Data Centers Manufacturing Steel: Rethinking Industrial Networks in the Age of IT

Nikolaos Mitsakis[1,2], Marco Reisacher[1,3], Matthias Eichholz[1], Yannic Breiting[1], Harald Albrecht[1], Andreas Blenk[1], Oliver Hohlfeld[2]
[1]Siemens AG [2]University of Kassel [3]University of Stuttgart

Abstract

Industrial networks are undergoing a radical shift from closed, static OT environments towards open networks that integrate IT and OT. This shift applies IT operation principles to OT environments, such as virtualizing Programmable Logic Controllers and using Artificial Intelligence to increase production and process efficiency. While there is a huge effort to integrate IT principles, this paper demonstrates that IT/OT convergence remains an underexplored area of research, leaving out critical research opportunities for future networking systems. We identify three core challenges: timing constraints, service availability, and changing network traffic characteristics. For each challenge, we provide a concrete use case that demonstrates early findings and opens up new avenues for research within SIGCOMM.

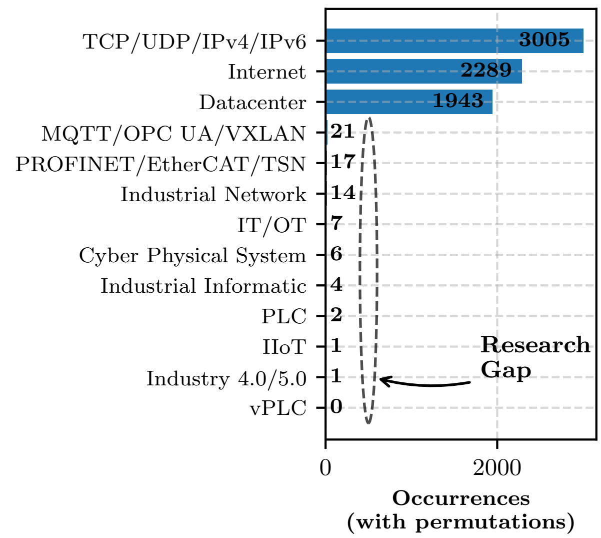


Figure 1: Industrial networking terms are underrepresented in recent SIGCOMM and HotNets proceedings.

CCS Concepts

• **Networks** → **Data center networks**; **Cyber-physical networks**; • **Computer systems organization** → **Dependable and fault-tolerant systems and networks**.

Keywords

IT/OT Convergence, Virtual PLC, ML for Manufacturing

ACM Reference Format:
Nikolaos Mitsakis, Marco Reisacher, Matthias Eichholz, Yannic Breiting, Harald Albrecht, Andreas Blenk, Oliver Hohlfeld. 2025. Data Centers Manufacturing Steel: Rethinking Industrial Networks in the Age of IT. In *The 24th ACM Workshop on Hot Topics in Networks (HotNets '25), November 17–18, 2025, College Park, MD, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3772356.3772407

1 Introduction

A Data Center in Steel-Toed Boots. Industrial networks are on the brink of a radical transformation. For the first time in history, they are converging with modern data center infrastructure—pushing computation, control, and communication into a virtualized and cloud-native domain. This shift, driven by the convergence of Information Technology (IT) and Operational Technology (OT), opens up a unique opportunity: to apply decades of advances in data center networking, virtualization, traffic engineering, and distributed systems to the physical backbone of manufacturing, energy, and critical infrastructure. The SIGCOMM community, with its expertise in building robust, scalable, and efficient networked systems, is perfectly positioned to shape the next generation of industrial networks—if it chooses to engage.

The Research Opportunity. IT/OT convergence is transforming factories into software-defined, data-driven environments—but today's networking and software stacks are not ready (see § 2). Industrial networks impose strict availability and real-time guarantees: delays or jitter can halt production or damage machinery. Unlike cloud or enterprise networks, even short communication stops can have physical consequences, demanding ultra-high availability. Their network topologies are also fundamentally different—decentralized, tightly coupled to physical layouts, and not optimized for fat trees or Clos designs. At the same time,

This work is licensed under a Creative Commons Attribution 4.0 International License.
HotNets '25, College Park, MD, USA
© 2025 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2280-6/25/11
https://doi.org/10.1145/3772356.3772407

this shift opens the door for the SIGCOMM community to apply proven ideas from data center networking. In future factories, production cells may be designed and networked much like data center pods—modular, scalable, and software-defined. IT/OT convergence reveals profound, unresolved challenges at the intersection of real-time communication, resilience, and topology design, presenting SIGCOMM with a compelling research agenda.

New Topic at SIGCOMM. We analyzed the full papers from recent editions of ACM SIGCOMM [2, 4] and ACM HotNets [1, 3], searching for terminology related to industrial networks and their communication protocols. As shown in Figure 1, the results are striking: virtually no papers in these top venues engage with the domain of industrial networking. Even the HotNets 2025 HotCRP omits industrial networks entirely from its list of topic areas while explicitly mentioning cellular, data center, enterprise, oceanic, social, and space networks. This silence is not a coincidence; it reveals a blind spot in our community. At a time when industrial networks are undergoing a once-in-a-generation transformation through IT/OT convergence, this lack of engagement risks leaving critical innovation on the table. Now is the moment for the SIGCOMM community to step up and bring its deep expertise to a domain that is ripe for disruption.

Contributions. This paper puts IT/OT convergence on the networking research agenda. We identify a critical gap: despite its growing relevance, this space remains largely unexplored by the SIGCOMM community. We outline key research challenges arising from strict timing requirements, extreme availability demands, and fundamentally different network topologies. To ground these challenges, we present three motivating use cases: (1) revealing the hidden non determinism in modern software stacks, demonstrated through `Traffic Reflection`—a method exposing timing variability in eBPF-XDP (§ 3); (2) the potential of programmable networks to boost availability by gracefully handling infrastructure changes such as VM restarts (§ 4); and (3) the clash between deterministic OT control loops and the non deterministic, data-hungry nature of AI workloads in automation (§ 5). Each case reveals concrete opportunities where networking research can make a lasting impact.

1.1 Present and Future Industrial Networking

What is an Industrial Network? An industrial control network (or simply an industrial network) is a specialized communication network that connects equipment for monitoring and controlling physical machinery and processes in industrial environments. Industrial networks have strict performance requirements, such as predictable, deterministic, and reliable communication. They must also fulfill demanding safety requirements that protect humans, machinery, and the environment. Often, separate dedicated safety networks and special safety protocols, such as PROFIsafe, are used for this purpose [41]. Industrial networks are the backbone of many sectors and are used, for example, in manufacturing, utilities (water, electricity, etc.), transportation, and even nuclear plants [41]. Our focus is on networks for factories.

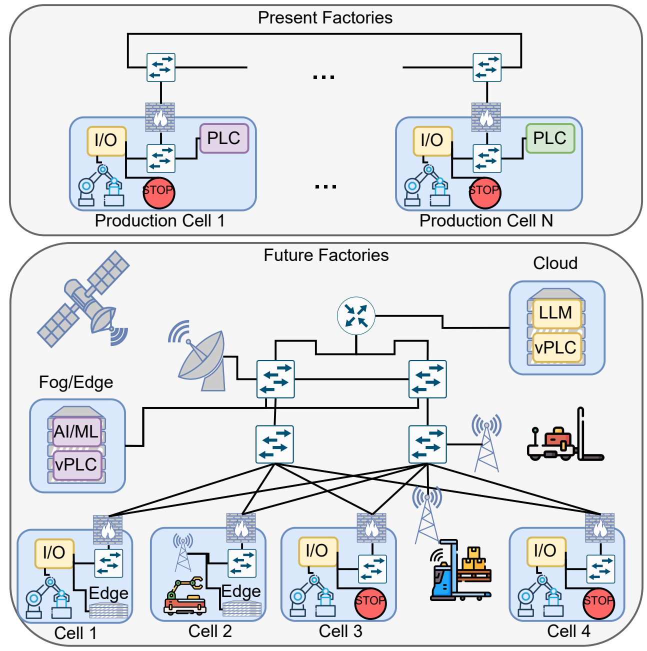


Figure 2: Factories are shifting from closed to open, adaptable systems that integrate IT and OT.

Present Factory. To understand what makes factories work, we look at the core of industrial automation: the so-called Programmable Logic Controllers (PLCs). Traditionally, factory floors rely on on-site control systems consisting of networks of PLCs connected to Input/Output (I/O) devices, responsible for collecting sensor readings and instructing actuators [10]. Meanwhile, Industry 4.0 envisions intelligent production environments that rely on Artificial Intelligence (AI) and are, in addition, flexible and adaptable [7, 49, 56]. Flexibility and adaptability can mean that when products are updated, production processes must also be updated accordingly. This happens today more frequently, thus requiring more changes.

Traditional hardware-based PLCs do not offer this level of flexibility, adaptability, and scalability [56]. This makes it challenging to accommodate frequent changes, future applications, and resource-intensive tasks [56], and the increasing demands of modern data-driven manufacturing [44].

Future Factory. To increase flexibility, adaptability, and scalability in factories, there is a growing interest in (see Fig. 2): (1) cloud computing and virtualization for industrial automation [18, 45, 82, 83, 101, 102, 104]; (2) virtual-PLCs [27, 43, 56, 89] running in a data center on off-the-shelf servers; and (3) intelligence in factory floors using AI, such as Large Language Models (LLMs), or Tiny Language Models (TLMs) for factory configuration and control tasks [59, 62, 64, 79, 111, 112]. These developments require the convergence of IT and OT, integrating hardware, software, emerging technologies, and physical processes. As a result, industrial networks must become flexible, presenting significant challenges [24]. However, automation applications increasingly demand deterministic networks [24], which contradicts traditional IT networks.

TSN in the wild. Time Sensitive Networking (TSN) was introduced to advance industrial networks. Traditionally, the industrial communication market has been dominated by multiple Ethernet-based systems that, despite having similar requirements, have different implementations and ecosystems, creating high costs and limiting the adoption of Industrial Internet of Things (IIoT) [20]. In response, TSN emerged to provide real-time communication, while addressing interoperability challenges [119]. TSN enables new configuration freedom. For example, by enabling the usage of arbitrary scheduling algorithms [95] that define pre-computed transmission schedules for pre-defined flows to meet real-time requirements. While TSN provides new freedom, it does not address all open problems. Problems such as faulty packets—those lost or arriving outside scheduled times–can lead to delays and switch congestion [33]. Moreover, configuring TSN switches is complex, requiring schedules to be set up initially and adjusted as devices change within the network [96]. Furthermore, TSN does not inherently address the challenges that arise from the convergence of IT and OT systems.

2 Research Challenges

2.1 Timing: Going Down to 1 µs Jitter

The first challenge involves realizing (virtualization) stacks that meet strict timing requirements with very low jitter.

Requirements. Manufacturing and process automation systems both have specific timing requirements [67]: machine tools demand cycle times as low as 500 µs [37], while high-speed motion control, e.g., for battery manufacturing, require latencies as low as 250 µs [42, 75] and jitter less than 1 µs [42]. In contrast, process automation can handle more relaxed cycle times, typically ranging from 10 ms to 100 ms [42]. To meet such strict timing demands, specialized dedicated hardware is used; for example, using Application-Specific Integrated Circuits (ASICs) [11], or Field Programmable Gate Arrays (FPGAs) [28]. In contrast, vPLCs do not rely on special hardware but instead rely on host networks and virtualization stacks.

Current stacks do not meet these requirements. First, host networks introduce various contention sources that impact application performance [107]. Poor coordination among processors, memory, and peripheral interconnects creates contention [107]. Additionally, IO memory management can reduce the available NIC-to-CPU bandwidth, resulting in hundreds of microseconds of delays and packet drops [8]. This contention is potentially further intensified by the growing complexity of modern data center host networks [107].

Second, PCIe, the de-facto standard I/O interconnect used in servers, has a heavy latency toll for small packets—common in industrial automation—contributing to more than 90% to the overall NIC latency [9, 77].

Third, multiple flows sharing host resources, such as the same NUMA node or packets from different flows arriving at the same NIC, lead to increased packet processing overhead [22]. Even mixing long and short flows on the same CPU core leads to lower per-core throughput [22].

Prior work does not achieve OT requirements. Recent work redesigns host network stacks to isolate latency-critical traffic, achieving microsecond-scale latencies at the 99.9^{th} percentile [23]. However, they do not consider OT requirements, such as even stricter timing requirements or tiny packet payloads, down to 20 bytes (see § 2.3).

Receiving a packet from and sending a packet to a NIC are just two factors; the kernel also impacts real-time application performance. Although dual-kernel solutions typically outperform Linux with the PREEMPT_RT patch [84], they increase complexity by requiring applications to use specialized dual-kernel system calls [84]. On the other hand, Linux kernels with the PREEMPT_RT patch cannot be considered hard real-time—particularly in safety-critical scenarios—due to unpredictable kernel-induced latencies [84].

Existing work evaluating robot control with Robot Operating System (ROS)-2 [81, 116]—software tools and libraries for robot control—on Linux (with the PREEMPT_RT patch) or vPLCs [39, 40, 47, 56] have several limitations. They either do not perform independent evaluations over extended periods of time, or collect insufficient data to draw conclusions.

Additionally, they only consider basic application scenarios, such as simple ping-pong tests. They do not evaluate realistic industrial automation applications, e.g., a production line or robot arms. To make things worse, existing work also often fails to report critical performance metrics such as jitter and worst-case latency/jitter. Importantly, consecutive jitter events—periods where jitter repeatedly occurs cycle after cycle—or bursts are not reported. This, however, is critical because industrial devices halt operation (for safety reasons) when no packets arrive for several consecutive cycles (known as a watchdog counter expiration in PROFINET [14]). Finally,

current evaluations omit how systems scale; e.g., how performance changes when multiple robot applications, vPLCs, or other sources of network traffic are running simultaneously.
Research challenge. Industrial control systems require ultra-low jitter and tightly bounded latency—properties at odds with cloud-native and virtualized networking stacks. As virtual PLCs move into containerized environments, the lack of timing guarantees across software layers and virtualization boundaries becomes a major challenge. These stacks often fail to deliver or even expose their timing behavior, and measuring microsecond-level jitter is non-trivial. We propose `Traffic Reflection`, a diagnostic technique that reveals hidden timing drift in eBPF/XDP pipelines (§ 3)—a first step toward meeting real-time demands in future stacks.

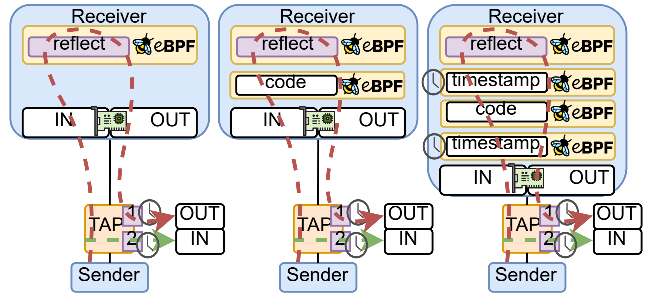


Figure 3: Benchmarking methodology for measuring the timing overhead introduced by eBPF code.

2.2 Service Availability: ≥ 99.9999

Requirements. For industrial automation, high-availability is an absolute necessity [7]. With use cases such as motion control, mobile robots, and process monitoring requiring extreme service availability—at least 99.9999 [5, 42]. This corresponds to a downtime of less than 31.5 s per year [42]. In process automation, even if a failure occurs, operation must continue in a safe mode. In contrast, despite the importance of networking, failures happen regularly in data centers, which typically aim for monthly downtime of a few minutes, potentially multiples of 31.5 s [46, 66]. The reliability of data center networks heavily depends on fiber links. These fiber links differ widely in reliability, with large variations in how often they fail (mean time between failures) and how quickly they can be repaired (mean time to repair) [72].
Research challenge. Achieving this level of availability in virtualized, cloud-native environments poses a fundamental challenge. Classical OT systems rely on distributed, fault-tolerant architectures where automation cells can continue operating independently during failures. However, consolidating virtual PLCs in centralized data centers increases potential for failures: even a short-lived outage can simultaneously affect dozens of production cells. Current virtualization stacks are not designed to isolate or gracefully recover from such disruptions in real-time control contexts. In § 4, we explore how programmable networks can help restore fault containment by enabling seamless switchovers and minimizing disruption during infrastructure failures.

2.3 New Traffic Mix: Deterministic Never-Ending Microflows at Scale

Requirements. Industrial networks are built for predictability and determinism. Their topologies—line, ring, star, or tree—are carefully engineered to support a fixed number of devices (e.g., switches, I/O devices, and PLCs) with guaranteed timing, and remain largely static after commissioning [41]. Time-critical traffic involves small, periodic packets with tight latency and jitter constraints. Time-critical traffic varies from very short cycle times ($< 2\,\mathrm{ms}$) with small payloads ($20 - 50$ bytes), to slightly longer cycles ($1 - 10\,\mathrm{ms}$) and larger payloads (40 to 250 bytes) [5].

In contrast, data center networks today feature highly diverse and dynamic topologies, such as Clos, fat trees, leaf-spine, BCube, or DCell. Data center topologies might have to change on-the-fly to best meet traffic type and demand [48]. Additionally, data center traffic is inherently diverse [48], although the majority of flows tend to be relatively small in size and typically last only a few milliseconds [15]. Flows within data centers can be categorized (with slight variations in definition across previous work) into three general types: (1) Small flows (mice flows), typically short in duration, latency-sensitive, and small in size—around 5KB [48] or less than 10KB [114], (2) Medium flows, typically around 0.5MB [48], (3) Large flows (elephant flows), flows that are large in size, larger than 1 GB [48]. However, with vPLCs, a new type of flow appears. This new flow type blends characteristics of existing categories: it exhibits both latency sensitivity (part of 1) and continuous traffic (part of 3). Specifically, it is cyclic, with the transmission of small packets, strict deterministic timing requirements, and is never-ending.
Research challenge. This emerging class of never-ending, deterministic micro-flows creates a mismatch with data center traffic engineering practices, which are typically optimized for flow completion and throughput [6, 12, 32]. Additionally, dynamic topologies and link aggregation mechanisms may further disrupt the timing properties critical to vPLC operation. In § 5, we discuss how these network mismatches complicate the co-existence of latency-critical control and data-intensive ML workloads—an increasingly common scenario in next-generation industrial automation.

3 Revealing eBPF's Hidden Delays

To take a first step toward addressing the timing challenges outlined in § 2.1, we propose `Traffic Reflection`, a lightweight measurement method to reveal timing drift and nanosecond-level jitter in eBPF/XDP-based packet processing.

Context. XDP and eBPF are used to address timing challenges, such as those described in § 2.1, by moving packet processing closer to NICs [54, 60, 91, 93, 115]. XDP's and eBPF's main advantages are that they provide a flexible and safe programmable environment within the Linux kernel [55, 105]. While XDP and eBPF try to avoid non-deterministic behavior, for instance, by restricting floating-point arithmetic [78] there are no guaranteed latency and jitter upper bounds. Such guarantees however, are essential in industrial automation.

Previous work studied the performance impact of using XDP and eBPF for packet filtering [54, 60, 91, 93], forwarding latency [13, 55, 99], and system overhead [106]. Measuring the exact timing overhead (jitter and latency) introduced by eBPF is challenging [106], but an absolute necessity to guarantee the timings in § 2.1.

Contribution. `Traffic Reflection` reveals the non deterministic behavior of eBPF and XDP applications. It uses a network tap and a "reflection point" in the eBPF code (see Fig. 3). This has the advantage that all packet capture timestamps come from a single clock (the tap's clock), avoiding measurement errors caused by clock synchronization problems. Even though IEEE 1588 Precision Time Protocol (PTP) can achieve synchronization accuracy below 1 µs, it encounters challenges related to asymmetric delays and network inconsistencies, affecting precision [63]. Note that the network taps have their own timestamping precision, which is acceptably low with 8 *ns*. Additionally, our test end-hosts have the Linux PREEMPT_RT patch.

We evaluate six eBPF programs running in XDP native mode to demonstrate unpredictability when processing real-time TSN flows. Each program builds on a base version: (1) the base program reflects packets back to the NIC (Base), (2) adds one timestamp (TS), (3) adds two timestamps (TS-TS), (4) adds timestamps to a ring buffer (TS-RB), (5) adds timestamps into the packet's payload (TS-OW), and (6) adds the difference of two timestamps to the ring buffer (TS-D-RB). All results are presented as Cumulative Distribution Functions (CDFs). Our experiments show that (see Fig. 4): (1) small code changes in eBPF/XDP applications lead to noticeable delay differences; and (2) more real-time flows handled by eBPF/XDP lead to increased jitter.

Discussion. `Traffic Reflection` is a first step in understanding how specific eBPF code and execution environments relate to execution time, enabling the development of eBPF applications with predictable timing guarantees, even if it is not possible for the underlying kernel.

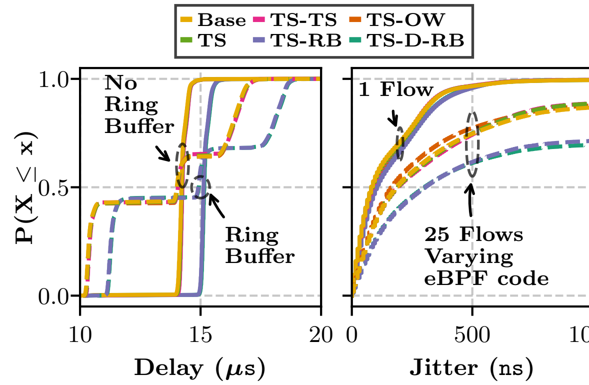


Figure 4: Even minor code changes in an eBPF application cause noticeable delays (left), while more real-time flows handled by XDP and eBPF increase jitter (right).

4 High-Availability with SDN

Industrial automation achieves the strict service availability requirements described in § 2.2 by using redundant PLC pairs: one active primary and one passive secondary on standby. If the primary PLC fails, the secondary PLC takes over, depending on manufacturer and device typically within 50 ms to 300 ms [98]. Note that this setup requires special hardware settings such as dedicated links between the PLC pairs for synchronization and heartbeats [98].

Existing work replicates this availability approach for vPLCs as standalone applications [58], or as Kubernetes pods [57]. However, existing work introduces high switchover delays (≈ 110 ms to ≈ 55.4 s) [57] or require three dedicated links between vPLCs for synchronization and preventing both controllers from simultaneously assuming the primary role [58].

Contribution. We propose `InstaPLC`, an in-network application that uses programmable networks to address the high-availability of vPLCs. Programmable networks have shown advantages in industrial control scenarios [61, 90, 103, 109], including enabling seamless vPLC migration [73]. `InstaPLC` enables seamless switchovers between vPLC pairs without requiring dedicated links between the vPLCs for synchronization.

`InstaPLC`, built with the Data Plane Development Kit (DPDK) Software Switch (SWX) pipeline [31] and P4 [16] for the data plane, and Python for the control plane, works as follows: When a vPLC attempts to connect to an I/O device, it is designated as the primary vPLC for this specific I/O device. It also adds table entries to the switch, allowing the vPLC and I/O device to establish a communication relationship. A communication relationship means that the vPLC configures what data is exchanged with the I/O device and how often. It also configures how long each device can continue working without receiving new data from each other. During this, InstaPLC extracts information from the exchanged packets and, with this information, creates a digital twin of the I/O device. Once the configuration is complete, real-time data transmission between the vPLC and I/O device begins.

If another vPLC subsequently attempts to connect to an already controlled I/O device, this new vPLC is designated as a secondary. The secondary vPLC establishes a communication relationship with the digital twin I/O device. From the perspective of the secondary vPLC, communicating with the digital twin is identical to communicating with the actual I/O device. After the configuration of the digital-twin I/O device, real-time communication flow through InstaPLC is as follows: (1) packets from the digital twin to the secondary vPLC are dropped at the switch; (2) packets from the secondary vPLC are forwarded to the digital twin only; (3) packets from the physical I/O device are forwarded to both primary and secondary vPLCs; this ensures both vPLCs know the exact state of the I/O; and (4) packets sent by the primary vPLC are forwarded directly to the physical I/O device.

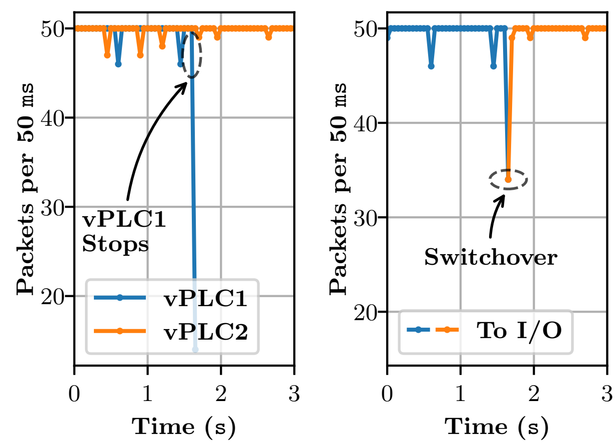
Finally, InstaPLC continuously monitors packets in the data plane. If the primary vPLC stops sending packets for a configurable number of I/O cycles InstaPLC dynamically triggers a switchover, in the data plane, handing control from the primary to the secondary vPLC (see Fig. 5).

Discussion. InstaPLC represents an important first step towards programmable networks that enable vPLC high availability. It opens new research directions on efficiently ensuring high availability across heterogeneous industrial environments, each with distinct communication protocols, configuration models, and timing requirements.

5 Towards Machine Learning Aware Industrial Networks

Finally, it is not only about virtualizing PLCs, but also about integrating new technologies such as Machine Learning (ML). More and more, PLCs and ML are expected to work together. For instance, PLCs are controlling a robot while receiving input from ML applications, e.g., detecting the objects moved by a robot. However, the networking requirements of both PLCs and ML stand traditionally in stark contrast leading to severe challenges [38, 113]. PLCs have cyclic, deterministic network traffic (PLCs), whereas ML produce data-intensive, non-deterministic, dynamic network traffic [21, 26].

ML is a key ingredient of advanced industrial automation and process optimization [97, 100], such as garbage sorting, bottle manufacturing, and vaccine production. ML supports automated optical inspection, quality assurance, Automated Guided Vehicles (AGVs) navigation, and data-driven digital twin analysis [51, 69, 80]. In short, (v)PLCs make decisions dependent on the outcome of ML. And the next technological leap is already knocking on the door with the evolution of



(a) From vPLCs. (b) To I/O.

Figure 5: InstaPLC: When the primary controller (vPLC1) for an I/O device fails, InstaPLC detects this, and dynamically switches to a backup controller (vPLC2). As a result, the I/O device remains controlled.

industrial applications of Generative Artificial Intelligence (GenAI), LLMs, and Agentic AI [71].

Research challenge. It is now up to operators to integrate virtualized control loops, which rely on ML, on converged IT/OT infrastructures. The problem is that ML inference in industrial settings can significantly suffer when exposed to network-induced data degradation, such as compression artifacts, frame loss, or jitter [86–88, 94]. These effects are especially pronounced in video-centric tasks (e.g., automated optical inspection), where model performance is tightly coupled to input fidelity [17, 53, 85]. Benchmarking these ML models against degraded input data confirms that model robustness is insufficient without a network-aware design, even for pre-trained models in well-known industrial scenarios, such as casting defect detection [29].

Recent data center networking architectures such as Fat-Tree, BCube, DUO, or optical variants like OSA and Rotor-Net [25, 34, 50, 70, 118] significantly advanced the performance of non-industrial applications. Furthermore, existing ML-focused data center optimizations primarily target training workloads; but they overlook inference requirements under volatile input and constrained edge or fog compute [38, 68, 76]. However, our observations indicate that these topologies do not meet the real-time and reliability constraints of ML inference in industrial environments [35, 113].

Contribution. We conduct a simulation-based optimization of 3 topologies: a classic industrial ring, a leaf spine, and a traffic-aware setup. The traffic input comes from analyzing ML models with degraded input data. The simulation provides insights on the latency of ML applications for an increasing number of ML clients. Fig. 6 shows how legacy

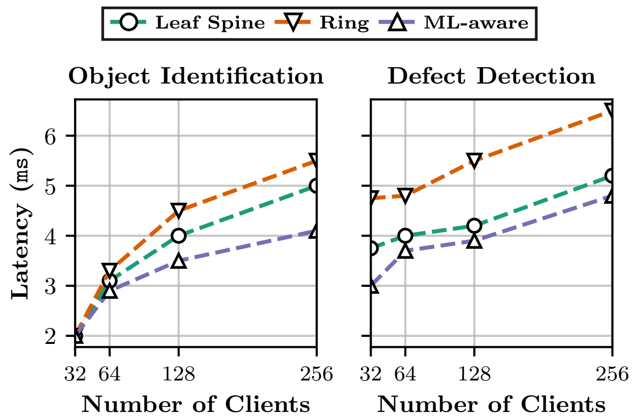


Figure 6: ML-aware topologies achieve the lowest latency for both defect detection and object identification compared to traditional IT and OT networks.

OT network topologies hurt latency for ML applications compared to modern IT derivatives. As expected, the ring shows the worse performance; but also the leaf spine can only slightly improve the performance. To combat this issue, the industrial ML-traffic-aware topology design shows the best performance. It takes volatile input and constrained edge and fog computing environments into account. The preliminary design aligns inference accuracy with infrastructure cost and network dimensioning.

Discussion. This case study highlights the fundamental tension between ML-driven workloads and the timing and reliability constraints of industrial networks. Our analysis focuses on topology design as a first-order lever to align infrastructure behavior with the inference performance of critical ML applications. While we illustrate the potential of traffic-aware designs, they represent only an initial step. The broader challenge lies in co-designing network topologies, ML model placement, and resource allocation strategies that meet industrial real-time, availability, and safety requirements.

6 Conclusion

Much like Ethernet and existing wireless technologies, they were developed for use outside of industrial networks and have no considerations for real-time response or determinism [41, 92]. Later, significant efforts were required to integrate or adapt these technologies for industrial use, such as 5G URLLC or TSN, with many challenges still open [52, 65, 74, 117, 119].

Today, history repeats itself with the convergence of IT and OT, exemplified by virtualization providers only now introducing industrial virtual switches [19]. However, in-depth performance evaluations of these virtual switches are publicly by far unexplored [19], in contrast to virtual switches for cloud networking [30, 36, 108, 110].

While the convergence of IT and OT is gaining momentum, fundamental challenges remain unresolved. In this paper, we present three concrete scenarios that expose critical integration gaps and outline pressing research opportunities. Today, industrial networks and IT/OT convergence are significantly underrepresented in the systems and networking research community, slowing innovation in this space. Our goal is to help change that: by highlighting these challenges and proposing initial solutions, we aim to pave the way for a broader research agenda toward designing the next generation of industrial networks—scalable, flexible, adaptive, and built to meet strict real-time and availability demands. This work does not raise any ethical concerns.

Acknowledgment

This work is partially funded by the Deutsche Forschungsgemeinschaft (DFG) in SPP 2378 (Resilient Worlds) project ReNO (grant number 511099228) and the German Federal Ministry of the Interior (BMI), represented by the Federal Agency for Public Safety Digital Radio (BDBOS), under the funding code 16BEC0020. Responsibility for the content of this publication lies with the authors.

References

[1] 2022. *Proceedings of the 21st ACM Workshop on Hot Topics in Networks.* Association for Computing Machinery.
[2] 2022. *Proceedings of the ACM SIGCOMM 2022 Conference.* Association for Computing Machinery.
[3] 2023. *Proceedings of the 22nd ACM Workshop on Hot Topics in Networks.* Association for Computing Machinery.
[4] 2023. *Proceedings of the ACM SIGCOMM 2023 Conference.* Association for Computing Machinery.
[5] 3GPP. 2020. *TR 22.804 V16.3.0 Study on Communication for Automation in Vertical Domains.* Technical Report.
[6] Vamsi Addanki, Oliver Michel, and Stefan Schmid. 2022. {PowerTCP}: Pushing the Performance Limits of Datacenter Networks. In *USENIX NSDI.* 51–70.
[7] Siemens AG. 2022. *Network concepts Factory Automation 109802750 V1.0.*
[8] Saksham Agarwal, Rachit Agarwal, Behnam Montazeri, Masoud Moshref, Khaled Elmeleegy, Luigi Rizzo, Marc Asher de Kruijf, Gautam Kumar, Sylvia Ratnasamy, David Culler, and Amin Vahdat. 2022. Understanding host interconnect congestion. In *ACM HotNets.* 198–204.
[9] Mohammad Alian and Nam Sung Kim. 2019. NetDIMM: Low-Latency Near-Memory Network Interface Architecture. In *IEEE/ACM International Symposium on Microarchitecture.* 699–711.
[10] Ephrem Ryan Alphonsus and Mohammad Omar Abdullah. 2016. A review on the applications of programmable logic controllers (PLCs). *Renewable and Sustainable Energy Reviews* (2016), 1185–1205.
[11] N. Aramaki, Y. Shimokawa, S. Kuno, T. Saitoh, and H. Hashimoto. 1997. A new architecture for high-performance programmable logic controller. In *International Conference on Industrial Electronics, Control,*

and Instrumentation. 187–190.

[12] Serhat Arslan, Yuliang Li, Gautam Kumar, and Nandita Dukkipati. 2023. Bolt: {Sub-RTT} Congestion Control for {Ultra-Low} Latency. In *USENIX NSDI*. 219–236.

[13] Joshua Bardinelli, Yifan Zhang, Jianchang Su, Linpu Huang, Aidan Parilla, Rachel Jarvi, Sameer G. Kulkarni, and Wei Zhang. 2024. hyDNS: Acceleration of DNS Through Kernel Space Resolution. In *ACM Workshop on eBPF and Kernel Extensions*. 58–64.

[14] Paolo Bellagente, Paolo Ferrari, Alessandra Flammini, Stefano Rinaldi, and Emiliano Sisinni. 2016. Enabling PROFINET devices to work in IoT: Characterization and requirements. In *IEEE Instrumentation and Measurement Technology Conference*. 1–6.

[15] Theophilus Benson, Aditya Akella, and David A. Maltz. 2010. Network traffic characteristics of data centers in the wild. In *ACM IMC*. 267–280.

[16] Pat Bosshart, Dan Daly, Glen Gibb, Martin Izzard, Nick McKeown, Jennifer Rexford, Cole Schlesinger, Dan Talayco, Amin Vahdat, George Varghese, and David Walker. 2014. P4: programming protocol-independent packet processors. *SIGCOMM Comput. Commun. Rev.* (2014), 87–95.

[17] Fabio Brau, Giulio Rossolini, Alessandro Biondi, and Giorgio Buttazzo. 2023. On the Minimal Adversarial Perturbation for Deep Neural Networks With Provable Estimation Error. *IEEE Trans. Pattern Anal. Mach. Learn.* (2023), 5038–5052.

[18] Hongyu Pei Breivold, Anton Jansen, Kristian Sandström, and Ivica Crnkovic. 2013. Virtualize for Architecture Sustainability in Industrial Automation. In *IEEE Computational Science and Engineering*. 409–415.

[19] Broadcom 2025. *VMware Cloud Foundation 9.0 Release Notes*. Broadcom.

[20] Dietmar Bruckner, Marius-Petru Stănică, Richard Blair, Sebastian Schriegel, Stephan Kehrer, Maik Seewald, and Thilo Sauter. 2019. An Introduction to OPC UA TSN for Industrial Communication Systems. *Proc. IEEE* (2019), 1121–1131.

[21] Philippe Buschmann, Arne Broering, Georg Carle, and Andreas Blenk. 2025. LLMs on Edge: Network Traffic Characteristics of Distributed Inference under the Loupe. In *Workshop on Networks for AI Computing*. 108–114.

[22] Qizhe Cai, Shubham Chaudhary, Midhul Vuppalapati, Jaehyun Hwang, and Rachit Agarwal. 2021. Understanding host network stack overheads. In *ACM SIGCOMM*. 65–77.

[23] Qizhe Cai, Midhul Vuppalapati, Jaehyun Hwang, Christos Kozyrakis, and Rachit Agarwal. 2022. Towards s tail latency and terabit ethernet: disaggregating the host network stack. In *ACM SIGCOMM*. 767–779.

[24] Marco Carugi and Zhe Lou. 2021. Future Industrial Networks: Requirements, Challenges, Research and Standardization Needs. In *Connecting Physical and Virtual Worlds*. 1–9.

[25] Kai Chen, Ankit Singla, Atul Singh, Kishore Ramachandran, Lei Xu, Yueping Zhang, Xitao Wen, and Yan Chen. 2014. OSA: an optical switching architecture for data center networks with unprecedented flexibility. *IEEE/ACM Trans. Networking* (2014), 498–511.

[26] Ben Cottier, Robi Rahman, Loredana Fattorini, Nestor Maslej, Tamay Besiroglu, and David Owen. 2025. The rising costs of training frontier AI models. http://arxiv.org/abs/2405.21015

[27] Tiago Cruz, Paulo Simões, and Edmundo Monteiro. 2016. Virtualizing Programmable Logic Controllers: Toward a Convergent Approach. *IEEE Embedded Systems Letters* (2016), 69–72.

[28] Robert Czerwinski, Miroslaw Chmiel, and Wojciech Wygrabek. 2013. FPGA implementation of programmable logic controller compliant with EN 61131-3. *IFAC Proceedings Volumes* (2013), 138–143.

[29] Ravirajsinh Dabhi. 2020. casting product image data for quality inspection. https://www.kaggle.com/datasets/ravirajsinh45/real-life-industrial-dataset-of-casting-product

[30] Michael Dalton, David Schultz, Jacob Adriaens, Ahsan Arefin, Anshuman Gupta, Brian Fahs, Dima Rubinstein, Enrique Cauich Zermeno, Erik Rubow, James Alexander Docauer, Jesse Alpert, Jing Ai, Jon Olson, Kevin DeCabooter, Marc de Kruijf, Nan Hua, Nathan Lewis, Nikhil Kasinadhuni, Riccardo Crepaldi, Srinivas Krishnan, Subbaiah Venkata, Yossi Richter, Uday Naik, and Amin Vahdat. 2018. Andromeda: Performance, Isolation, and Velocity at Scale in Cloud Network Virtualization. In *USENIX NSDI*. 373–387.

[31] DPDK. 2025. Pipeline Application. https://doc.dpdk.org/guides/sample_app_ug/pipeline.html

[32] Nandita Dukkipati and Nick McKeown. 2006. Why flow-completion time is the right metric for congestion control. *SIGCOMM Comput. Commun. Rev.* (2006), 59–62.

[33] Manuel Eppler, Steffen Lindner, Lukas Osswald, Thomas Stüber, and Michael Menth. 2025. Impact of Packet Loss and Timing Errors on Scheduled Periodic Traffic with Time-Aware Shaping (TAS) in Time-Sensitive Networking (TSN). *arXiv preprint arXiv:2510.05290* (2025).

[34] Nathan Farrington, George Porter, Sivasankar Radhakrishnan, Hamid Hajabdolali Bazzaz, Vikram Subramanya, Yeshaiahu Fainman, George Papen, and Amin Vahdat. 2010. Helios: a hybrid electrical/optical switch architecture for modular data centers. *ACM SIGCOMM Comput. Commun. Rev.* (2010), 339–350.

[35] Leonhard Faubel, Klaus Schmid, and Holger Eichelberger. 2023. MLOps challenges in industry 4.0. *SN Comput. Sci.* (2023), 828.

[36] Daniel Firestone. 2017. VFP: A Virtual Switch Platform for Host SDN in the Public Cloud. In *USENIX NSDI*. 315–328.

[37] Andreas Frotzscher, Ulf Wetzker, Matthias Bauer, Markus Rentschler, Matthias Beyer, Stefan Elspass, and Henrik Klessig. 2014. Requirements and current solutions of wireless communication in industrial automation. In *Conference on Communications Workshops*. 67–72.

[38] Abd El Hedi Gabsi. 2024. Integrating artificial intelligence in industry 4.0: insights, challenges, and future prospects–a literature review. *Ann. Oper. Res.* (2024). https://doi.org/10.1007/s10479-024-06012-6

[39] Massimiliano Gaffurini, Paolo Bellagente, Alessandro Depari, Alessandra Flammini, Dennis Brandão, Stefano Rinaldi, Emiliano Sisinni, and Paolo Ferrari. 2024. On computing and real-time communication performance of containerized virtual PLCs. In *IEEE Factory Communication Systems*. 1–4.

[40] Massimiliano Gaffurini, Paolo Bellagente, Alessandro Depari, Alessandra Flammini, Emiliano Sisinni, and Paolo Ferrari. 2024. Virtual PLC in Industrial Edge Platform: Performance Evaluation of Supervision and Control Communication. *IEEE Transactions on Instrumentation and Measurement* (2024), 1–10.

[41] Brendan Galloway and Gerhard P. Hancke. 2013. Introduction to Industrial Control Networks. *IEEE Communications Surveys Tutorials* (2013), 860–880.

[42] Sandip Gangakhedkar, Hanwen Cao, Ali Ramadan Ali, Karthikeyan Ganesan, Mohamed Gharba, and Josef Eichinger. 2018. Use Cases, Requirements and Challenges of 5G Communication for Industrial Automation. In *Conference on Communications Workshops*. 1–6.

[43] Omid Givehchi, Jahanzaib Imtiaz, Henning Trsek, and Juergen Jasperneite. 2014. Control-as-a-service from the cloud: A case study for using virtualized PLCs. In *IEEE Workshop on Factory Communication Systems*. 1–4.

[44] René Glebke, Martin Henze, Klaus Wehrle, Philipp Niemietz, Daniel Trauth, Patrick Mattfeld, and Thomas Bergs. 2019. A case for integrated data processing in large-scale cyber-physical systems. In *Conference on System Sciences*. 7252–7261.

[45] Thomas Goldschmidt, Mahesh Kumar Murugaiah, Christian Sonntag, Bastian Schlich, Sebastian Biallas, and Peter Weber. 2015. Cloud-based control: A multi-tenant, horizontally scalable soft-PLC. In *Conference*

on Cloud Computing. 909–916.

[46] Ramesh Govindan, Ina Minei, Mahesh Kallahalla, Bikash Koley, and Amin Vahdat. 2016. Evolve or Die: High-Availability Design Principles Drawn from Googles Network Infrastructure. In *ACM SIGCOMM*. 58–72.

[47] Varun Gowtham, Oliver Keil, Aniket Yeole, Florian Schreiner, Simon Tschöke, and Alexander Willner. 2021. Determining Edge Node Real-Time Capabilities. In *IEEE/ACM Symposium on Distributed Simulation and Real Time Applications*. 1–9.

[48] Chen Griner, Johannes Zerwas, Andreas Blenk, Manya Ghobadi, Stefan Schmid, and Chen Avin. 2021. Cerberus: The Power of Choices in Datacenter Topology Design - A Throughput Perspective. *Proc. ACM Meas. Anal. Comput. Syst.* (2021).

[49] Michael Gundall, Daniel Reti, and Hans D Schotten. 2020. Application of virtualization technologies in novel industrial automation: Catalyst or show-stopper?. In *Conference on Industrial Informatics*. 283–290.

[50] Chuanxiong Guo, Guohan Lu, Dan Li, Haitao Wu, Xuan Zhang, Yunfeng Shi, Chen Tian, Yongguang Zhang, and Songwu Lu. 2009. BCube: a high performance, server-centric network architecture for modular data centers. *ACM SIGCOMM Comput. Commun. Rev.* (2009), 63–74.

[51] Adawiya Hamzah and Abbas Abbas. 2022. Developed an automated design sorting system according to outer edges of object. *Sci. World J.* (2022), 1–10.

[52] Md. Emdadul Haque, Faisal Tariq, Muhammad R. A. Khandaker, Kai-Kit Wong, and Yangyang Zhang. 2023. A Survey of Scheduling in 5G URLLC and Outlook for Emerging 6G Systems. *IEEE Access* (2023), 34372–34396.

[53] Dan Hendrycks and Thomas Dietterich. 2019. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. https://arxiv.org/abs/1903.12261

[54] Jörn-Thorben Hinz, Vamsi Addanki, Csaba Györgyi, Theo Jepsen, and Stefan Schmid. 2023. TCP's Third Eye: Leveraging eBPF for Telemetry-Powered Congestion Control. In *ACM SIGCOMM Workshop on eBPF and Kernel Extensions*. 1–7.

[55] Toke Høiland-Jørgensen, Jesper Dangaard Brouer, Daniel Borkmann, John Fastabend, Tom Herbert, David Ahern, and David Miller. 2018. The eXpress data path: fast programmable packet processing in the operating system kernel. In *ACM CoNEXT*. 54–66.

[56] Diogenes Javier Perez, Josef Waltl, Laurin Prenzel, and Sebastian Steinhorst. 2022. How Real (Time) Are Virtual PLCs?. In *IEEE ETFA*. 1–8.

[57] Bjarne Johansson, Mats Rågberger, Thomas Nolte, and Alessandro V. Papadopoulos. 2022. Kubernetes Orchestration of High Availability Distributed Control Systems. In *IEEE Conference on Industrial Technology*. 1–8.

[58] Thomas Kampa, Amer El-Ankah, and Daniel Grossmann. 2023. High Availability for virtualized Programmable Logic Controllers with Hard Real-Time Requirements on Cloud Infrastructures. In *IEEE Conference on Industrial Informatics*. 1–8.

[59] Talha Ahmed Khan, Syed Mubashir Ali, Kanwar Mansoor Ali, Asif Aziz, Sadique Ahmad, Azeem Anwar, and Sharfuddin Ahmed Khan. 2025. Harnessing Artificial Intelligence for Optimum Performance in Industrial Automation. *Engineering Proceedings* (2025), 105.

[60] Stefan Lachnit, Sebastian Gallenmüller, Dominik Scholz, and Henning Stubbe. 2021. xdpcap: XDP Packet Capture. *Network* (2021).

[61] Sándor Laki, Csaba Györgyi, József Pető, Péter Vörös, and Géza Szabó. 2022. {In-Network} Velocity Control of Industrial Robot Arms. In *USENIX NSDI*. 995–1009.

[62] Ismail Lamaakal, Yassine Maleh, Khalid El Makkaoui, Ibrahim Ouahbi, Paweł Pławiak, Osama Alfarraj, May Almousa, and Ahmed A Abd El-Latif. 2025. Tiny language models for automation and control: Overview, potential applications, and future research directions. *Sensors* (2025), 1318.

[63] Martin Lévesque and David Tipper. 2016. A survey of clock synchronization over packet-switched networks. *IEEE Communications Surveys & Tutorials* (2016).

[64] Yiwei Li, Huaqin Zhao, Hanqi Jiang, Yi Pan, Zhengliang Liu, Zihao Wu, Peng Shu, Jie Tian, Tianze Yang, Shaochen Xu, et al. 2024. Large language models for manufacturing. *arXiv preprint arXiv:2410.21418* (2024).

[65] Zexian Li, Mikko A. Uusitalo, Hamidreza Shariatmadari, and Bikramjit Singh. 2018. 5G URLLC: Design Challenges and System Concepts. In *Symposium on Wireless Communication Systems*. 1–6.

[66] Bingzhe Liu, Colin Scott, Mukarram Tariq, Andrew Ferguson, Phillipa Gill, Richard Alimi, Omid Alipourfard, Deepak Arulkannan, Virginia Jean Beauregard, Patrick Conner, et al. 2024. {CAPA}: An Architecture For Operating Cluster Networks With High Availability. In *USENIX NSDI*. 1995–2010.

[67] Aamir Mahmood, Muhammad Ikram Ashraf, Mikael Gidlund, and Johan Torsner. 2018. Over-the-Air Time Synchronization for URLLC: Requirements, Challenges and Possible Enablers. In *Symposium on Wireless Communication Systems*. 1–6.

[68] Othmane Marfoq, Chuan Xu, Giovanni Neglia, and Richard Vidal. 2020. Throughput-optimal topology design for cross-silo federated learning. https://doi.org/10.48550/arXiv.2010.12229

[69] Jasprabhjit Mehami, Mauludin Nawi, and Ray Y Zhong. 2018. Smart automated guided vehicles for manufacturing in the context of industry 4.0. *Procedia Manuf.* (2018), 1077–1086.

[70] William M. Mellette, Rob McGuinness, Arjun Roy, Alex Forencich, George Papen, Alex C. Snoeren, and George Porter. 2017. RotorNet: a scalable, low-complexity, optical datacenter network. In *Proc. Conf. ACM Special Interest Group Data Commun.* 267–280.

[71] Shoeb Ahmed Memon, Waled Shehata, Steve Rowlinson, and Riza Yosia Sunindijo. 2025. Generative Artificial Intelligence in Architecture, Engineering, Construction, and Operations: A Systematic Review. *Buildings* (2025), 1–6.

[72] Justin Meza, Tianyin Xu, Kaushik Veeraraghavan, and Onur Mutlu. 2018. A Large Scale Study of Data Center Network Reliability. In *IMC*. 393–407.

[73] Nikolaos Mitsakis, Andreas Blenk, and Oliver Hohlfeld. 2024. DEMO: P4PLC: How Dynamic In-Network Control Enables an Interruption-free Migration of vPLCs. In *ACM SIGCOMM Posters and Demos*. 95–97.

[74] Ehsan Mohammadpour and Jean-Yves Le Boudec. 2022. On Packet Reordering in Time-Sensitive Networks. *IEEE/ACM Transactions on Networking* (2022), 1045–1057.

[75] Thomas Müller and Hans Dermot Doran. 2018. Protecting PROFINET cyclic real-time traffic: A performance evaluation and verification platform. In *Workshop on Factory Communication Systems*. 1–4.

[76] Giovanni Neglia, Gianmarco Calbi, Don Towsley, and Gayane Vardoyan. 2019. The role of network topology for distributed machine learning. In *IEEE INFOCOM - IEEE Conf. Comput. Commun.* 2350–2358.

[77] Rolf Neugebauer, Gianni Antichi, José Fernando Zazo, Yury Audzevich, Sergio López-Buedo, and Andrew W. Moore. 2018. Understanding PCIe performance for end host networking. In *ACM SIGCOMM*. 327–341.

[78] Atsuya Osaki, Manuel Poisson, Seiki Makino, Ryusei Shiiba, Kensuke Fukuda, Tadashi Okoshi, and Jin Nakazawa. 2024. Dynamic Fixed-point Values in eBPF: a Case for Fully In-kernel Anomaly Detection. In *Asian Internet Engineering Conference 2024*. 46–54.

[79] John Oyekan, Christopher Turner, Michael Bax, and Erich Graf. 2025. Applying Ontologies and Knowledge Augmented Large Language Models to Industrial Automation: A Decision-Making Guidance for Achieving Human-Robot Collaboration in Industry 5.0. *arXiv preprint*

arXiv:2505.18553 (2025).

[80] Siby Jose Plathottam, Arin Rzonca, Rishi Lakhnori, and Chukwunwike O. Iloeje. 2023. A review of artificial intelligence applications in manufacturing operations. *J. Adv. Manuf. Process.* (2023).

[81] L. Puck, P. Keller, T. Schnell, C. Plasberg, A. Tanev, G. Heppner, A. Roennau, and R. Dillmann. 2021. Performance Evaluation of Real-Time ROS2 Robotic Control in a Time-Synchronized Distributed Network. In *IEEE Conference on Automation Science and Engineering*. 1670–1676.

[82] Rui Queiroz, Tiago Cruz, Jérôme Mendes, Pedro Sousa, and Paulo Simões. 2023. Container-based virtualization for real-time industrial systems—a systematic review. *Comput. Surveys* (2023), 1–38.

[83] Rui Queiroz, Tiago Cruz, and Paulo Simões. 2023. Testing the limits of general-purpose hypervisors for real-time control systems. *Microprocessors and Microsystems* (2023), 104848.

[84] Federico Reghenzani, Giuseppe Massari, and William Fornaciari. 2019. The Real-Time Linux Kernel: A Survey on PREEMPT_RT. *ACM Comput. Surv.* (2019), 1–36.

[85] Marco Reisacher, Ann-Kristin Bergmann, Andreas Blenk, and Stefan Schmid. 2025. Breaking the Vision: Assessing and Mitigating the Impact of Video Artifacts on ML Models in Industrial Use Cases. In *IEEE International Conference on Industrial Informatics*. 1–7.

[86] Marco Reisacher, Reinhard Frank, and Andreas Blenk. 2023. FactoryDC: network and resource planning for emerging applications in future factories. In *Workshop Enhanced Netw. Techn. Technol. Ind. IoT to Cloud Continuum*. 1–7.

[87] Marco Reisacher, Nikolaos Mitsakis, and Andreas Blenk. 2025. DIAMOND: Dynamic Industrial and ML-Optimized Network Design. In *IEEE International Conference on Factory Communication Systems*. 1–8.

[88] Marco Reisacher, Nikolaos Mitsakis, and Andreas Blenk. 2025. INOUT OPTIMA: Trading Off Machine Learning Prediction Quality with Data Quantity for Network Optimization. In *IEEE Network Operations and Management Symposium*. 1–5.

[89] Lorenzo Rosa, Andrea Garbugli, Lorenzo Patera, and Luca Foschini. 2023. Supporting vPLC Networking over TSN with Kubernetes in Industry 4.0. In *Workshop on Enhanced Network Techniques and Technologies for the Industrial IoT to Cloud Continuum*. 15–21.

[90] Jan Rüth, René Glebke, Klaus Wehrle, Vedad Causevic, and Sandra Hirche. 2018. Towards In-Network Industrial Feedback Control. In *ACM Workshop on In-Network Computing*. 14–19.

[91] Christian Scheich, Marius Corici, Hauke Buhr, and Thomas Magedanz. 2023. eXpress Data Path Extensions for High-Capacity 5G User Plane Functions. In *ACM Workshop on EBPF and Kernel Extensions*. 86–88.

[92] Ben Schneider, Björn Richerzhagen, Michael Bahr, and Georg Carle. 2024. Scheduled Trigger Frames: Enabling Worst-case Latency Bounds for Wi-Fi Industrial Use. In *Wireless Communications and Mobile Computing*. 1080–1085.

[93] Dominik Scholz, Daniel Raumer, Paul Emmerich, Alexander Kurtz, Krzysztof Lesiak, and Georg Carle. 2018. Performance Implications of Packet Filtering with Linux eBPF. In *International Teletraffic Congress*. 209–217.

[94] Jonas Schulze, Nils Strassenburg, and Tilmann Rabl. 2025. PQ Bench: Benchmarking Pruning and Quantization Techniques. In *Proceedings of the Workshop on Data Management for End-to-End Machine Learning*.

[95] Eike Schweissguth, Helge Parzyjegla, Peter Danielis, Gero Mühl, Dirk Timmermann, Stefan Mehner, Oliver Hohlfeld, David Hellmanns, and Jonathan Falk. 2023. TSN Scheduler Benchmarking. In *IEEE International Conference on Factory Communication Systems (WFCS)*.

[96] Mohamed Seliem and Dirk Pesch. 2022. Software-Defined Time Sensitive Networks (SD-TSN) for Industrial Automation. In *Conference on Computational Intelligence and Communication Networks*. 1–7.

[97] Natalia Shchepkina, Awadhesh Chandramauli, Suniana Ahuja, P. Prathibha Swaraj, and Rajiv Ranjan. 2024. Deep learning algorithms in industry 5.0: a comprehensive experimental study. *BIO Web of Conferences* (2024), 1067.

[98] Siemens AG 2024. *Simatic S7-1500R/H Redundant System System Manual A5E41814787-AF*. Siemens AG.

[99] Jianchang Su, Yifan Zhang, Linpu Huang, and Wei Zhang. 2024. BOAD: Optimizing Distributed Communication with In-Kernel Broadcast and Aggregation. In *ACM Workshop on EBPF and Kernel Extensions*. 51–57.

[100] Imran Taj and Noor Zaman. 2022. Towards industrial revolution 5.0 and explainable artificial intelligence: challenges and opportunities. *Int. J. Comput. Digit. Syst.* (2022), 285–310.

[101] Timur Tasci, Jan Melcher, and Alexander Verl. 2018. A Container-based Architecture for Real-Time Control Applications. In *IEEE Engineering, Technology and Innovation*. 1–9.

[102] Alexander Verl, Armin Lechler, Stefan Wesner, Andreas Kirstädter, Jan Schlechtendahl, Lutz Schubert, and Sebastian Meier. 2013. An Approach for a Cloud-based Machine Tool Control. In *Conference on Manufacturing Systems*. 682–687.

[103] Jonathan Vestin, Andreas Kassler, and Johan Åkerberg. 2018. FastReact: In-Network Control and Caching for Industrial Control Networks using Programmable Data Planes. In *IEEE ETFA*. 219–226.

[104] Axel Vick, Christian Horn, Martin Rudorfer, and Jörg Krüger. 2015. Control of robots and machine tools with an extended factory cloud. In *Conference on Factory Communication Systems*. 1–4.

[105] Marcos A. M. Vieira, Matheus S. Castanho, Racyus D. G. Pacífico, Elerson R. S. Santos, Eduardo P. M. Câmara Júnior, and Luiz F. M. Vieira. 2020. Fast Packet Processing with eBPF and XDP: Concepts, Code, Challenges, and Applications. *ACM Comput. Surv.* (2020).

[106] Simon Volpert, Sascha Winkelhofer, Jörg Domaschka, and Stefan Wesner. 2025. Towards eBPF Overhead Quantification: An Exemplary Comparison of eBPF and SystemTap. In *ACM/SPEC Conference on Performance Engineering*. 122–129.

[107] Midhul Vuppalapati, Saksham Agarwal, Henry Schuh, Baris Kasikci, Arvind Krishnamurthy, and Rachit Agarwal. 2024. Understanding the Host Network. In *ACM SIGCOMM*. 581–594.

[108] Xin Wang, Deguo Li, Zhihong Wang, Lidong Jiang, Shubo Wen, Daxiang Kang, Engin Arslan, Peng He, Xinyu Qian, Bin Niu, Jianwen Pi, Xiaoning Ding, Ke Lin, and Hao Luo. 2025. Byte vSwitch: A High-Performance Virtual Switch for Cloud Networking. In *EuroSys*. 1417–1432.

[109] Zeyu Wang, Jingao Xu, Xu Wang, Xiangwen Zhuge, Xiaowu He, and Zheng Yang. 2023. Industrial Knee-jerk: In-Network Simultaneous Planning and Control on a TSN Switch. In *ACM MobiSys*. 356–369.

[110] Chengkun Wei, Xing Li, Ye Yang, Xiaochong Jiang, Tianyu Xu, Bowen Yang, Taotao Wu, Chao Xu, Yilong Lv, Haifeng Gao, Zhentao Zhang, Zikang Chen, Zeke Wang, Zihui Zhang, Shunmin Zhu, and Wenzhi Chen. 2023. Achelous: Enabling Programmability, Elasticity, and Reliability in Hyperscale Cloud Networks. In *ACM SIGCOMM*. 769–782.

[111] Yuchen Xia, Nasser Jazdi, Jize Zhang, Chaitanya Shah, and Michael Weyrich. 2024. Control Industrial Automation System with Large Language Models. *arXiv preprint arXiv:2409.18009* (2024).

[112] Yuchen Xia, Jize Zhang, Nasser Jazdi, and Michael Weyrich. 2024. Incorporating Large Language Models into Production Systems for Enhanced Task Automation and Flexibility. *arXiv preprint arXiv:2407.08550* (2024).

[113] Jiawen Xu, Matthias Kovatsch, Denny Mattern, Filippo Mazza, Marko Harasic, Adrian Paschke, and Sergio Lucia. 2022. A review on AI for smart manufacturing: deep learning challenges and solutions. *Appl. Sci.* (2022), 8239.

[114] Siyu Yan, Xiaoliang Wang, Xiaolong Zheng, Yinben Xia, Derui Liu, and Weishan Deng. 2021. ACC: automatic ECN tuning for high-speed datacenter networks. In *ACM SIGCOMM*. 384–397.

[115] Rui Yang and Marios Kogias. 2023. HEELS: A Host-Enabled eBPF-Based Load Balancing Scheme. In *ACM SIGCOMM Workshop on eBPF and Kernel Extensions*. 77–83.

[116] Yuqing Yang and Takuya Azumi. 2020. Exploring Real-Time Executor on ROS 2. In *IEEE Embedded Software and Systems*. 1–8.

[117] Kouros Zanbouri, Md. Noor-A-Rahim, Jobish John, Cormac J. Sreenan, H. Vincent Poor, and Dirk Pesch. 2025. A Comprehensive Survey of Wireless Time-Sensitive Networking (TSN): Architecture, Technologies, Applications, and Open Issues. *IEEE Communications Surveys Tutorials* (2025), 2129–2155.

[118] Johannes Zerwas, Csaba Györgyi, Andreas Blenk, Stefan Schmid, and Chen Avin. 2023. Duo: a high-throughput reconfigurable datacenter network using local routing and control. *Proc. ACM Meas. Anal. Comput. Syst.* (2023), 1–25.

[119] Tianyu Zhang, Gang Wang, Chuanyu Xue, Jiachen Wang, Mark Nixon, and Song Han. 2024. Time-Sensitive Networking (TSN) for Industrial Automation: Current Advances and Future Directions. *Comput. Surveys* (2024), 1–38.