INVITED PAPER

An Introduction to OPC UA TSN for Industrial Communication Systems

This paper is concerned with the adoption of OPC UA TSN for real-time industrial communication. This new protocol architecture could become the solution for establishing a unified communication, from the sensor to the cloud.

By Dietmar Bruckner, *Senior Member IEEE*, Marius-Petru Stănică, Richard Blair, Sebastian Schriegel, Stephan Kehrer, *Member IEEE*, Maik Seewald, *Member IEEE*, and Thilo Sauter, *Fellow IEEE*

ABSTRACT | The industrial communication market is dominated by Ethernet-based fieldbus systems. Although they share similar requirements and market segments, their implementations and ecosystems differ considerably. As a result, end customers and device manufacturers are faced with a multitude of technologies that need to be produced, run, diagnosed, maintained, and kept in stock. Although the availability of products and services is largely satisfactory, dealing with multiple solutions generates high costs and limits IoT capability. This paper introduces Open Platform Communication Unified Architecture Time-Sensitive Networking (OPC UA TSN) as a new technology and presents the current view. This time, the industrial prospects of fulfilling industrial communication requirements while leveraging the cost benefits of standard Ethernet hardware in the midterm are in reach. We anticipate that OPC UA TSN will reveal itself as a game changer in the field of industrial automation, being a candidate for establishing a holistic communication infrastructure from the sensor to the cloud.

KEYWORDS | Converged network; industrial automation; industrial communication; industrial Ethernet; Industrial Internet of Things (IIoT); Open Platform Communication Unified Architecture (OPC UA); real time; Time-Sensitive Networking (TSN).

Manuscript received September 17, 2018; revised November 26, 2018; accepted December 9, 2018. Date of publication January 11, 2019; date of current version May 28, 2019. *(Corresponding author: Dietmar Bruckner.)*

D. Bruckner is with the Department of System Software, B&R Industrial Automation GmbH, 5142 Eggelsberg, Austria (e-mail: dietmar.bruckner@br-automation.com).

M.-P. Stănică is with the Department of Drives Products, ABB Automation Products GmbH, 68526 Ladenburg, Germany (e-mail: marius-petru.stanica@de.abb.com).

R. Blair is with the Department of Industry Business, Schneider Electric, Andover, MA 01810 USA (e-mail: rick.blair@schneider-electric.com).

S. Schriegel is with the Department of Digital Infrastructure, Fraunhofer IOSB-INA, 32657 Lemgo, Germany. (e-mail: Sebastian.schriegel@iosb-ina.fraunhofer.de).

S. Kehrer is with CTO Office, Hirschmann Automation and Control GmbH, 72654 Neckartenzlingen, Germany (e-mail: stephan.kehrer@belden.com).

M. Seewald is with the Department of CTO Innovation Labs, Cisco, 85748 Garching, Germany (e-mail: maseewal@cisco.com).

T. Sauter is with the Institute of Computer Technology, Technische Universität Wien, 1040 Vienna, Austria, and also with the Department of Integrated Sensor Systems, Danube University Krems, 3500 Wiener Neustadt, Austria (e-mail: thilo.sauter@donau-uni.ac.at).

Digital Object Identifier 10.1109/JPROC.2018.2888703

I. MOTIVATION

Today's automation systems traditionally apply a hierarchical architecture known as automation pyramid (Fig. 1). This architecture dates back to the 1970s and was devised as a means to structure the anticipated complexity of automation systems that were evolving at the time. Typical functional layers and associated tools in this pyramid are shop floor, programmable logic controller (PLC), supervisory control and data acquisition (SCADA), manufacturing execution system (MES), and enterprise resource planning (ERP) [1], even though this is by no means the only possible structure and many variants of the automation pyramid have been defined over the years [2].

A focal element in the automation pyramid is communication. Actually, the starting point for the definition of the hierarchical structure was the observation that information exchange between the automation functions is necessary but lacking. Likewise missing were communication systems as a means for data transfer within and between the functional layers. This stimulated the development of a plethora of industrial communication systems specifically

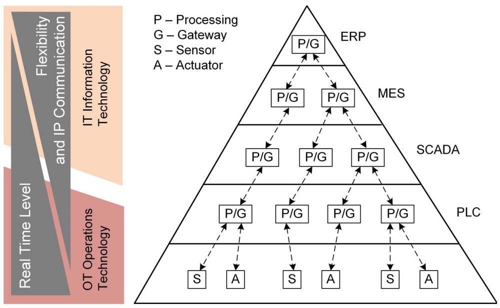

Fig. 1. ***Today's production systems apply the automation pyramid [4].***

tailored to the task of exchanging mostly real-time critical data on the lower layers of the hierarchy. These lower layers are today summarized under the term operational technology (OT). Industrial communication systems like the Ethernet-based POWERLINK,[1] PROFINET,[1] or EtherCAT[1] as well as older so-called fieldbus systems [3] are used to fulfill the requirements regarding deterministic low latency, low jitter, and tight synchronization. The upper layers of the pyramid apply standard information technology (IT), which is, today, exclusively based on Internet technology, i.e., Internet protocol (IP) communication. Interconnection of these two different communication philosophies with significantly diverging requirements has always been a challenge. A common strategy to connect the two worlds is to use gateways [4]–[6].

Pure data transfer is only one aspect of information exchange in automation systems. Another, equally important aspect is data modeling and providing access to data across platforms. Given the large variety of communication systems, a network-independent solution is required as well. Over the years, Open Platform Communication (OPC) evolved as a neutral modeling approach that was supported by an increasing number of device vendors and gradually turned into a commonly accepted information handling standard. OPC defines many functions not addressed by pure industrial communication systems and is, therefore, also a suitable link between the OT and IT layers [5].

The recent Industry 4.0 movement is changing the face of automation systems. First, it defines new requirements or puts old ones in new light [4]: mass customization and efficient automated production of small lot sizes have been a vision for decades. New concepts are data-driven cloud-based services that help to increase the overall efficiency (energy optimization, condition monitoring, and predictive maintenance) [8]. Second, it builds on the Internet of Things (IoT) paradigm that suggests a flat cloud of interconnected devices rather than a sophisticated hierarchy. Applying these new technologies leads to a new automation architecture called the Industrial IoT (IIoT, Fig. 2) [4]. The IIoT uses fewer gateways and a more uniform communication based on IP in all functional layers. The typical automation-related communication requirements such as low latency, high availability, time synchronization, high amount of data and devices as well as seamless reconfiguration (Hot Plug), plug and play, sensor-to-cloud communication [9], and convergence are present and addressed on all functional layers [6].

However, a flatter hierarchy also demands adequate communication systems. It requires the coexistence of OT and IT aspects, in particular, with respect to deterministic communication. In addition, the IoT concept necessitates IP access down to the field devices, which is not possible with legacy fieldbus systems and still partly difficult with today's real-time ethernet (RTE) solutions. To address the new requirements of IIoT, a further evolution of industrial communication seems reasonable. Ethernet Time-Sensitive Networking (TSN) is a recent standardization activity within IEEE 802.1 to include real-time capabilities in the Ethernet standard which ultimately will have the potential to supersede today's industrial communication systems. The second ingredient is the mature OPC Unified Architecture (OPC UA). The combination of both is a candidate that promises to accomplish all requirements of IIoT and Industry 4.0. This paper will introduce the concept of OPC UA TSN in character to [7] and, on top of that, it discusses the current development status as well as open issues.

II. ETHERNET AND REAL-TIME CAPABILITIES

Ethernet (IEEE 802.3) was designed in the 1970s for office communication applications. At the time, it was the first really viable local area network (LAN) technology. The main design goals, in the beginning, were higher bandwidth, easy cabling, and scalability issues. Although using Ethernet in control applications was not the goal when it was designed, there were early thoughts to use this new networking technology also for automation [10]. However, the main hindrance was the random medium access mechanism that precluded deterministic behavior. With the success of Ethernet, its prevalence in office automation and later also in the consumer domain, and particularly technology advancements such as switching

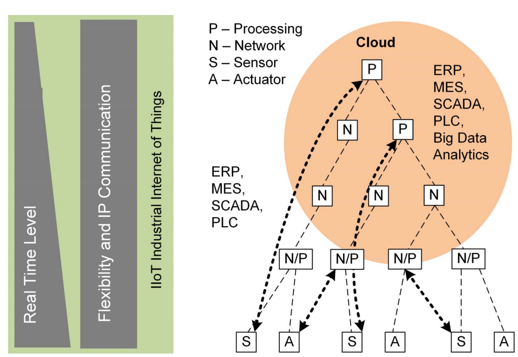

Fig. 2. ***Industry 4.0 architecture: IIoT [4].***

[1]A brand name of a particular real-time industrial Ethernet technology.

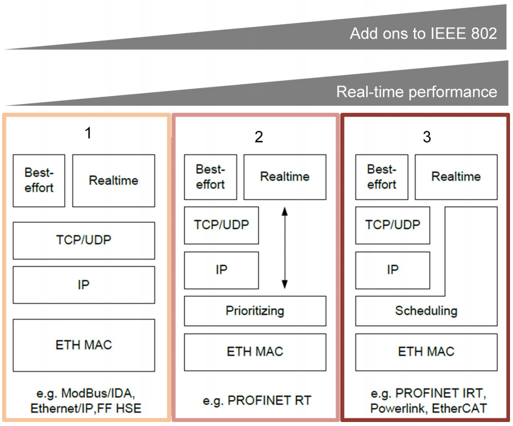

Fig. 3. ***Classification of RTE systems [13].***

technology, Ethernet became interesting again around the year 2000. Various application domains (automotive, energy, industry) and organizations came up with solutions and optimizations to operate Ethernet in real time [11]. These solutions and optimizations have, however, not been included in the related IEEE standards. They are standardized within IEC and only build on the IEEE standards, and they are not interoperable with each other. Their features and characteristics (real-time performance [12], convergence, costs, flexibility, etc.) are different in many details, though they target the same markets and applications.

Ethernet-based industrial communication systems can be classified into three categories ("A"–"C"), which differ in their real-time performance versus their distance to the IEEE standard [13] (Fig. 3). Modbus/TCP[1] is an example for class A that, while offering no special real-time, is based on standard IEEE Ethernet. PROFINET RT[1] is an example for class B that uses virtual LAN (VLAN) prioritization and omits layer 3 and 4 headers for better real-time performance. PROFINET IRT[1] and EtherCAT[1] are examples for class C that use their own medium access control (MAC) layer [a sublayer of Open Systems Interconnection (OSI) layer 2—Data Link Layer] extensions for hard real-time communication: time-controlled media access, frame aggregation/summation frame, and frame fragmentation [13], [14]. Such mechanisms are typically not compatible with standard Ethernet [15]. There are discussions and different opinions on whether these dedicated RTE concepts should still be called "Ethernet", given that the only common denominator is the Ethernet cable (or in OSI terms, the physical layer).

Although the RTE solutions mentioned above were developed mostly by automation system vendors, consumer electronics manufacturers pursued a different path toward deterministic Ethernet. Ethernet audio video bridging (AVB) was the precursor of Ethernet TSN. AVB was developed for time-synchronized professional audio and video streaming over Ethernet networks. Contrary to industrial RTE solutions, this paper was done in the context of the IEEE Ethernet standards. At present, most of its mechanisms have become part of IEEE Std 802.1Q (the standard for "Bridges and Bridged Networks" in IEEE) and the application of AVB is specified as a profile (IEEE Std 802.1BA) of Ethernet TSN. In 2011, first ideas and concepts appeared for applying IEEE AVB technology to industrial automation use cases like control [16]–[18]. The work showed that AVB did not fulfill the requirements and that additional standardization of features was needed. In 2012, the AVB task group (TG) was renamed to TSN TG with the goal of developing IEEE standards that can be used for control applications in industry and automotive areas.

A determinstic wireless Ethernet communication has been a research topic for several years [21]–[24] and leads to first solutions like Isochronous WLAN. The integration of TSN and wireless [19] and the dependencies of 5G and TSN are currently in discussion [20].

III. BACKGROUND

A. OPC UA

Over the years, OPC UA has become omnipresent in industrial automation systems, in particular for the upper layers of the automation pyramid (refer to Fig. 1: ERP, MES, and SCADA). Pervasiveness and increased real-time capabilities together with scalable footprint also make it a very promising technology for the lower layers. The history of OPC began when four automation companies started a task force to develop a standard for data access (DA) based on technologies from Microsoft.

In 1987, as part of Windows 2.0, Microsoft introduced Dynamic Data Exchange (DDE) which allowed programs to exchange data. Three years later, DDE evolved into Object Linking and Embedding (OLE). In 1995, the task force began to work on a DA specification that utilized OLE, which became the basis for the classic OPC (OLE for process control). This was the first useful approach to get automation data into an IT context, independent of particular communication systems. Consequently, OPC found wide support among device vendors and automation solution providers, even though the reliance on proprietary Microsoft concepts was always perceived as a shortcoming and potential threat. Nevertheless, OPC soon evolved into a popular standard for including automation data from the lower OT layers into tools located on the IT layers of the automation pyramid [25], [26]. Over the years, additional specifications were added to enhance the DA specification and the meaning of OPC was redefined to "Open Platform Communication."

The breakthrough toward a truly independent solution came in 2006 with the first specification of OPC UA as a company-independent standard building on modern IT concepts like web services, object-oriented models, and more efficient encodings and protocols. Publisher/subscriber subscriptions were recently added to the current specification release (v1.04) to increase the

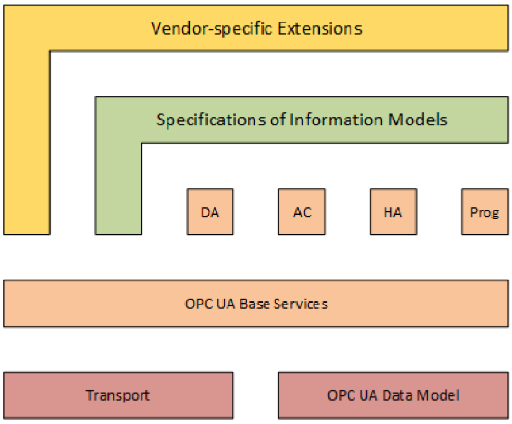

Fig. 4. ***Overview of OPC UA building blocks.***

real-time capability of OPC UA by allowing time-triggered sending of point-to-multipoint messages.

At present, the OPC UA specification consists of 14 parts and a number of companion specifications. Companion specifications usually cover a number of use cases of domain-specific information models and state machines (e.g., for injection molding or robotics). OPC UA features built-in security, and the specified services include Alarms and Conditions (AC), Historizing, Server Aggregation, and Discovery.[2]

Fig. 4 shows the modularity of the specifications that allow for future extensions. Individual parts are dedicated to mapping to transport layers, modeling rules for information models, as well as base services. On top, built-in services such as DA, Historical Access (HA), and AC are available. Programs (Prog) specifies a mechanism to start, manipulate, and monitor the execution of programs. Other organizations can build their models on top of the UA base services or on top of the OPC information model, exposing their specific information via OPC UA. Those organizations specify their extensions in the form of companion specifications. Finally, utilizing each of the introduced layers, every vendor can extend and customize the specifications according to his own business needs, if required.

The technology of OPC UA basically consists of three elements: 1) a metamodel or language that can be used to define specific information models; 2) transport protocol specifications for data exchange between devices; and 3) a server to host the information model and implement the communication protocols, as well as further specified services. The counterpart of the server is seen in the client, which only needs a minimum set of functions compared to a server.

In terms of DA, OPC UA provides read/write access for clients, client/server subscriptions, and publisher/subscriber subscriptions. A subscribed client will receive a message from a server when a predefined condition has been met (for instance, the value of a node has changed). The information model consists of a set of mutually referenced nodes (comparable to a folder structure on a desktop computer), where each node can comprise methods, metadata, and actual data. Finally, OPC UA defines facets for implementations of a server, where each facet enables the utilization of a particular service, for instance, HA.

Hence, the footprint of the implementation can be reduced for resource-constraint devices by limiting the implemented feature set.

[2]https://opcfoundation.org/developer-tools/specifications-unified-architecture

B. Time-Sensitive Networking

The term TSN originates as the name of a TG of the IEEE, more specifically, of the IEEE 802.1 working group. The TSN TG is responsible for developing standards to achieve "guaranteed data transport with bounded latency, low delay variation, and extremely low loss." In order to achieve these goals, a set of standards that provide different tools has already been published (see Fig. 5). However, even more standards projects are currently active or in discussion.

It is important to note that the goals of the TSN TG can be achieved while implementing different sets of tools, among which the traffic shapers are the most relevant [27]. For instance, the goal of achieving low latency can be achieved with (i) implementing dedicated time windows for time-critical traffic (.1Qbv), or with (ii) giving time-critical frames higher priority and the ability to interrupt lower priority frames (.1Qbu and .3br). A third method (iii) would be to restrict the possible bandwidth a sender uses for time-critical traffic, as well as the number of senders, and reserve bandwidth for that time-critical traffic in a network (.1Qav) in return. A fourth method (iv) [a restricted version of method (i)] is to divide the traffic in time-critical and nontime-critical and switch forwarding between the two on a regular schedule (.1Qch). Bounded latency can be achieved with all of these four methods; however, the lower bound varies greatly. Furthermore, methods (i) and (iii) require the network to be precisely time-synchronized, while methods (ii) and (iv) do not.

Similar considerations and options exist for achieving the other goals of TSN. To complicate the situation even more, competing approaches for configuring individual tools in a TSN network exist, which are mutually exclusive and support individual feature sets of tools.

The most relevant functionalities of the TSN specifications are as follows.

1) IEEE Std 802.1AS(-Rev) "Timing and Synchronization for Time-Sensitive Applications," and its upcoming revision are the TSN standards for time synchronization. They are inspired by IEEE 1588 (the Precision Time Protocol) including their basic behavior, however, there are notable differences in terms, functions, and features [28]–[30].
2) IEEE Std 802.1Qbv (already included in 802.1Q-2018) "Enhancements for Scheduled Traffic"

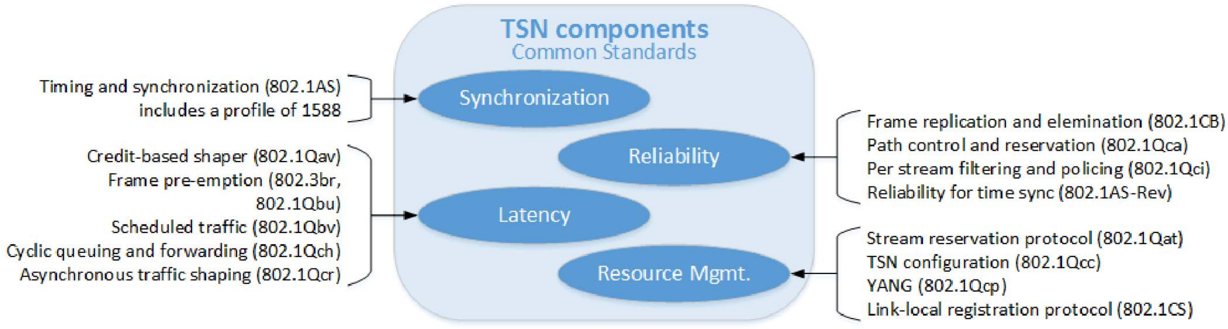

Fig. 5. ***Important TSN standard projects according to the TSN TG. Adopted from http://www.ieee802.org/1/files/public/docs2018/tsn-farkas-intro-0318-v01.pdf.***

introduces time-triggered switch-over between virtual egress queues in a switch (also referred to as the "time-aware shaper"). It requires clock synchronization and provides precise bounded latencies for multiple stream types in a network. In combination with cut-through switching (not an IEEE standard) and time-triggered send capabilities of endpoints, it provides the smallest possible (repeatable) end-to-end latencies for periodic message exchange.

3) IEEE Std 802.1Qav (already included in 802.1Q) "Forwarding and Queuing Enhancements for Time-Sensitive Streams" introduces bandwidth limitations for multiple stream types in a network (also referred to as the "credit-based shaper"). In return, it provides bounded latency per stream type. It has been developed for professional audio and video applications, and the major application can still be seen in such streams.
4) IEEE Std 802.1CB "Frame Replication and Elimination for Reliability" provides means for duplicating streams (in order to send it over multiple paths) and (re)merging duplicates back into a single stream. The mechanism is transparent for other network components.
5) IEEE Std 802.1Qcc "Stream Reservation Protocol Enhancements and Performance Improvements" provides three models for configuring the parameters of the other TSN standards. The two currently examined configuration models are the fully centralized and the distributed configuration models.
6) IEEE Std 802.1Qbu (already included in 802.1Q-2018) (together with IEEE Std 802.3br) "Frame Preemption" (together with "Specification and Management Parameters for Interspersing Express Traffic"), i.e., the switch-related part and the endpoint-related part of frame preemption provide a mechanism to allow higher priority frames to interrupt lower priority frames (that are currently being sent on the same egress port) at every multiple of 64 Bytes (the allowed minimum size of an Ethernet frame).

In order to utilize TSN in the industrial field, a limited set of tools, configuration options, and feature sets need to be defined to create a consensual view for all industrial stakeholders in order to operate multi-vendor installations. The IEC/IEEE 60802 project of an "Industrial TSN profile" currently pursues this goal [33]. Above-mentioned standards will play a major role in TSN's industrial future.

IV. SYSTEM ARCHITECTURE OPC UA TSN

In traditional fieldbuses, a reduced version of the ISO/OSI model has been used to locate functionality. IEC 61784-2 distinguishes only in fieldbus data link layer and fieldbus application layer. The reason for this simplification was the complexity and performance penalty associated with the implementation of a full communication stack. However, for a converged IT and OT network, the range of involved protocols on such a network will be much larger and the full number of layers has to be respected. Fig. 6 shows the minimum protocol suite required to operate such a network.

The upper three items of Fig. 6 belong to layer 7, i.e., the DA method (client–server and publisher–subscriber) as well as the base device information model (which is common for all devices) and subsequent device type-specific information models (controller, input/output, drive, valve, etc.).

A. Industrial Automation Requirements

Today's fieldbuses offer many services to automate setup and operation of networks. A market-based standard is a mandatory set of features that are assessed in independent conformance tests—assuring interoperability among vendors. The feature set usually includes state machines for orchestrated boot-up, device-type profiles for like devices, address management (IP, hostname, other identifiers (IDs), and for TSN traffic also streams), and device identification. Most fieldbuses offer a common device configuration approach for file-based and tool-integration variants (field device tools)/device type manager or field device integration as well as services and state machines for firmware upgrade. For real-time fieldbuses, the principal network

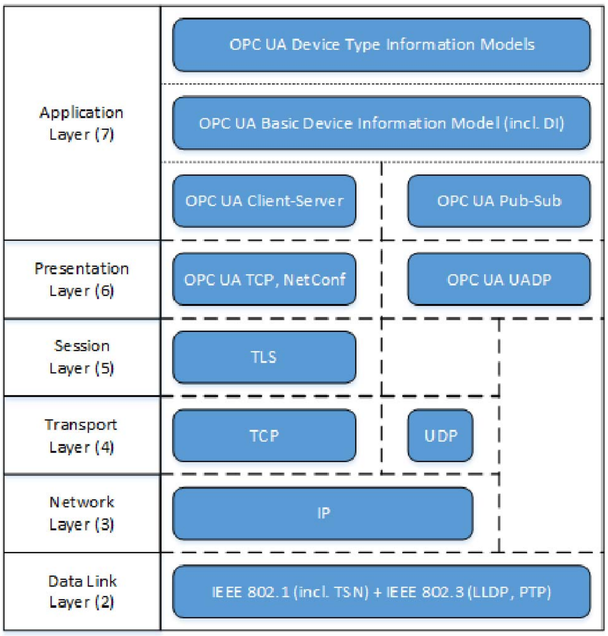

Fig. 6. ***OPC UA TSN in the ISO/OSI layer reference model.***

schedule is given and concrete frame sending times are (pre)calculated in a tool.

In the field of serial machine building, for instance, all information regarding the automation system is stored in a central location (e.g., a CompactFlash card). The devices can be taken fresh from the stock, only some ID (usually a node number) needs to be manually set (or has been preset by the device vendor). The entire life cycle of the devices (including their replacement in case of failure) can happen autonomously.

For OPC UA TSN to be a competitive successor technology to fieldbuses, many such features need to be developed, and, indeed, are under current development. However, it is important to note that the features cannot be mapped in a 1:1 relation, but the functionality may be spread among different entities, protocols, etc., in the new environment. One such important puzzle piece can be seen in the definition of industrial traffic types and their mapping to TSN mechanisms [31].

B. Contemporary Characteristics

OPC UA and TSN represent two very distinct technologies with historically grown stakeholders, standardization procedures, users, and implicit expectations when utilizing their technologies.

For instance, OPC UA expects to have an OPC UA server running on a machine. It implicitly expects a completed boot procedure, IP address allocation, and some available protocols and operating system features. It does not give hints on how fast new configurations are applied (e.g., when does a publisher start to publish valid data?). If the firmware of the machine has to be updated, the procedure takes place out of scope/sight of the server.

From a network point of view, machines running OPC UA are only loosely coupled. The configuration of a publisher and a subscriber are independent of each other. Following boot-up and configuration, the process data exchange starts. In the TSN world, most features are add-ons to good, old, unmanaged Ethernet. Hence, basic forwarding capability of frames will be available all the time (when devices are operational) with potentially degraded performance. With no stream configuration deployed, for example, TSN streams will be treated as multicast frames and distributed in the entire network. To set up the TSN functions, a configuration has to be deployed to each single infrastructure device, utilizing network management (.1Qcc), i.e., a Yet Another Next Generation[3] parameter exchanging protocol like NetConf, which for itself requires IP addresses and a secure layer like Transport Layer Security. Out of the box, TSN devices support topology discovery and all possible network topologies. Another aspect is that TSN devices implement many features in a vendor-specific way (e.g., the existence of certain features can be a vendor-specific specialty).

Altogether, setting up a network of OPC UA TSN devices today to operate properly and then maintain operation requires considerable manual intervention and monitoring.

V. A CALL TO ARMS

OPC UA TSN cannot be seen as a fully developed technology. Both, OPC UA and TSN have active standardization projects running in their respective organizational framework. Even if, for the application in the industrial domain, all the required mechanisms are available in order to operate an OPC UA TSN system there exists, neither for a system provider nor a device vendor, a specification that enforces vendor-independent interoperability for all features of an industrial system. Hence, currently, many vendors and groups of vendors are developing such specifications of varying scope.

This section introduces known open issues in OPC UA and TSN that need specifications for industrial OPC UA TSN.

A. State Machines

Industrial control systems are comprised of a myriad of interconnected devices in order to control a machine or a process. For obvious reasons, all devices required for the industrial application to run should be operational prior to starting production. Application state machines are used within industrial devices to manage this.

Here, a basic state machine for industrial devices will be described. It is important to understand that this is the only representative and individual device types will have

[3] A data modeling language.

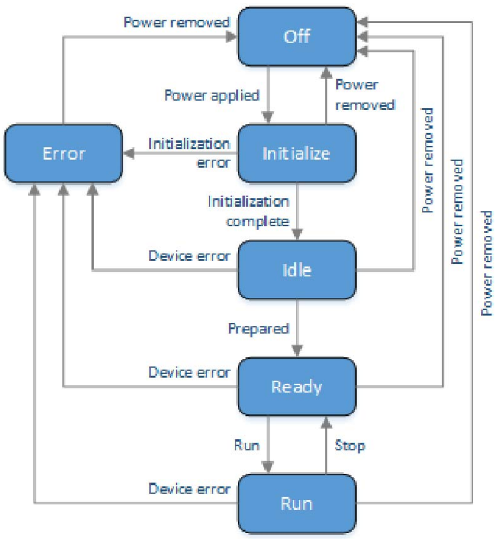

Fig. 7. *Template of an industrial state machine.*

variations to the basic state machine (this is especially true for motion control products). The term application is used to describe the program a device executes to perform its particular function.

The basic state machine consists of six states (Fig. 7).

1) *Power Off:* The device is not operational and is waiting for power or, in the case of standby, a signal to begin operation.
2) *Initialize:* The device is performing initialization tasks, like Power On Self Tests, initializing communication stacks, preparing hardware components, and so on.
3) *Idle:* The device is successfully initialized but the application is not running (e.g., no data exchange, user program not operating, etc.).
4) *Ready:* In this state:
    a) the device has successfully received its applications;
    b) the device has its clock (in case one is present) synched to the master clock (TSN domain master clock);
    c) TSN:
        i) request for TSN related configuration;
        ii) wait for TSN configuration;
        iii) reception of the configuration;
        iv) verification of configuration;
        v) possible needed confirmations are sent out.

    The reason for having this state separate is to insert a clean separation between the basic initialization and the configuration, from various perspectives, starting from application level and ending with the network and TSN levels. Another reason is to wait for an orchestrated start of a number of devices.
5) *Run:* The device's application is operational (e.g., data exchanges functional, user program functioning, etc.).
6) *Error:* The device has detected an error in the device. This is not an application error, but device related (e.g., bad memory chip, unable to initialize a hardware component, etc.).

In today's industrial architectures, it was sufficient to verify that devices were operational before beginning production. TSN now adds the additional requirement that the infrastructure devices (i.e., Ethernet switches) must also be configured prior to data exchanges occurring.

B. Domains

As indicated in Section III-B, one of the current problems of TSN is in its competing, mutually exclusive configuration approaches, namely the fully centralized model [32] and the distributed model.[4] The current standards expect that one configuration model for a TSN network is selected. Since both models require quite distinct sets of protocols, vendors of embedded devices wish to only implement one of them. This situation, if it remains unresolved, can be seen as the major source of incompatibility of TSN networks.

A proposal to overcome the situation is the definition of "TSN domains." Currently, the term domain in the context of TSN is only in use for time synchronization. However, domains for configuration and/or scheduling are required as well as to cope with scalability referring to system size, or in the logical communication paths in a system. The discussion about their specification is ongoing in the IEEE 802.1 TSN TG.

C. Automatic Endpoint Configuration

One of the challenges faced by vendors of OPC UA TSN-based devices is providing commissioning and configuration methods that are robust yet easy to use for end users. Adding to this challenge is the need for increased security measures as more and more industrial systems appear in the crosshairs of attackers.

During commissioning, devices are wired, interconnected, and configured. A combination of manual and automatic methods is used to initially get the system operational. Human observation is used to verify the correctness of network, device, and system operation.

Once commissioned, it is imperative that a system can be powered off and when powered back on, resume operation without any manual intervention. This can be accomplished in several different ways. The first is to statically configure each device on the network during commissioning. This mandates that devices requiring configuration have some type of persistent storage for their configuration data, which may not be practical for highly

[4]There exists actually a third model in the standard, the hybrid model. However, to the knowledge of the authors, there is no implementation available.

constrained devices. In addition, replacement devices must be manually configured prior to insertion into the system. It is also highly unlikely that a mechanism would exist in a statically configured system to validate each device and its associated configuration.

A better approach is to provide a function on the network (Configuration Manager) that contains all pertinent configuration information for the entire system and provides at least the following capabilities.

1) Knowledge of the network topology.
2) Knowledge of each device's type, configuration, and firmware version.
3) Capability of comparing stored topology with actual.
4) Capability of comparing model numbers, configuration data and firmware versions with actual device data.
5) Capability of dispensing configuration and firmware data to devices.
6) Capability of informing applications about the correctness of a system's devices and configurations.

The Configuration Manager can be part of an existing device, (e.g., a controller) or a standalone device.

When a device powers up, passes security checks (e.g., certificates, etc.) and obtains its addressing credentials, it will converse with the Configuration Manager to determine suitability for operation. The Configuration Manager will learn the device's identity, configuration, and firmware level. If all is well, the device is informed as such. Otherwise, the Configuration Manager will update the device accordingly.

TSN adds additional complexity to the configuration, especially for devices that contain embedded switches (bridged endpoints). The configuration data for the switches not only affect the end device but also the communication between other devices. Hence, the dispersal of configuration data may need to be staged, first to switches and then, finally, to devices.

D. Larger Networks—Layer *3* and Cloud Connectivity

OPC UA TSN provides deterministic behavior for a variety of existing use cases in industrial automation. The resulting deployment and application scenarios will evolve over time. Usability and ease of use in configuration and management are important success criterions as well as areas for improvements and innovation. Tight integration between control plane (Central Network Configuration[5] and Central User Configuration[6]) and data plane will enhance the capabilities to configure and manage the network based on intent. This will change the way personnel works and how an automation and control system is managed. From an organizational perspective, the process of OT and IT convergence would be supported in a very pragmatic and intuitive way.

[5]See IEEE 802.1Qcc.
[6]See IEEE 802.1Qcc.

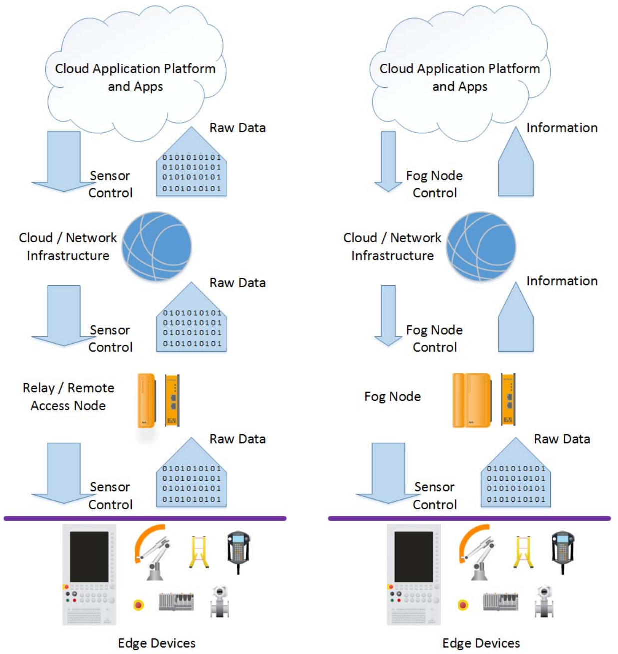

Fig. 8. ***Classic cloud application without fog computing (left) and with fog computing available (right). Control of the process and data aggregation are closer to the actual process.***

However, there will be also completely new use cases and derived application scenarios. OPC UA TSN has the potential to function as an enabler for virtualization of automation and control devices and functions. Bywords such as "Virtual Controller" or "Virtual PLC" are in the (technical) news and a part of roadmaps and future directions. The advantages are manifold. However, this new approach in deployment needs a robust, reliable, and especially deterministic communication technology to connect the virtual environment with physical devices (actors and sensors). OPC UA TSN provides all features to meet the underlying requirement: bounded latency, bounded jitter, and very-low packet loss ratio. Based on this, two deployment scenarios are foreseeable (Fig. 8).

1) Cloud-based deployments: the virtualized control devices are hosted close to industrial automation and control systems, SCADA, and even IT/ERP systems. This would allow the implementation of the so-called large control loops tightly integrated with other systems at this layer.
2) Fog-based deployments: virtualized control devices would reside close to the automation systems but integration with other systems is much easier to achieve. This enables new use cases and fosters distributed intelligence on the manufacturing floor.

With the development in Internet Engineering Task Force Deterministic Networking (Detnet), WAN and metropolitan area network extensions between TSN-based networks (LANs) are achievable. Based on the charter of the Detnet group, such an extension is expected to be possible in a way compatible to TSN. This would especially enable the cloud-based deployments but also connectivity between network segments within an automation solution. The latter use case comprises machine-to-machine communication between PLCs but also connectivity between PLCs and the Supervisory PLC (S-PLC) as described in draft-ietf-detnet-use-cases-05.[7]

Beyond the domain of industrial automation, OPC UA with deterministic behavior has prospects also in other domains such as utility automation or in the process industries leading to new use cases and derived applications.

VI. CONCLUSION AND OUTLOOK

The current move toward the application of IoT concepts in automation systems is stimulating a change in the automation architectures. Specifically, it calls for a rethinking of industrial communication systems and the way of handling automation data. The vision of IoT is a flat communication infrastructure building on Internet technology and ultimately IT standards down to the device level, and it must accommodate the needs of automation applications. This paper has introduced the history of Ethernet, TSN, and OPC UA and their application to industrial use cases. Although the basic building blocks are available, the glue logic in-between still requires standardization efforts. We have sketched some current hot topics of standardization in more detail; however, others have been just touched upon during the introduction of the bigger picture.

Currently, a number of industry-driven standardization efforts (most notably the IEC/IEEE 60802 industrial TSN profile and the "Field-Level Communication" working group in the OPC Foundation) are ongoing to fill the blank spots of OPC UA TSN. For the first time, all of the top five automation players (in Europe) and many more support these initiatives. However, there is still a lot to be done, and the outcome will be rewarding. If forces are joined wisely, we can ultimately achieve what has been a vision for almost half a century: a truly uniform and single standardized approach to industrial communication. Throughout the evolution of fieldbus systems and RTE solutions, many company politics and technical difficulties got in the way to reach this ambitious goal. If OPC UA and TSN technologies are properly evolving, there is another, maybe unique, chance to achieve it. The signs have never been more promising than they are today. ■

REFERENCES

[1] 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.

[2] T. Sauter, S. Soucek, W. Kastner, and D. Dietrich, "The evolution of factory and building automation," *IEEE Ind. Electron. Mag.*, vol. 5, no. 3, pp. 35–48, Sep. 2011.

[3] G. Zucker, D. Bruckner, and D. Dietrich, "Trends and challenges for industrial communication systems," in *The Industrial Electronics Handbook, Part IV: Industrial Communication Systems*, B. M. Wilamowski and J. D. Irwin, Eds., 2nd ed. Boca Raton, FL, USA: CRC Press, 2011.

[4] S. Schriegel, T. Kobzan, and J. Jasperneite, "Investigation on a distributed SDN control plane architecture for heterogeneous time sensitive networks," in *Proc. 14th IEEE Int. Workshop Factory Commun. Syst. (WFCS)*, Imperia, Italy, Jun. 2018, pp. 1–10.

[5] T. Sauter and M. Lobashov, "How to access factory floor information using Internet technologies and gateways," *IEEE Trans. Ind. Informat.*, vol. 7, no. 4, pp. 699–712, Nov. 2011.

[6] S. Heymann *et al.*, "Cloud-based plug and work architecture of the IIC testbed smart factory Web," in *Proc. IEEE 23rd Int. Conf. Emerg. Technol. Factory Automat. (ETFA)*, Turin, Italy, Sep. 2018, pp. 187–194.

[7] D. Bruckner *et al.*, "OPC UA TSN—A new solution for industrial communication," White Paper, Jan. 2018.

[8] K. Al-Gumaei *et al.*, "A survey of Internet of Things and big data integrated solutions for industrie 4.0," in *Proc. IEEE Int. Conf. Emerg. Technol. Factory Automat. (ETFA)*, Turin, Italy, Sep. 2018, pp. 1417–1424.

[9] T. Kobzan, S. Schriegel, S. Althoff, A. Boschmann, J. Otto, and J. Jasperneite, "Secure and time-sensitive communication for remote process control and monitoring," in *Proc. IEEE Int. Conf. Emerg. Technol. Factory Automat. (ETFA)*, Turin, Italy, Sep. 2018, pp. 1105–1108.

[10] J. D. Decotignie, "Ethernet-based real-time and industrial communications," *Proc. IEEE*, vol. 93, no. 6, pp. 1102–1117, Jun. 2005.

[11] M. Felser, "Real-time Ethernet—Industry prospective," *Proc. IEEE*, vol. 93, no. 6, pp. 1118–1129, Jun. 2005.

[12] J. Robert, J.-P. Georges, E. Rondeau, and T. Divoux, "Minimum cycle time analysis of Ethernet-based real-time protocols," *Int. J. Comput., Commun. Control*, vol. 7, no. 4, pp. 743–757, 2012.

[13] J. Jasperneite, M. Schumacher, and K. Weber, "Limits of increasing the performance of industrial Ethernet protocols," in *Proc. IEEE Conf. Emerg. Technol. Factory Autom. (EFTA)*, Sep. 2007, pp. 17–24.

[14] D. Dietrich, D. Bruckner, G. Zucker, and P. Palensky, "Communication and computation in buildings: A short introduction and overview," *IEEE Trans. Ind. Electron.*, vol. 57, no. 11, pp. 3577–3584, Nov. 2010, doi: 10.1109/TIE.2010.2046570.

[15] R. Schlesinger, A. Springer, and T. Sauter, "Concept for the coexistence of standard and real-time Ethernet," in *Proc. 14th IEEE Int. Workshop Factory Commun. Syst. (WFCS)*, Imperia, Italy, Jun. 2018, pp. 1–10.

[16] J. Imtiaz, J. Jasperneite, and S. Schriegel, "A proposal to integrate process data communication to IEEE 802.1 audio video bridging (AVB)," in *Proc. 16th IEEE Int. Conf. Emerg. Technol. Factory Automat. (ETFA)*, Toulouse, France, Sep. 2011, pp. 1–8.

[17] S. Schriegel and J. Jasperneite, "Taktsynchrone Applikationen mit PROFINET IO und Ethernet AVB," in *Proc. VDI Kongress Autom.*, Baden-Baden, Germany, Jun. 2011.

[18] N. Finn, "Time-sensitive and deterministic networking," Huawei Technol., Shenzhen, China, White Paper, Jul. 2017.

[19] F. Bonanati, "TSN over wireless—Dream or soon reality?" in *Proc. TSN/A Conf.*, Stuttgart, Germany, 2018.

[20] A. Nasrallah *et al.* (2018). "Ultra-low latency (ULL) networks: The IEEE TSN and IETF DetNet standards and related 5G ULL research." [Online]. Available: https://arxiv.org/pdf/1803.07673.pdf

[21] H. Trsek, T. Tack, O. Givehchi, J. Jasperneite, and E. Nett, "Towards an isochronous wireless communication system for industrial automation," in *Proc. 18th IEEE Int. Conf. Emerg. Technol. Factory Automat. (ETFA)*, Cagliari, Italy, Sep. 2013, pp. 1–4.

[22] F. Ring, A. Mahmood, L. Wisniewski, and H. Trsek, "Wireless real-time communication for industrial automation systems in simulation and reality," in *Proc. 13th Mechatronics Forum Int. Conf.*, Linz, Austria, Sep. 2012.

[23] A. Mahmood, G. Gaderer, H. Trsek, S. Schwalowsky, and N. Kerö, "Towards high accuracy in IEEE 802.11 based clock synchronization using PTP," in *Proc. Int. IEEE Symp. Precis. Clock Synchronization Meas., Control Commun. (ISPCS)*, Munich, Germany, Sep. 2011, pp. 13–18.

[24] H. Trsek and J. Jasperneite, "An isochronous medium access for real-time wireless communications in industrial automation systems—A use case for wireless clock synchronization," in *Proc. Int. IEEE Symp. Precis. Clock Synchronization Meas., Control Commun. (ISPCS)*, Munich, Germany, Sep. 2011, pp. 81–86.

[25] P. M. P. A. Blanco, M. A. Poli, and M. R. P. Barretto, "OPC and CORBA in manufacturing execution systems: A review," in *Proc. IEEE Conf. Emerg. Technol. Factory Automat. (EFTA)*, Lisbon, Portugal, vol. 2, Sep. 2003, pp. 50–57.

[26] D. W. Holley, "Understanding and using OPC for maintenance and reliability applications," *IEE Comput. Control Eng.*, vol. 15, no. 1, pp. 28–31, 2004.

[27] S. Schriegel *et al.*, "Ethernet TSN Nano Profil—Migrationshelfer vom industriellen Brownfield zum Ethernet TSN-basierten IIoT," in *Proc. Jahreskolloquium Kommunikation Autom. (KommA)*, Lemgo, Germany, Nov. 2018.

[28] R. Hummen, S. Kehrer, and O. Kleineberg, "White Paper: TSN—Time sensitive networking," Belden, St. Louis, MI, USA, Tech. Rep., Feb. 2017.

[29] S. Schriegel and J. Jasperneite, "Investigation of industrial environmental influences on clock sources and their effect on the synchronization accuracy of IEEE 1588," in *Proc. ISPCS*, Vienna, Austria, Oct. 2007, pp. 50–55.

[7] https://tools.ietf.org/html/draft-ietf-detnet-use-cases-05/

[30] S. Schriegel, A. Biendarra, O. Ronen, H. Flatt, G. Leßmann, and J. Jasperneite, "Automatic determination of synchronization path quality using PTP bridges with integrated inaccuracy estimation for system configuration and monitoring," in *Proc. ISPCS*, Beijing, China, Oct. 2015, pp. 52–57.
[31] P. Didier *et al.*, "Time sensitive networks for flexible manufacturing testbed—Description of converged traffic types, version 2," Industrial Internet Consortium, White Paper, 2018.
[32] E. Gardiner, "Theory of operation for TSN-enabled systems," AVnu Alliance, Tech. Rep., Feb. 2017.
[33] *IEC IEEE 60802 TSN Profile for Industrial Automation*. [Online]. Available: https://1.ieee802.org/tsn/iec-ieee-60802-tsn-profile-for-industrial-automation/
[34] S. Zuponcic, R. Klecka, M. Hantel, and P. Didier, "TSN influences on ODVA technologies: IEEE-802.1, AVnu, IETF," ODVA, Tech. Rep., Feb. 2017.

ABOUT THE AUTHORS

Dietmar Bruckner (Senior Member, IEEE) received the Dipl.-Ing. degree in electrical engineering and information technology and the Ph.D. degree in technical sciences from the Technische Universität Wien, Vienna, Austria, in 2004 and 2007, respectively.

From 2004 to 2013, he was with the Technische Universität Wien, where he managed the Cognitive Automation Research Group, Institute of Computer Technology. Since 2013, he has been with B&R Industrial Automation GmbH, where he manages the Group Connectivity, Department System Software. He was involved in hard real-time, industrial Ethernet-based communication systems. He has authored more than 70 scientific publications. He edited one IEEE standard. He holds five international patents. He is currently involved in organizing international conferences, editing high-ranked international journals, and drafting international standards.

Dr. Bruckner is an IEEE IES AdCom member and a member of the Austrian Electrotechnical Association. He is a Co-Editor-in-Chief of the Industrial Electronics Technology Transfer News. He serves as an Associate Editor for the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS and the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS.

Richard Blair was born in Milwaukee, WI, USA. He received the B.S. degree in electrical engineering and computer science and the M.S. degree in electrical engineering, with a focus on control theory, from the University of Wisconsin, Milwaukee, WI, USA, in 1977 and 1984, respectively.

From 1977 to 1987, he was with Unico, Inc., Franksville, WI, USA, where he held numerous positions and was involved in motion control. He was with Square D/Schneider Electric, Andover, MA, USA, where he held various positions. He was a Principal System Architect with Schneider Electric, Andover, MA, USA, where he was involved in industrial communications. He has authored or co-authored numerous papers in various industrial organizations such as ODVA, IIC, and FDT Group. He holds multiple patents. His current research interest includes real-time communications and user interface design, with a focus on accessibility.

Marius-Petru Stănică received the B.Sc. and M.Sc. degrees in automation and computer engineering from the Politehnica University of Bucharest, Bucharest, Romania, in 2000 and 2001, respectively, and the Ph.D. degree in model-checking control systems from the University of Rennes I. and Supélec (École Supérieure d'Eléctricité), Cesson-Sévigné, France, in 2005.

He has authored several publications in network-based control and in network management systems. He was with the Politehnica University of Bucharest, where he was involved in teaching. In 2005, he was with ABB, covering several European countries: Norway, Romania, Italy, and Germany. He was involved in research, system architecture and product management responsibilities on industrial automation Ethernet-based networks for the oil and gas, power generation, and electrical drives industries and markets.

Dr. Stănică is a member of the German IEC National Committee (DKE) and a participant within the IEEE/IEC Joint Project 60802 TSN Industrial Automation Profile, as well as of the OPC Foundation (TSN Subgroup and Field-Level Communication working groups).

Sebastian Schriegel received the Dipl.-Ing. (FH) degree in electrical engineering and the M.Sc. degree in mechatronic engineering from the Ostwestfalen-Lippe University of Applied Sciences, Lemgo, Germany, in 2005 and 2010, respectively.

From 2005 to 2009, he was with the Institute Industrial IT, Ostwestfalen-Lippe University of Applied Sciences, where he was involved in the development of an application-specified integrated circuit for RTE communication. Since 2009, he has been with the Fraunhofer IOSB-INA, Lemgo, where he is currently the Manager of the Research Group Communications Systems. He has authored more than 30 scientific publications. He holds two international patents. His current research interests include real-time communication, high-precision time synchronization, automatic management of time-sensitive networks and migration strategies, and interoperability in industrial communication.

Mr. Schriegel was a recipient of the Best Paper Award from the IEEE ISPCS 2007 in Vienna and the Best Paper Award from the Gesellschaft für Informatik e.V. Conference "Echtzeit" (Real-Time) in 2014. He served as a General Co-Chair for the IEEE International Symposium of Precision Clock Synchronization (ISPCS) in 2013 and a Program Co-Chair in 2014. He continues to be active in the ISPCS program committee.

Stephan Kehrer (Member, IEEE) received the Dipl.-Inf. degree in information technology from the Eberhard Karls Universität Tübingen, Tübingen, Germany, in 2007.

He was involved in industrial Ethernet networks with a focus on deterministic communication and redundancy mechanisms. Since 2007, he has been with Hirschmann Automation and Control GmbH, Neckartenzlingen, Germany, a company belonging to Belden, Inc., where he is currently a Senior Architect with CTO Office and is involved in research, evaluation, and development of new technologies for the industrial communication area. He holds several pending patents. Since 2013, he has been involved in drafting international standards.

Mr. Kehrer is an Active Member of the IEEE 802.1 Working Group as well as of working groups within the IEC. In IEEE 802.1, he is currently focused on the standardization of Time-Sensitive Networking, where he is the editor of an ongoing standards project. He is currently a public speaker.

Maik Seewald (Member, IEEE) received the Dipl.-Ing. degree in telecommunication engineering from the University of Applied Science, Dresden, Germany, in 1989.

He was with Advanced Micro Devices (AMD), Sunnyvale, Germany, and AMD, Dresden, Germany; Audi, Ingolstadt, Germany; and Siemens, Munich, Germany, and Siemens, Nuremberg, Germany. He is currently a Senior Technical Leader with the Cisco's Enterprise Networking Group, where he focuses on the development of the Internet-of-Things (IoT) architecture, security, and standards for Cisco's CTO Team. He is a frequent public speaker and a technical writer with a focus on cybersecurity and industrial IoT. He has authored *Inventions*. His current research interests include deterministic networking, distributed intelligence, and security and privacy for IoT systems.

Mr. Seewald is an Active Member of IEC working groups and of ETSI TC CYBER in the standardization areas of networking technologies and security.

Thilo Sauter (Fellow, IEEE) received the Dipl.-Ing. and Ph.D. degrees in electrical engineering from the Technische Universität Wien (VUT), Vienna, Austria, in 1992 and 1999, respectively.

From 1992 to 1996, he was a Research Assistant with the Institute of General Electrical Engineering, VUT, where he was involved in the research on programmable logic and analog application-specified integrated circuit design. Subsequently, he joined the Institute of Computer Technology, VUT, where he led the Factory Communications Group. From 2004 to 2013, he was the Founding Director of the Institute for Integrated Sensor Systems, Austrian Academy of Sciences, Wiener Neustadt, Austria. Since 2013, he has been with the Department for Integrated Sensor Systems, Danube University Krems, Wiener Neustadt. In 2014, he became a Tenured Associate Professor of automation technology with VUT. He has been involved in the standardization of industrial communication systems for more than 20 years. His current research interests include smart sensors and automation networks with a focus on real-time, security, interconnection, and integration issues. He has authored more than 300 scientific publications.

Dr. Sauter was elected IEEE Fellow for his contributions to synchronization and security in automation networks. He is the Vice President of the Austrian Association for Instrumentation, Automation, and Robotics, an AdCom Member of the IEEE Industrial Electronics Society, and a Treasurer of the IEEE Austria Section. He held leading positions in renowned IEEE conferences. He is currently the Editor-in-Chief of *IEEE Industrial Electronics Magazine* and an Associate Editor of the IEEE Transactions on Industrial Informatics and the IEEE sensors Journal.