100 ns for TSN network of 20 devices/ 5 hops (20 ns each) and hence, for each TSN/IOE domain where partitioning keeps domains small and timing mismatch low. This value is, however, very unrealistic ("ideal" case in Fig. 8) for a flat TSN with hundreds of devices and tens of hops where scheduling is performed with network-wide synchronization. Achieving strict synchronization at ns scale in such conditions would require an additional network such as white rabbit [41], or a multiplexing hierarchy and would be complicated and/or expensive. For this reason, we assume for the flat TSN a mismatch four times as large (400 ns) for a diameter four times as large (20 hops), and a number of devices ten times as large (>200). Fig. 8 reports the average scheduling time, as a function of the number of end-devices. The curves in Fig. 8 stop at the limit where one of the flows is blocked by the schedule because of a shortage of times slots to host it. This limit tells the maximum number of critical end devices that the network can manage.

From Fig. 8 we can see that flat TSN scales drastically worse than TSN/IOE. The average time to compute the schedule grows faster with the number of devices and is more impaired by fragmentation in the time domain. Fragmentation is caused by scheduling the flows of different periodicities and sizes, which eventually constrains the number of devices that can be scheduled. We see that even at the unrealistic/ideal timing mismatch of 100 ns, the flat TSN cannot exceed 440 devices whereas TSN/IOE supports 520 with the same timing mismatch. When compared to the heuristic used here for scheduling, ideal algorithms could potentially eliminate this difference but would require computation power largely in excess of today's computer capabilities [38]. Higher timing mismatches require larger guard bands and reduce the efficiency. For 400 ns timing mismatch, the number of served devices in the flat TSN network drops down to 200. The proposed TSN/IOE architecture relaxes these limitations. Its scheduler is still a heuristic but runs per domain where fragmentation is easily solved, making possible to manage 520 time-sensitive devices. The scheduling time in TSN/IOE grows linearly to the number of domains since it is calculated independently for each domain, while the global scheduling for flat TSN rises super-linearly with the developed heuristic (and exponentially with optimal algorithms). The only drawback is that IOE backbone introduces a slightly higher latency from its asynchronous operation, and the use of the jitter compensation buffer at the egress node. This additional latency is given by $T = S/b$, where S is the transport container/slot size and b the reservation bit rate. In our simulation study we chose a container/slot sizes of 1000 bytes, but this is a reference for a design parameter of the IOE protocol which can be reduced. Moreover, IOE can tradeoff latency for oversubscription per isochronous reservation [44], e.g. reserving double capacity for a connection would half the experienced latency.

C. *Proof of Concept of TSN Over an Industrial Optical Ethernet Backbone*

In this section we report on measurements and interoperability experiments that confirm the ability to support multiple independent TSN domains with the proposed solution. The experiments were based on the FPGA-based platform of [44] which performs isochronous path reservations on a time slotted optical bus, but it was modified with a smaller transport container (slot) size of 800 bytes instead of 9000 and a cut through add/drop pipeline.

In a first series of measurements we proved the strict mutual isolation between the concurrent isochronous path reservations of the BE flows. We do not reproduce the result here since they were obtained with minor changes with respect to [44]. and the best effort (BE) flows. There was no measurable impact on the path service quality, even in case of excessive overload

In a second step we investigated the latency and jitter of a reserved isochronous path. Fig. 9a shows the measured latency of a constant bit rate (CBR) flow of 64-byte packets carried over an IOE isochronous path with 2 slots in a cycle of 16 in a 10 Gb/s bus, and container sizes of 800 Bytes. The histogram in Fig. 9a corresponds to the measurements performed with the IOE architecture with 10 GE add/drop interfaces which involve store-and-forward for the rate conversion from the port rate (10 GE) to the reservation bit rate (1.2Gb/s) at the add and inversely at the drop. The right boxes report the results with the latest IOE implementation with cut-through add-drop port of 1 GE. For the rate conversion implementation, we found the mean latency to be 8.6 μs, the peak-to-peak latency to be 160 ns and the residual jitter to be 31 ns (standard deviation of latency). The primary source of latency is the jitter compensation mechanism at the egress port, where early-arriving packets are delayed so that all packets experience the same latency, which amounts to $T = S/b = 5.3$ μs. Other sources of latency are the transit node delay, as

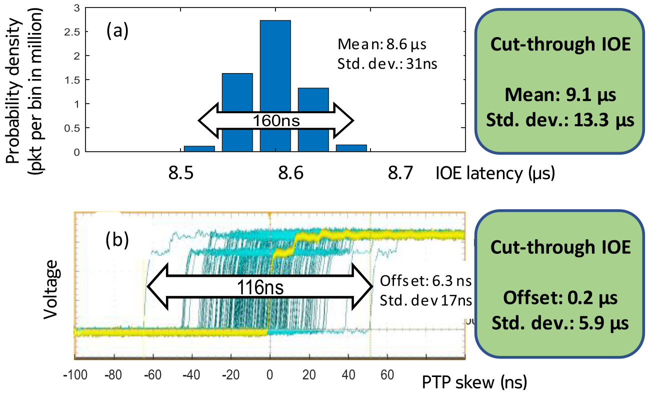


Fig. 9. IOE path latency/jitter and resulting clock alignment of two PTP synchronized TSN devices: (a) Tester histogram, (b) oscillogram of master (yellow) and slave (green) clocks pulses. Graphs report the measurements using the IOE implementation with rate converters in the add/drop ports while the green boxes report measurements using the updated IOE implantation with cut-through add/drop ports.

little as 500ns per node. With the cut-through implementation we observed a small increase of the mean latency to 9.1 ns, but a considerably smaller peak-to-peak latency and jitter.

We then performed interoperability tests with TSN equipment. The proposed architecture claims to synchronize and preserve timing properties of TSN 802.1Qbv flows at both ends of an isochronous path/tunnel. Hence assessing the compatibility and the accuracy of IEEE 1588 PTP protocol over an IOE path is an important prerequisite. We used two experimental TSN end-devices and connected them over an IOE path reservation or, alternatively, through a direct cable. Both devices were equipped with free running clock sources at 50ppm precision. Once connected, the PTP protocol elects one of them as master clock and synchronizes the other one, the slave, to the master. Fig. 9b shows an oscillogram of the two devices' hardware clocks. We report in the figure the measurements using the IOE rate converted add/drop ports and in the box the measurements with the cut-through IOE add/drop ports. In Fig. 9b we see that the subsequent slopes of the slave clock pulses were well aligned around the master clock pulse at mean offset of 6.3 ns and standard deviation of 17 ns (jitter). Note that the PTP messages were sent together with 500Mb/s background traffic. When using the cut-through IOE add/drop ports we observed a substantial reduction in the offset (0.2 μs) and of the jitter (5.9 ns). For reference, when both TSN devices were connected over a direct Ethernet Cat 6 copper cable the mean offset was 3.6 ns, and the standard deviation 3.9 ns. Overall, Fig. 9b shows that PTP effectively reduces the already small residual jitter of IOE seen in Fig. 9a. The transmission over the IOE isochronous path /tunnel on PTP had the same effect as a direct Ethernet cable. Note also that the IOE nodes, and hence the schedule on the bus, were not synchronized with the devices.

Next, we prove that we can support multiple independent TSN domains over a shared IOE infrastructure, as schematized in Fig. 6 bottom. We connect two pairs of TSN end-devices over two IOE reserved paths/tunnels formed over a common optical link (Fig. 10a). Each pair had its own time domain, i.e. own PTP instance and own clock master. Our oscilloscope recorded simultaneously all clocks of the four devices. As shown in Fig. 10b, we observed two well-aligned clock pairs and one pair randomly drifting against the other, proving independence of the two domains. As before, IOE and its schedule were not synchronized to any of the two clock masters.

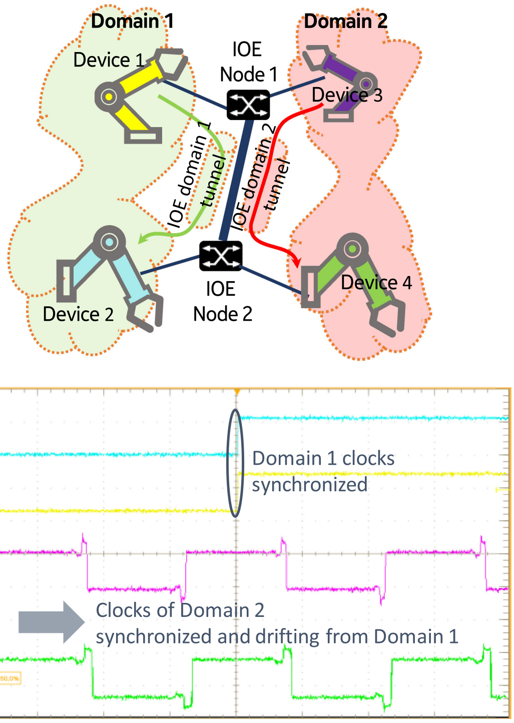


Fig. 10. (a) Multi-domain experiment: 4 TSN end-devices, forming two domains, each with two devices connected over a separate IOE isochronous path / tunnel. (b) Oscillogram of the four independent device clocks. The two clocks of each domain are synchronised, independently of the other domain.

Finally, to demonstrate the capability of IOE to tunnel TSN traffic and maintain TSN scheduling consistency, we injected two CBR flows from devices 1 and 2, with packet periodicities of 8 µs and 16 µs, respectively, assumed to belong to the same domain. We jointly scheduled them in a 16 μs cycle in two interconnected TSN switches. We measured the latency after the second TSN switch, while emulating a possible disparity between the schedules of the two switches, by forcing a variable time offset to the schedule of TSN switch 2 with respect to switch 1 (Fig. 11). We repeated the experiment after connecting the TSN switches over an IOE tunnel. As shown in Fig. 11 we found the timing properties of the two flows unchanged, as indicated by the latency patterns which are just shifted by the additional latency of IOE. IOE latency in this experiment was measured to be 14.1 µs; in addition to the 9.1 µs measured in Fig. 9a we also included media converters to/from the TSN switches that contributed $\sim$5 µs of latency. In Fig. 11 we also show the gating for the aligned schedules in both TSN and TSN/IOE networks. Overall, IOE tunnel performs like a cable with constant delay which can be precisely accounted for by the scheduler to achieve strict deterministic flow performance.

Overall, by partitioning TSN into scheduling domains and using our proposed IOE architecture as a backbone to tunnel

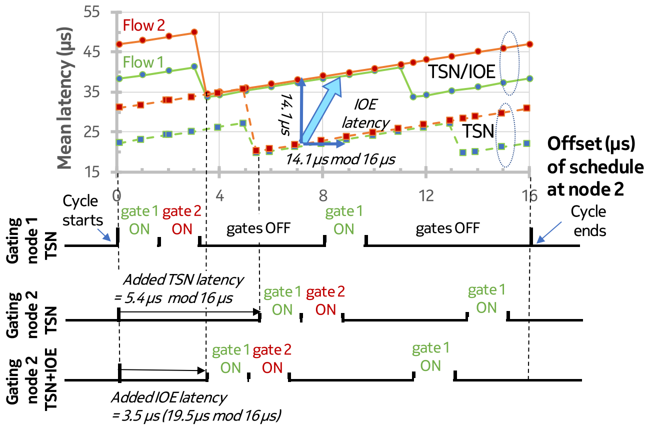


Fig. 11. Measured latency of two flows with 2 TSN switches directly connected or connected over IOE, as a function of TSN node 2 offset, and corresponding cyclic gating.

domains traffic, we avoid global synchronization and scheduling. We not only estimated that this approach can allow for a typical increase of the number of time-critical end-devices by +160% but also proved experimentally the feasibility of supporting multiple independent domains over a converged network infrastructure

VI. Conclusion

In this article, we reviewed the requirements for time-sensitive industrial use cases, highlighting the need for deterministic performance without giving up on dynamic turn-up and tear-down of the applications. While slotted scheduled and synchronized networking is the most promising approach to host such use cases, it is also the common denominator of a family of candidate technologies. They would require some enhancements to allow for cross-technology interoperability which preserves determinism end-to-end. Time sensitive networking (TSN) is well prepared for that goal but the accurate timing prevents the deterministic interconnection of more than a couple hundred of machines. We proposed and demonstrated an optical backbone that augments TSN, by slicing it into multiple islands of connected machines with independent timing.

Acknowledgment

The authors would like to thank Lars Dembeck for the fruitful discussions which influenced this work.

References

[1] E. Oztemel and S. Gursev, “Literature review of Industry 4.0 and related technologies,” *J. Intell. Manuf.*, vol. 31, pp. 127–182, 2020.

[2] M. Wollschlaeger, T. Sauter, and J. Jasperneite, “The future of industrial communication: Automation networks in the Era of the Internet of Things and Industry 4.0,” *IEEE Ind. Electron. Mag.*, vol. 11, no. 1, pp. 17–27, Mar. 2017, doi: 10.1109/MIE.2017.2649104.

[3] L. Geng *et al.*, “Problem statement of edge computing on premises for industrial IoT,” *Internet Eng. Task Force*, Mar. 2018.

[4] 5G-PPP, “5G and the factories of the future,” White Paper, 2015. [Online]. Available: https://5g-ppp.eu/wp-content/uploads/2014/02/5G-PPP-White-Paper-on-Factories-of-the-Future-Vertical-Sector.pdf.

[5] E. Grossman, “Deterministic networking use cases,” May 2019, IETF draft. [Online]. Available: RFC8578-DeterministicNetworkingUseCases(ietf.org)

[6] N. Finn, “Time-Sensitive and Deterministic Networking Whitepaper,” IEEE Mentor User Documentation, Jul. 2017. [Online]. Available: https://mentor.ieee.org/802.24/dcn/17/24-17-0020-00-sgtg-contribution-time-sensitive-and-deterministic-networking-whitepaper.pdf

[7] N. Finn, P. Thubert, B. Varga, and J. Farkas, “Deterministic networking architecture,” IETF draft, Oct. 2019. [Online]. Available: https://tools.ietf.org/html/rfc8655

[8] N. Benzaoui *et al.*, “Deterministic dynamic networks (DDN),” *IEEE/OSA J. Lightw. Technol.*, vol. 37, no. 14, pp. 3465–3474, Jul. 2019.

[9] K. Kitamura, F. Inuzuka, T. Tanaka, and A. Hirano, “Novel ODU4 path protection without bit disruption for low latency and resilient Core/Metro network,” in *Proc. Eur. Conf. Opt. Commun.*, Rome, 2018, pp. 1–3,

[10] S. Taherizadeh, V. Stankovski, and M. Grobelnik, “A capillary computing architecture for dynamic internet of things: Orchestration of microservices from Edge devices to fog and cloud providers,” *Sensors*, vol. 18, no. 9, 2018, Art. no. 2938.

[11] T. Benson, A. Akella, and D. A. Maltz, “Network traffic characteristics of data centers in the Wild,” in *Proc. 10th ACM SIGCOMM*, 2010, pp. 267–280.

[12] H. Z. Roy, J. Bagga, G. Porter, and A. Snoeren, “Inside the social network’s (Datacenter) network,” in *Proc. ACM Conf. Special Int. Group Data Commun.*, 2015, pp. 123–137.

[13] A. Rotem-Gal-Oz, “Fallacies of distributed computing explained,” *Doctor Dobbs J.*, Jan. 2008.

[14] S. Bigo, “Overturning the Eight fallacies of distributed computing with the octopus Edge network,” in *Proc. Opt. Fiber Commun. Conf. (OFC’20)*, San Diego, CA, USA, Mar. 2020, Paper Th3K. 2.

[15] M. Szczerban, S. Bigo, and N. Benzaoui, “On end-to-end latency variation compensation mechanism for time-slotted optical networks,” in *Proc. Eur. Conf. Opt. Commun.*, Brussels, Dec. 2020*,* Paper Tu1K-5.

[16] S. Sahoo, N.-H. Bao, S. Bigo, and N. Benzaoui, “Deterministic dynamic network-based just-in-time delivery for distributed edge computing,” in *Proc. Eur. Conf. Opt. Commun.*, Brussels, Dec. 2020, Paper Tu1K-5.

[17] T. Yang, R. Gifford, A. Haeberlen, and L. T. X. Phan, “The synchronous data center,” in *Proc. 17th Workshop Hot Topics Operating Syst. (HotOS ’19)*, Bertinoro, Italy, May 2019.

[18] A. Lechler and A. Verl, “Software defined manufacturing extends cloud-based control,” in *Proc. ASME Int. Manuf. Sci. Eng. Conf.*, Jun. 2017, vol. 3, Los Angeles, CA, USA.

[19] T. Doukoglou *et al.*, “Vertical Industries Requirements Analysis & Targeted KPIs for Advanced 5G Trials,” in *Proc. Eur. Conf. Netw. Commun. (EuCNC)*, Valencia, Spain, Jun. 2019, pp. 95–100.

[20] NGMN Alliance, “V2X White Paper,” v1.0, Jul. 2018. [Online]. Available: https://www.ngmn.org/publications/v2x-task-force-white-paper-v1-0.html

[21] 5G Americas, “5G Services & Use Cases,” 5G Americas White Paper, Nov. 2017. [Online]. Available: https://www.5gamericas.org/5g-services-use-cases/

[22] 3GPP, “Study on communication for automation in vertical domains,” 3rd Generation Partnership Project, Tech. Rep. TR 22.804, v16.3.0, 2020.

[23] 5G Alliance for Connected Industries and Automation, “5G for connected industries and automation,” Whitepaper, Second Edition, Feb. 2019. [Online]. Available: https://www.5g-acia.org/publications/5g-for-connected-industries-and-automation-white-paper/

[24] 3GPP, “Enhancement of 3GPP support for V2X scenarios,” 3rd Generation Partnership Project, Tech. Rep. TR 22.186, v16.2.0, 2019

[25] K. Antonakoglou, X. Xu, E. Steinbach, T. Mahmoodi, and M. Dohler, “Toward haptic communications over the 5G tactile internet,” *IEEE Commun. Surv. Tut.*, vol. 20, no. 4, pp. 3034–3059, Oct./Nov. 2018.

[26] IEEE Time Sensitive Networking Task Group. [Online] Available: http://www.ieee802.org/1/pages/tsn.html

[27] K. Christodoulopoulos *et al.*, “Enabling the scalability of industrial networks by independent scheduling domains,” in *Proc. Opt. Fiber Commun. Conf.*, San Diego, Paper Th2A.24, Mar. 2020.

[28] 5G Americas, “New services & Applications With 5G Ultra-Reliable Low Latency Communications,” 5G Americas White Paper, Nov. 2018. [Online]. Available: https://www.5gamericas.org/new-services-applications-with-5g-ultra-reliable-low-latency-communications/

[29] A. Nasrallah *et al.*, “Ultra-low latency (ULL) networks: The IEEE TSN and IETF DetNet standards and related 5G ULL research,” *IEEE Commun. Surv. Tut.*, vol. 21, no. 1, pp. 88–145, Oct./Nov. 2019.

[30] 3GPP, "Service requirements for the 5G system," 3rd Generation Partnership Project, Tech. Rep. TR 22.261, v16.13.0, 2020

[31] S. Bidkar, J. Galaro and T. Pfeiffer, "First demonstration of an ultra-low-latency fronthaul transport over a commercial TDM-PON platform," in *Proc. Opt. Fiber Commun. Conf.*, San Diego, CA, USA, 2018, Paper Tu2K-3.

[32] N. Benzaoui *et al.*, "CBOSS: bringing traffic engineering inside data center networks," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 10, no. 7, pp. 117–125, Jul. 2018

[33] P. Bakopoulos *et al.*, "NEPHELE: An end-to-end scalable and dynamically reconfigurable optical architecture for application-aware SDN cloud data centers," in *IEEE Commun. Mag.*, vol. 56, no. 2, pp. 178–188, Feb. 2018

[34] X. Xue *et al.*, "SDN-controlled and orchestrated OPSquare DCN enabling automatic network slicing with differentiated QoS provisioning," *J. Lightw. Technol.*, vol. 38, no. 6, pp. 1103–1112, Mar. 2020

[35] R. Veisllari, S. Bjornstad, J. P. Braute, K. Bozorgebrahimi, and C. Raffaelli, "Field-trial demonstration of cost efficient sub-wavelength service through integrated packet/circuit hybrid network [Invited]," *J. Opt. Commun. Netw.*, vol. 7, pp. A379–A387, 2015.

[36] G. S. Zervas *et al.*, "Time shared optical network (TSON): A novel metro architecture for flexible multi-granular services," *Opt. Exp.*, vol. 19, pp. B509–B514, 2011.

[37] D. Bruckner *et al.*, "An introduction to OPC UA TSN for industrial communication systems," in *Proc IEEE*, vol. 107, no. 6, pp. 1121–1131, Jun. 2019.

[38] S. S. Craciunas, R. S. Oliver, M. Chmelík, and W. Steiner, "Scheduling real-time communication in IEEE 802.1 Qbv time sensitive networks," in *Proc Int. Conf. Real-Time Netw. Syst.*, 2016.

[39] 802.1AS-2011 - IEEE Standard for Local and Metropolitan Area Networks - Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks.

[40] D. Fontanelli and D. Macii, "Accurate time synchronization in PTP-based industrial networks with long linear paths," in *Proc. IEEE Int. Symp. Precis. Clock Synchronization for Meas., Control Commun.*, 2010.

[41] F. Torres-Gonzalez, J. Dıaz, E. Marín-López, and R. Rodriguez-Gómez, "Scalability analysis of the white-rabbit technology for cascade-chain networks," in *Proc IEEE Int. Symp. Precis. Clock Synchronization for Meas., Control, Commun.*, 2016.

[42] M. Gutiérrez, W. Steiner, R. Dobrin, and S. Punnekkat, "Synchronization quality of IEEE 802.1AS in large-scale industrial automation networks," in *Proc. IEEE Real-Time Embedded Technol. Appl. Symp.*, 2017.

[43] K. Jeffay, D. F. Stanat, and C. U. Martel, "On non-preemptive scheduling of periodic and sporadic tasks," in *Proc. IEEE Real-Time Syst. Symp.*, 1991, pp. 129–139

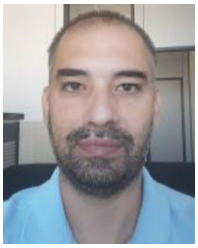
[44] W. Lautenschlaeger, L. Dembeck, and U. Gebhard, "Prototyping optical ethernet—A network for distributed data centers in the edge cloud," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 10, no. 12, pp. 1005–1014, Dec. 2018.

Sébastien Bigo (Fellow, IEEE) received the graduate degree from the Institut d'Optique Graduate School, Palaiseau, France, in 1992 and the Ph.D. degree in physics in 1996. In 1993, he joined Nokia Bell Labs (Alcatel Research & Innovation at the time),

After his Ph.D. on all-optical processing and soliton transmission, he studied high-capacity WDM transmission systems, demonstrating 30 record experiments in a row, at 10Gbit/s, 40Gbit/s, 100Gbit/s, and 400Git/s channel rates. He is currently the Director of the IP and Optical Networking Research Group. He has authored and coauthored more than 360 journal and conference papers, and 46 patents. His research interests include automated, dynamic, elastic optical networks.

He was the Bell Labs Fellow in 2012. He was the recipient of the General Ferrié Award in 2003 from the French ICT Society, the IEEE/SEE Brillouin Award in 2008, two Chéreau-Lavet Inventor-Engineer Awards in 2010 and 2017, and the IMT Grand Prize of the Academy of Science of France in 2017.

Nihel Benzaoui received the Engineering degree from the Institut National des Telecommunications et des Technologies de l'Information et de la Communication, Oran, Algeria, in 2010, the specialized master's degree in network architecture and design from Telecom ParisTech, Paris, France, in 2012, and the Ph.D. degree from Nokia Bell Labs, Nozay, France, in 2015. She was appointed as a Permanent Researcher Engineer in 2015. Her current research interests include architecture design and evaluation of multilayer mechanisms for optical networks in support for time sensitive 5G networks and beyond.



Konstantinos Christodoulopoulos received the Diploma from the School of Electrical and Computer Engineering (ECE), National Technical University of Athens (NTUA), Athens, Greece, in 2002, the M.Sc. degree in advance computing from the Imperial College of London, London, U.K., in 2004, and the Ph.D. degree from Computer Engineering and Informatics Department (CEID), University of Patras (UoPatras), Patras, Greece, in 2009. He is currently a Researcher with Nokia Bell Labs, Stuttgart, Germany. He was an Adjunct Assistant Professor with CEID-UoP and a Senior Researcher with Computer Technology Institute (CTI), Greece, ECE-NTUA, Trinity College Dublin, Ireland, and IBM Dublin, Ireland. He has authored or coauthored more than 100 peer-reviewed journal and conference papers and has more than 3000 citations. His research interests include optimization and control of datacom and telecom optical networks. He was an Associate Editor for the IEEE/OSA JOURNAL OF OPTICAL COMMUNICATIONS AND NETWORKING and in various conference committees.

Ray Miller received the Electrical Engineering degree from Rutgers University, New Brunswick, NJ, USA, in 1987. He joined Bell Labs, Stuttgart, Germany, in 1997 and has worked on advanced systems and architectures with responsibilities and duties in a wide range of telecommunications technologies, including core optical systems, metro Ethernet systems, and 4G or 5G wireless systems. He is currently a Member of the End-to-End Network and Service Automation Research Lab. His research interests include the definition and optimization of 5G services and deterministic time sensitive networks

Wolfram Lautenschlaeger received the Diploma degree in electrical engineering from the St. Petersburg Electrotechnical University "LETI," Saint Petersburg, Russia, in 1982, and the Dr.-Ing. degree from the University of Public Transportation and Communication, Dresden, Germany, in 1989. Since 1993, he has been with Nokia Bell Labs, Stuttgart, Germany. His research interests include fiber optical transmission and networking aspects, ranging from optical burst switching and frame layer topics up to traffic statistics, dimensioning, TCP, and queue management.

Florian Frick graduated with a double-degree from the University of Stuttgart, Stuttgart, Germany, and Telecom ParisTech, Paris, France. He is leading the Group Real-Time Communication and Control Hardware with the Institute for Control Engineering (ISW), University of Stuttgart. His research interests include industrial communication, converged networks, and especially on TSN. He is also active in multiple organizations and initiatives as well as leading the IIC-TSN-Testbed.