Real-Time Capable Internet Technologies for Wired Communication in the Industrial IoT—a Survey

Peter Danielis, Henning Puttnies, Eike Schweissguth and Dirk Timmermann
University of Rostock, Faculty of Computer Science and Electrical Engineering
Institute of Applied Microelectronics and Computer Engineering, 18051 Rostock, Germany
Email: {peter.danielis,henning.puttnies,eike.schweissguth,dirk.timmermann}@uni-rostock.de

Abstract—**This survey addresses the question if existing real-time capable Internet technologies for wired communication can meet the requirements of future industrial IoT scenarios with a rising number of heterogeneous devices to be connected and increasing amounts of data to be exchanged in real-time. The development towards an industrial IoT is further referred to as Industry 4.0 in Germany and Industrial Internet in the USA, respectively. We first investigate selected widespread technologies at all layers of the ISO/OSI model with respect to their real-time capability, scalability and dynamic reconfiguration, standard compliance and platform complexity as well their capability to integrate non-real-time devices. On the one hand, we note that TSN technology at physical and link layer is standardized but exhibits very high platform complexity for the switches and is thus costly. Subsequently, it is hence discussed if purely software-based approaches can enable RT communication over Ethernet. Moreover, even though TSN-enabled network components can enforce real-time behavior, scheduling and routing algorithms for computing the respective network configuration are not part of the TSN standards. Those algorithms could be executed on a central SDN controller to achieve high performance and real-time capability, however, the scalability of such a centralized approach is limited by the fact that corresponding algorithms have exponential computational complexity. Hence, one of the future research directions outlined proposes to trade off distributed against centralized scheduling and routing approaches with regard to scalability and dynamic reconfiguration, real-time capability, and platform complexity. We conclude there is a need for the advancement of existing and for the development of new, possibly hybrid, real-time capable approaches that combine the advantages of centralized and distributed solution in order to meet all requirements.**

I. INTRODUCTION

Our environment is pervaded by a growing number of networked devices. This development is often referred to as Internet of Things (IoT) if the devices are connected by using Internet protocols [1]. This development continues in industrial automation as industrial IoT (IIoT) [2]. In Germany, the specific term “Industry 4.0” has been coined for this trend [3]. Moreover, the US-American company General Electric initiated a comprehensive research initiative called “Industrial Internet” and predicts an increasing number of device in industrial environments as well [2]. In industrial environments, *real-time (RT) capability* must be guaranteed for many applications. This can be done by calculating routes and schedules to enable TDMA for the communication between devices, i.e., their flows, and hence by introducing time slots, during which the devices may communicate over the computed route. Established solutions realize the RT capability through costly hardware adjustments. However, an important future challenge is to ensure interoperability between devices with respect to different device classes, e.g., embedded microcontrollers or office PCs, operating systems, and processor architectures [4]. Consequently, a *standardized approach* [5] that requires the lowest possible complexity for the platform and ensures the RT capability, possibly at the application layer, might be suitable for this purpose. In addition, many established solutions rely on a central component such as an software-defined networking (SDN) controller to coordinate the network nodes to ensure RT capability by controlling their routing and by scheduling their send times. Such centralized approaches do not scale well because solving the routing and scheduling problem exhibits exponential complexity [6], [7]. However, since a large number of devices are to be networked with each other in the IIoT, *scalability* with regard to the number of devices plays a decisive role for future networking technologies. In particular, the *reconfiguration* must be possible *dynamically at runtime* in order to react flexibly to changing system parameters regarding participants that enter or leave the network or changes in communication patterns and requests. Hence, distributed designs may be beneficial in this regard or even hybrid approaches combining the advantages of centralized and distributed solutions may turn out to be advantageous. The IEEE Time-Sensitive Networking (TSN) standards, which are still under development, also focus on realizing RT capability at the lower layers at the expense of introcuding high platform complexity for network components [8], [9], [4].

In this survey paper, we classify various widespread Internet technologies and protocols at all layers of the ISO/OSI model, bottom up starting with with fieldbus- and Ethernet-based communication, for their suitability for use in the IIoT regarding the requirements stated below. Actually, Ethernet-based technologies tailored to the requirements in industrial environments has overtaken fieldbuses in terms of the number of newly installed nodes in factory automation [10]. We focus on wired networks at the physical and data link layer, as wireless networks for RT communication at these layers constitute a comprehensive research field of their own and wireless technologies such as 5G may be intended to complement but not replace wired technologies [4]. However, our explanations for the higher layers are largely independent of the underlying layers, although reliability aspects in wireless networks play a greater role than for wired networks and should therefore be considered at higher layers. Please observe that an overview that completely covers all available industry and research technologies is not feasible in this paper. Hence, our focus is on widely adopted technologies and promising research works.

We show the benefits and shortcomings of the technologies under investigation and derive future research directions to overcome outlined shortcomings. IIoT scenarios namely raise new requirements, some of which are significantly different from those previously imposed on the IoT or the Internet:

- Requirement 1 (R1-RT), RT capability: A distinction has to be made between hard and soft RT requirements for the RT requirements of a system [11]. While applications with soft RT requirements such as audio/video streaming for maintenance tasks or non-critical monitoring of production processes do not necessarily have to guarantee the compliance with deadlines, applications with hard RT requirements for factory or power plant control must meet deadlines under all circumstances [12].
- Requirement 2 (R2-NetScal), scalability regarding number of devices: In the future, as part of the IIoT vision, several hundred or even thousands of devices must be connected in a network [2].
- Requirement 3 (R3-DatScal), scalability regarding data volume: Furthermore, the scalability of the supported data volume should be ensured: While current Internet traffic is significantly influenced by audio and video streaming, so far in the industrial environment other traffic patterns are typical. Many sensors typically send small process data packets to controllers, which in turn control a plurality of actuators. Perspectively, however, both the transmission of small (a few kilobytes) and larger amounts of data (data streams with a few megabytes) with and without RT requirements should be supported.
- Requirement 4 (R4-DR), dynamic reconfiguration: The network can also change at runtime and thus needs to be reconfigurable [2]. Communication in the IIoT as well as algorithms for reconfiguring new devices entering the network must therefore be highly scalable in terms of the number of devices supported.
- Requirement 5 (R5-SC), standard compliance: Extensive use of standard protocols and hardware: According to [2], the cost-efficiency of the IIoT is of particular importance, since many devices are to be networked together and the hardware used should therefore be cost-efficient in order to minimize the capital expenditure. Ideally, special and proprietary hardware should be avoided for the transmission of both non-RT and RT data. In addition, approaches with low platform requirements concerning their operating systems and processor architectures should be applied so that new platforms can be easily integrated into existing networks. Furthermore, it is stated in [1] that a software stack that is compatible with the Internet must be used in IIoT. Preferably, the standardized Internet stack should hence be used.
- Requirement 6 (R6-NRT), integration of non-RT devices into RT networks: The integration of non-RT devices into RT networks should be possible without disturbing the RT operation [12].

The rest of this paper is organized as follows. Section II gives an overview of technologies at physical and link layer. Section III presents technologies at network and transport layer. Section IV describes technologies at application layer. Future directions are given in Section V, and Section VI concludes the paper.

II. TECHNOLOGIES AT PHYSICAL AND LINK LAYER

Existing industrial networking technologies achieve RT capability through adjustments at the lower two ISO/OSI layers. Fieldbus systems can meet R1-RT and R6-NRT, but not R2-NetScal and R3-DatScal as well as R5-SC. Therefore, as mentioned above Ethernet has overtaken fieldbuses in terms of the number of newly installed nodes in factory automation [10].

Even standard Ethernet offers significant advantages over fieldbuses, allowing full vertical and horizontal integration from the field level to the corporate level [12]. This is crucial for the realization of the vision of the IIoT. Although Ethernet meets R2-NetScal, R3-DatScal, R4-DR, and R5-SC, it does not meet R1-RT and R6-NRT without additional mechanisms. If, for instance, network nodes send too much data to the same destination so that the available bandwidth on the link to the destination is exceeded they may experience buffer overflow and packet loss in the switches.

Therefore, a variety of industry-established RT Ethernet variants has been developed to meet R1-RT [13]. In the automation environment, hard RT requirements are placed on process control. In most cases, the communication pattern between devices repeats cyclically and offline schedules are created to allocate time slots to devices for their communication. Established RT capable Ethernet systems in this area are Modbus-TCP [14], Ethernet Powerlink [15], EtherCAT [16], TCnet [17], TTEthernet [18] , CC-Link IE Field [19], Profinet [20], Ethernet/IP [21], and SERCOS III [22]. Please note that Modbus-TCP is solely able to achieve soft RT, does not imply changes at lower layers, and is therefore only mentioned for completeness. Other solutions such as Ethernet/IP and Ethernet Powerlink provide software-only versions but require lower-layer changes to achieve the best possible RT properties. The systems meet R1-RT and often R2-NetScal, but not R3-DatScal as far as RT communication is concerned. Non-RT capable devices are normally supported and therefore also R6-NRT is fulfilled. The hardware requirements are usually proprietary, so that R5-SC is not met. Most systems (e.g., Ethernet Powerlink, EtherCAT, CC-Link IE Field, Profinet, and SERCOS III) are master/slave approaches. For instance, EtherCAT uses a master to synchronize all devices and control data exchange. In addition, solutions such as EtherCAT, TCnet, TTEthernet, and Profinet require the use of special, expensive and proprietary hardware for the best possible RT properties.

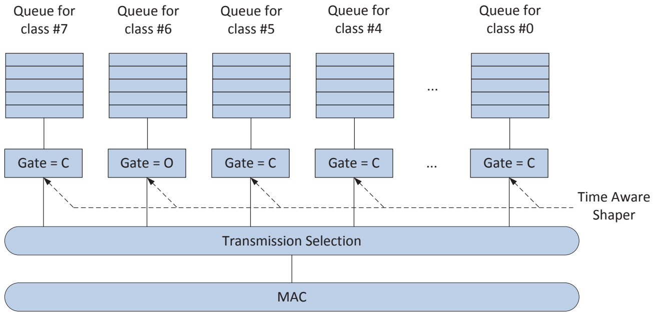

Fig. 1. Stream classes defined in IEEE 802.1Q [24].

In summary, there is no solution among industry-established RT Ethernet technologies that meets all requirements [23]. Thus, the conception and methodical investigation of a system architecture for the fulfillment of R3-DatScal and R4-DR regarding the reconfiguration at runtime without having to completely recompute schedules offline and R5-SC remain an open research question.

Furthermore, there are some current research approaches, which compensate for the weaknesses of established RT Ethernet systems. In [25], RTnet is presented as a pure software solution for RT communication but it is not only implemented at the application but also at lower layers. RTnet is based on the real-time application interface (RTAI) extension of Linux. However, some adjustments have been made in the driver as well as in a TDMA mechanism has to be integrated into the link layer, which violates R5-SC. Moreover, the IP protocol has been adapted with regard to routing and defragmentation. In addition, a stack manager is inserted to distribute the packages between the driver and the UDP/IP stack. Schmidt et al. propose an approach called Distributed Real-Time Protocols for Industrial Control Systems (DRTP) based on two additional proprietary layers above Ethernet to control media access [26]. However, no scalability analyses are performed (R2-NetScal and R3-DatScal). In addition, TCP, UDP, and IP are not supported, which violates R5-SC. Hu et al. introduce an RT Ethernet protocol called Design and application of a RT industrial Ethernet protocol (DARIEP) using a master/slave approach in [27]. The master was developed under Linux and uses RTAI. In addition, the slaves are based on special hardware in the form of FPGA and ARM chips. The work of Schlesinger and Springer [28] introduces the Very High Performance Automation Bus System (VABS), which is suitable both for the transmission of time-critical and non-time-critical data. However, VABS requires the introduction of a proprietary Ethernet MAC layer and a special process data protocol. In [29], the authors describe an FPGA-based open source RT Ethernet framework called Atacama. A specially developed application-specific instruction-set processor coordinates the time-critical data exchange between RT capable nodes. This research thus violates R5-SC and/or does not investigate scalability, R2-NetScal and R3-DatScal. The Flexible Time Triggered (FTT) model introduced in [30] is based on a master/slave architecture that enables RT communication while allowing dynamic scheduling. The master determines a send schedule for each cycle with the aid of any scheduling method according to the specifications of the slaves and informs its slaves at the beginning of a cycle. The FTT Switched Ethernet (SE) implementation allows the transmission of periodic and aperiodic data with and without RT guarantees when using a standard Ethernet switch, but requires a master for coordination. The Hard Real-Time Ethernet Switching Architecture (HaRTES) implementation uses an FPGA-based Ethernet switch with special firmware that takes over the role of the FTT master and ensures schedule compliance [31]. Further work extends FTT to multi-hop topologies with multiple HaRTES switches and investigates appropriate analysis methods and scheduling methods [32], [33], [34]. A protocol specifically implemented for HaRTES switches for the reconfiguration of data flows in multi-hop topologies at runtime is presented in [35]. This is capable of responding dynamically to network changes without interrupting continuous RT communication. However, using a master limits scalability, which violates R2-NetScal. So far, the current research offers no suitable approach that meets all requirements.

The need for new approaches is also reflected in the standardization efforts of the IEEE, which are dedicated to the development of Time Sensitive Networking (TSN) technology, to enable RT communication through measures such as time-slot assignment by link, customizable routing strategies and simultaneous communication via separate links at layer 1-2 [36]. For this purpose, new functions of the forwarding elements such as switches are defined.

For the time synchronization of TSN devices, the IEEE 802.1ASrev sub-standard defines the gPTP as a generalization of the PTP. If switches support gPTP, they are referred to as Time-Aware Bridges. In order to ensure time-controlled communication, a component called time-aware shaper (TAS) defined in IEEE 802.11Qbv is implemented on TSN capable switches to enforce that configured real-time constraints for specific flows are followed. IEEE 802.1Q defines eight stream classes for Ethernet frames and according priorities (see Figure 1). On Time-Aware Bridges, for each class a queue and a related transmission gate are available. These gates can be opened (O) or closed (C) by the TAS according to a time-based control list. Thereby, certain output queues of a switch can be selected for transmission and sending in time slots can be realized.

In [37], TSN technology was examined with regard to their RT capability. Although hard RT scheduling can be achieved using schedules that need to be applied to the devices and switches, it requires a significantly more complex platform, which contradicts the requirement to achieve the lowest possible platform complexity stated in R5-SC.

The IEEE 802.1 standards do not yet define a complete network architecture. In [36], a possible network architecture is presented, see Figure 2, in which a central user configurator and a centralized network configurator, which in their roles correspond to an SDN controller, are to determine a config-

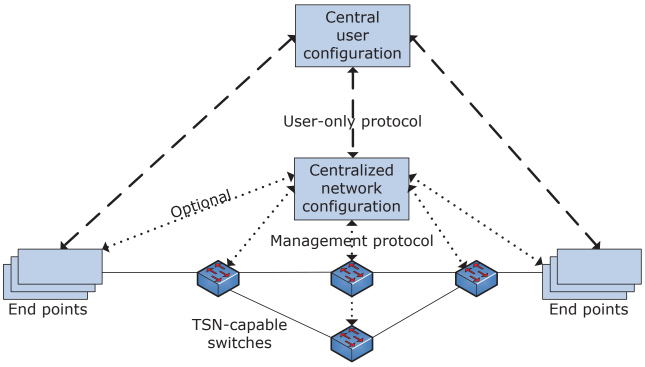

Fig. 2. Proposed TSN architecture according to [8].

uration of terminals and network infrastructure that enables communication in compliance with application requirements.

Overall, however, important questions remain unanswered: at present there are no standardized interfaces for the transmission of application requirements to the central management units. The specific functionality of these units and algorithms for determining an application-oriented network configuration are not part of the TSN standards either. Overall, TSN technology enables RT capability at layer 1 and 2 fulfilling R1-RT and is standard-compliant (R5-SC) but at the expense of introducing high platform complexity.

III. TECHNOLOGIES AT NETWORK AND TRANSPORT LAYER

For the sake of vertical and horizontal integration, it is indispensable that standard-compliant protocols are also used at layers 3 and 4 [1]. The IP protocol must be supported at the network layer to enable standard-compliant addressing of arbitrary devices in large networks such as the Internet [1] to meet R2-NetScal, R3-DatScal, R4-DR, and R5-SC. However, IP is only able to fulfill R1-RT and integrate non-RT devices into RT networks with additional mechanisms.

Differentiated services (DiffServ) provide a means to implement class-based quality of service locally on a router without considering the whole communication path [38]. Within an administrative domain, a set of service classes with respective forwarding rules is defined. Traffic flows registered for differentiated service are marked with the class they belong to. This information is entered into the DiffServ field of IP packets and evaluated at the routers. Classes are, however, solely characterized by their per-hop behavior and do not provide a network-wide QoS. Hence, RT capapility cannot be ensured thus violating R1-RT. As opposed to DiffServ, integrated services (IntServ) reserve resources in an end-to-end fashion so that QoS can be guaranteed for a traffic flow. The Resource reSerVation Protocol (RSVP) is used to implement the reservations on routers [39]. However, these service have hardly been installed. The reasons for this are manifold: among other things, mechanisms at application layer are easier to implement since the additional complexity and costs, i.e., R5-SC in terms of platform complexity, for installing and maintaining a network with IntServ functionality has been considered to be too high by Internet service provider, for whom the mechanism was originally designed.

Connection-oriented protocols such as TCP use flow and overload control when transferring data. As part of this flow and overload control of TCP, packet sizes and data rates are dynamically adjusted depending on the utilization of the receiver or the network. However, this readjustment depends on the communication channel is therefore not predictable and hence does not fulfill R1-RT. Further, TCP represents a stream-based approach, where large amounts of data must be transferred continuously and interruptions, which are needed to be able to adhere to time slots, are not envisaged although could be possibly configured. Since the Quick UDP Internet Connections (QUIC) transport protocol introduced by Google adopts the non-RT overload control at application layer from TCP [40], QUIC is also unsuitable for RT applications. Other transport protocols such as the Real-Time Transport Protocol (RTP) [41], the Datagram Congestion Control Protocol (DCCP) [42], and the Stream Control Transmission Protocol (SCTP) [43] also use non-RT overload control and are therefore unsuitable for hard RT applications.

In the existing RT Ethernet systems, standard protocols (TCP/IP or UDP/IP) are used for the transmission of non-time-critical IT data at layers 3 and 4, but special protocols are used for the transmission of RT critical process data [23]. These process data protocols are often system-specific (e.g., Profinet, EtherCAT, Powerlink) and provide time-deterministic data transfer. However, the protocols used are no longer compliant with standard Ethernet and do hence not fulfill R5-SC.

The choice of the transport protocol is challenging for hard RT requirements, since on the one hand the transport of large amounts of data will play a greater role in future scenarios (R3-DatScal), on the other hand, time-deterministic, predictable transmission times are to be assured (R1-RT). UDP can be extended to RT capability through application-layer TDMA mechanisms on the basis of sending individual independent datagrams (R1-RT). Moreover, UDP has to be further extended to provide a way to reassemble segmented data over multiple time slots, in order to meet R3-DatScal and to be able to integrate non-RT devices into RT networks (R6-NRT). In summary, for standard-compliant, RT communication, it is necessary to use UDP/IP at layer 3 and 4, while providing time-deterministic data transfer, and reassembly of segmented data at the application layer [1].

IV. TECHNOLOGIES AT APPLICATION LAYER

At the application layer, web services provide a consistent open standard for interaction between different devices. The OPC UA is a prominent service-oriented architecture example that has been developed as an integration framework for heterogeneous systems enabling compliance with the Industry 4.0 paradigm and is widely used [5]. OPC UA uses the protocols HTTP and TCP for data transmission, of which HTTP will be analyzed in the following. In addition to enabling the transmission of data, UPC UA also enriches them with a machine-readable semantic description to allow for a self-description of devices.

The RESTful protocol HTTP is based on TCP and hence does not fulfill R1-RT. Furthermore, the Message Queue Telemetry Transport (MQTT) protocol published as OASIS standard should be mentioned, which is suitable for bringing the REST architecture to devices with limited resources and thus integrating them into the IoT. MQTT relies on a publish/subscribe approach and requires a connection-oriented protocol at the transport layer [44]. As already shown, however, none of the standard connection-oriented protocols meets R1-RT. An alternative is MQTT-SN, which is based on MQTT, adapted to sensor actuator networks, and uses UDP as a transport protocol. Thus, MQTT-SN is theoretically RT capable [45], but does not support segmentation and reassembly functionality for large amounts of data [46]. Therefore, MQTT-SN does not meet R3-DatScal.

The Constrained Application Protocol (CoAP), which has been standardized as RFC 7252 [47], is also suitable for bringing the REST architecture to devices with limited resources, and thus to integrate them into the IoT according to [1], [44], [48]. In [49], it has been shown that CoAP is four times faster or lighter and thus better suited for the IIoT than the Internet-wide protocols HTTP and HTTP/2 (SPDY). Although the development of CoAP was inspired by HTTP, it still shows significant differences. Unlike HTTP, it is binary application layer protocol. As a result, the CoAP Request and CoAP Response messages are significantly smaller than those of HTTP. Another significant difference is the use of UDP on transport layer. CoAP also provides the ability to segment and reassemble large amounts of data across multiple messages using the header block option [50], meeting R3-DatScal. Since the standardization of CoAP, different RT and non-RT capable implementations have emerged. One of the most important is the Californium stack, which was developed by Kovatsch et al. [51]. Our own research shows that Californium achieves only strongly fluctuating processing times of 500 ms on average when using the RT-JVM Jamaica from Aicas and therefore does not fulfill R1-RT [52]. In addition to Californium, other implementations have been developed that are widely used but have not been developed for RT operation and therefore do not meet R1-RT [53], [54], [55].

In what follows, we want to explain by example of our own research works how an approach could look like that is able to fulfill all requirements stated in the introduction. Observe that, e.g., even if the approach fulfills the requirement of RT capability this does not mean that it outperforms any of the existing solution in terms of achievable latencies but rather that deadlines can be met at all. We used the advantages of the P2P overlay network Kad for Ethernet-based communication in the automation environment and extended Kad by the property of RT capability and called Hard Real-Time Kad (HaRTKad) [56]. The main idea behind HaRTKad is to correlate the hash values assigned to each Kad node in the network to time slots resulting in a distributed TDMA mechanism. The hash value of a device is determined by MD5 algorithm from a device ID such as, e.g., the MAC address. In its time slot, a device can exchange collision-free data, which in principle enables RT communication and data exchange between the devices. Furthermore, the processing of data on a device has to be carried out in RT, i.e, within specified time limits. This was achieved through the use of an RT operating system. In summary, HaRTKad enables coordinated media access in hard real time through a completely decentralized approach. Devices autonomously determine their time slots in this approach, resulting in an extraordinarily high degree of self-organization. An example network consisting of four devices with HaRTKad functionality is apparent from Figure 3. However, the determination of the scalability with respect to the achievable number of devices remains an open research question (R2-NetScal).

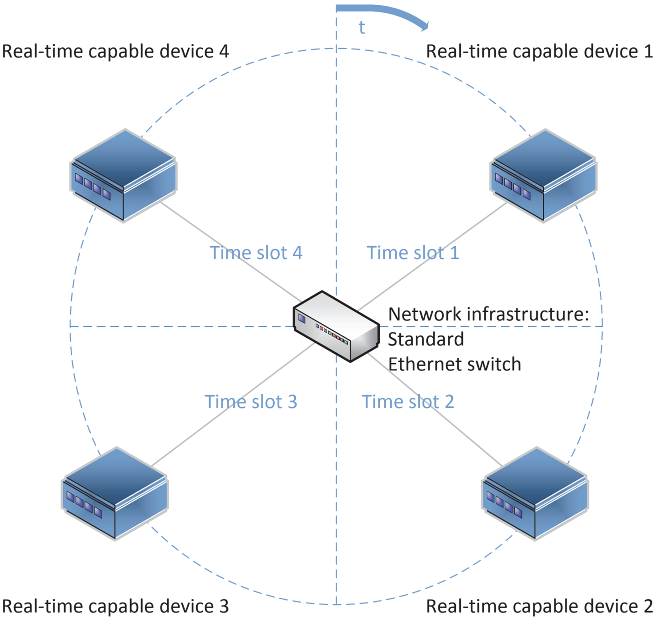

Fig. 3. An example network consisting of four devices with HaRTKad functionality.

CoHaRT, introduced in [57], extends HaRTKad with CoAP, enabling the RT capable Internet-compatible transmission of arbitrarily large amounts of data in segmented form (over several timeslots) and fulfilling R3-DatScal. However, devices that are not RT capable have not yet been considered in the CoHaRT concept (R6-NRT). With the addition of CoAP, the RT transmission of large amounts of data is also possible [50]. CoHaRT can interrupt the transmission at any time (for any period of time) by using caching queues until media access is allowed by time slot. Since UDP can only transmit individual packets and can not secure a continuous data stream, this task is taken over by CoAP at the application layer. CoHaRT itself thus provides a protocol for the transmission of large amounts of data. It already achieves guaranteed data rates of up to 65.72 Kbytes/s when transmitting 100 Kbytes between two devices

TABLE I

Comparison of existing systems and protocols with regard to the requirements stated in the introduction. ✓means that the respective requirement is fulfilled, in case of o this is possible with additional mechanisms, — means not fulfillable. *RT Ethernet industry solutions comprise Ethernet Powerlink [15], EtherCAT [16], TCnet [17], TTEthernet [18], CC-Link IE Field [19], Profinet [20], Ethernet/IP [21], and SERCOS III [22]. ** RT Ethernet research approaches include [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35].

ISO/OSI layer		**1. RT capability**	**2. Scalability (nr. of devices)**	**3. Scalability (data volumes)**	**4. Dynamic reconfiguration**	**5. Standard compliance**	**6. Integration of non-RT devices**
5-7	HTTP/2	—	✓	✓	✓	✓	o
	HTTP	—	✓	✓	✓	✓	o
	MQTT	—	✓	✓	✓	✓	o
	MQTT-SN	o	✓	o	✓	✓	o
	CoAP	o	✓	✓	✓	✓	o
	CoHaRT	✓	o	✓	✓	✓	o
4	TCP (QUIC, RTP)	—	✓	✓	✓	✓	o
	SCTP	—	✓	✓	✓	✓	o
	DCCP	—	✓	✓	✓	✓	o
	UDP	o	✓	o	✓	✓	o
3	IP	o	✓	✓	✓	✓	o
	DiffServ	—	✓	✓	✓	✓	o
	IntServ	✓	✓	✓	o	—	✓
1-2	Fieldbusses	✓	—	—	—	—	✓
	Standard Ethernet	o	✓	✓	✓	✓	o
	RT Eth. Ind.*	✓	o	—	—	—	✓
	RT Eth. Res.**	✓	o	—	—	—	✓
	TSN	✓	o	o	o	o	✓

connected via 1 Gbit/s Ethernet with a time slot of 1 ms per communication cycle. First analytical observations show that without optimization, cycle times of 50 ms can be achieved for 39 devices [57]. CoHaRT can therefore provide a transport mechanism for larger amounts of data to any RT application and thus act as middleware.

V. Future Research Directions

It can be concluded that there is a need for the advancement of existing or even for the development of new RT-capable approaches that meet all requirements, stated in the introduction, in future IIoT applications. An overview of existing systems and protocols is shown in Table I. A checkmark means that the respective requirement is fulfilled, in case of an circle this is possible with additional mechanisms, a hyphen means not fulfillable.

A purely software-based TDMA solution at the application layer represents an alternative to existing solutions that achieve RT capability at lower layers, possibly with complex platform requirements. However, please note that the advantages of a approach implemented at the application layer have to be traded off against the limitations regarding its performance as pointed out in [58]. It is potentially able to meet all requirements but its performance still needs to be determined in large-scale networks. The scalability with regard to the achievable number of devices and its capability to dynamically reconfigure the network at runtime without violating the real-time behavior should be investigated methodically (R2-NetScal and R4-DR). It remains to be examined to what extent an implementation exhibiting low platform complexity with hard RT properties for various RT operating systems and devices is feasible. It is specifically to determine which latencies in the communication between these devices can be reached and which scaling properties this methodology has. In addition, it is essential to investigate the possibility of incorporating non-RT capable devices (R6-NRT) without the network losing RT capability. The interoperability of non-RT capable devices with RT capable devices should be enabled in accordance with the standards. MQTT-SN and CoAP are also candidates for an RT-capable approach at application layer but need to be complemented by a TDMA mechanism to fulfill R1-RT. Additionally, once MQTT-SN or CoAP are RT-capable they would need to implement extra measures to integrate non-RT devices to meet R6-NRT such as the integration of appropriate mechanism at least in the switches, which receive best-effort traffic. Further, as mentioned above only end points of an RT application-layer approach adhere to time slots while standard Ethernet switches do not and time sensitive traffic cannot be prioritized. Standard Ethernet switches are neither time-controlled nor configurable and hence cannot be influenced concerning their switching behavior. This makes such an approach more suitable for networks with only RT traffic of small to medium size, in which the delay resulting from traversed switches can be taken into account to calculate the worst-case communication delay. On the contrary, the new IEEE standard technology TSN introduces new features into switches, which enable them to adhere to time slots and configurable routing rules as well as to prioritize RT traffic in front of best-effort traffic by its queuing and frame preemption mechanisms [8]. In combination with an SDN controller that calculates scheduling and routing rules, TSN-enabled switches allow for combined RT and best-effort networks of larger scale compared to an application-layer approach. However, these benefits are achieved at the expanse of introducing a central SDN controller and more complex, even though standardized technology into TSN capable switches. Especially, the high platform complexity might hinder a TSN implementation at large-scale since costs play a decisive role in future large-scale IIoT applications.

As part of future research, the advantages and drawback when using a centralized approach should be compared to the case when using a decentralized approach. A central instance like an SDN controller might be able to achieve better latencies for the RT communication than distributed approaches. It can leverage the network overview to compute a schedule for assigning concurrent time slot to nodes, whose communication takes place over non-overlapping network links, i.e., use SDMA in addition to TDMA. While this approach does not violate RT constraints, it can potentially increase the network utilization and at the same time meet the RT requirements of an increasing number of nodes. However, on the one hand the scalability of such an centralized approach is limited due to the imposed exponential complexity, which might complicate its application in large-scale networks, which violates R2-NetScal. We would like to remind the reader that the scalability and computational complexity is especially important if a network has to react to changing system parameters at runtime. On the other hand, the achievable performance of a distributed approach remains limited by the lack of a central instance, which has the overview over the whole network. We conclude there is a need for the advancement of existing and for the development of new, possibly hybrid, RT capable approaches that combine the advantages of centralized and distributed solution in order to meet all requirements.

VI. Conclusion

This survey investigated real-time capable Internet technologies for wired communication in the prospective IIoT. Therefore, we defined the requirements to be met as scalability and dynamic reconfiguration, standard compliance and platform complexity as well as their capability to integrate non-real-time devices in additional to real-time capable devices. It became apparent that none of the investigated Internet technologies meets all requirements. Hence, research questions to be answered in the future comprise but are not limited to the following itemization.

- Can lower-layer adaptations be dispensed in large networks of the future IIoT if the real-time capability is ensured at application layer while avoiding high platform complexity imposed by technologies such as TSN?
- Can a high-performance real-time system be implemented in a distributed way to achieve high scalability, is a centralized approach such as using an SDN controller feasible, or is it even better to realize hybrid solutions?
- Which system properties are decisive here for the achievable real-time capability?
- Can large amounts of data be transferred in guaranteed time?
- Can non-real-time devices be integrated in a dynamic and self-configuring manner without real-time capable devices losing their real-time capability if an application-layer approach is used?
- Which mechanism for incorporating non-real-time capable devices is best for real-time capability and scalability if an application-layer approach is used?

References

[1] M. R. Palattella, N. Accettura, X. Vilajosana, T. Watteyne, L. A. Grieco, G. Boggia, and M. Dohler, “Standardized protocol stack for the internet of (important) things,” *IEEE communications surveys & tutorials*, vol. 15, no. 3, pp. 1389–1406, 2013.

[2] P. C. Evans and M. Annunziata, “Industrial internet: Pushing the boundaries,” *General Electric Reports*, 2012.

[3] Communication Promoters Group of the Industry-Science Research Alliance, acatech - National Academy of Science and Engineering, “Recommendations for implementing the strategic initiative INDUSTRIE 4.0,” Industrie 4.0 Working Group, Tech. Rep., April 2013.

[4] 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 Industrial Electronics Magazine*, vol. 11, no. 1, pp. 17–27, 2017.

[5] J. Miranda, J. Cabral, S. Banerjee, D. Grossmann, C. F. Pedersen, and S. R. Wagner, “Analysis of opc unified architecture for healthcare applications,” in *Emerging Technologies and Factory Automation (ETFA), 2017 22nd IEEE International Conference on*. IEEE, 2017, pp. 1–4.

[6] P. Danielis, G. Dán, J. Gross, and A. Berger, “Dynamic flow migration for delay constrained traffic in software-defined networks,” in *GLOBECOM 2017-2017 IEEE Global Communications Conference*. IEEE, 2017, pp. 1–7.

[7] J. W. Guck, A. Van Bemten, M. Reisslein, and W. Kellerer, “Unicast qos routing algorithms for sdn: A comprehensive survey and performance evaluation,” *IEEE Communications Surveys & Tutorials*, 2017.

[8] G. A. Ditzel and P. Didier, “Time sensitive network (tsn) protocols and use in ethernet/ip systems,” in *2015 ODVA Industry Conference & 17th Annual Meeting*, 2015.

[9] N. G. Nayak, F. Dürr, and K. Rothermel, “Time-sensitive software-defined network (tssdn) for real-time applications,” in *Proceedings of the 24th International Conference on Real-Time Networks and Systems*. ACM, 2016, pp. 193–202.

[10] Anybus News, “Industrial Ethernet is now bigger than fieldbuses,” 2018. [Online]. Available: https://www.anybus.com/about-us/news/2018/02/16/industrial-ethernet-is-now-bigger-than-fieldbuses

[11] G. P. Fettweis, “The tactile internet: Applications and challenges,” *IEEE Vehicular Technology Magazine*, vol. 9, no. 1, pp. 64–70, 2014.

[12] T. Sauter, “Integration aspects in automation-a technology survey,” in *Emerging Technologies and Factory Automation, 2005. ETFA 2005. 10th IEEE Conference on*, vol. 2. IEEE, 2005, pp. 9–pp.

[13] M. Felser, “Real time ethernet: Standardization and implementations,” in *Industrial Electronics (ISIE), 2010 IEEE International Symposium on*. IEEE, 2010, pp. 3766–3771.

[14] Modbus-IDA.ORG, “Modbus messaging on tcp/ip implementation guide v1.0b,” 2006.

[15] Ethernet Powerlink Standardization Group, “Ethernet Powerlink,” 2018. [Online]. Available: https://www.ethernet-powerlink.org/

[16] EtherCAT Technology Group, “Ethercat technical introduction and overview,” 2018. [Online]. Available: https://www.ethercat.org/en/technology.html

[17] Toshiba, “Tcnet technology,” 2018. [Online]. Available: https://www.toshiba.co.jp/sis/en/seigyo/tcnet/technology.htm

[18] TTech, “Time-Triggered Ethernet A Powerful Network Solution for Multiple Purpose,” Tech. Rep., April 2015.

[19] CC-Link Partner Association, “CC-Link Partner Association,” 2018. [Online]. Available: http://am.cc-link.org/en/index.html

[20] R. Pigan and M. Metter, *Automating with PROFINET: Industrial communication based on Industrial Ethernet*. John Wiley & Sons, 2008.

[21] ODVA, “ODVA,” 2018. [Online]. Available: https://www.odva.org/

[22] SERCOS, “sercos the automation bus,” 2018. [Online]. Available: https://www.sercos.org/

[23] P. Danielis, J. Skodzik, V. Altmann, E. B. Schweissguth, F. Golatowski, D. Timmermann, and J. Schacht, “Survey on real-time communication via ethernet in industrial automation environments,” in *Emerging Technology and Factory Automation (ETFA), 2014 IEEE*. IEEE, 2014, pp. 1–8.

[24] J. Farkas, “Introduction to ieee 802.1: Focus on the time-sensitive networking task group,” 2017.

[25] J. Kiszka and B. Wagner, “Rtnet-a flexible hard real-time networking framework,” in *Emerging Technologies and Factory Automation, 2005. ETFA 2005. 10th IEEE Conference on*, vol. 1. IEEE, 2005, pp. 8–pp.

[26] K. W. Schmidt and E. G. Schmidt, “Distributed real-time protocols for industrial control systems: Framework and examples,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 10, pp. 1856–1866, 2012.

[27] T. Hu, P. Li, C. Zhang, and R. Liu, “Design and application of a real-time industrial ethernet protocol under linux using rtai,” *International Journal of Computer Integrated Manufacturing*, vol. 26, no. 5, pp. 429–439, 2013.

[28] R. Schlesinger and A. Springer, “Vabs-a new approach for real time ethernet,” in *Industrial Electronics Society, IECON 2013-39th Annual Conference of the IEEE*. IEEE, 2013, pp. 4506–4511.

[29] G. Carvajal, M. Figueroa, R. Trausmuth, and S. Fischmeister, “Atacama: An open fpga-based platform for mixed-criticality communication in multi-segmented ethernet networks,” in *Field-Programmable Custom Computing Machines (FCCM), 2013 IEEE 21st Annual International Symposium on*. IEEE, 2013, pp. 121–128.

[30] P. Pedreiras and A. Luis, “The flexible time-triggered (ftt) paradigm: an approach to qos management in distributed real-time systems,” in *Parallel and Distributed Processing Symposium, 2003. Proceedings. International*. IEEE, 2003, pp. 9–pp.

[31] M. Ashjaei, M. Behnam, P. Pedreiras, R. J. Bril, L. Almeida, and T. Nolte, “Reduced buffering solution for multi-hop hartes switched ethernet networks,” in *2014 IEEE 20th International Conference on Embedded and Real-Time Computing Systems and Applications*, Aug 2014, pp. 1–10.

[32] M. Ashjaei, P. Pedreiras, M. Behnam, R. J. Bril, L. Almeida, and T. Nolte, “Response time analysis of multi-hop hartes ethernet switch networks,” in *Factory Communication Systems (WFCS), 2014 10th IEEE Workshop on*. IEEE, 2014, pp. 1–10.

[33] M. Ashjaei, L. Silva, M. Behnam, P. Pedreiras, R. J. Bril, L. Almeida, and T. Nolte, “Improved message forwarding for multi-hop hartes real-time ethernet networks,” *Journal of Signal Processing Systems*, vol. 84, no. 1, pp. 47–67, 2016.

[34] M. Ashjaei, N. Khalilzad, and S. Mubeen, “Modeling, designing and analyzing resource reservations in distributed embedded systems,” in *Real-Time Modelling and Processing for Communication Systems*. Springer, 2018, pp. 203–256.

[35] M. Ashjaei, Y. Du, L. Almeida, M. Behnam, and T. Nolte, “Dynamic reconfiguration in hartes switched ethernet networks,” in *Factory Communication Systems (WFCS), 2016 IEEE World Conference on*. IEEE, 2016, pp. 1–8.

[36] G. A. Ditzel und P. Didier, “Time sensitive network (tsn) protocols and use in ethernet/ip systems,” 2015 ODVA Industry Conference & 17th Annual Meeting, 2015.

[37] S. Thangamuthu, N. Concer, P. J. Cuijpers, and J. J. Lukkien, “Analysis of ethernet-switch traffic shapers for in-vehicle networking applications,” in *Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition*. EDA Consortium, 2015, pp. 55–60.

[38] D. Grossman, “New terminology and clarifications for diffserv,” Tech. Rep., 2002.

[39] J. Wroclawski, “The use of rsvp with ietf integrated services,” 1997.

[40] G. Carlucci, L. De Cicco, and S. Mascolo, “Http over udp: an experimental investigation of quic,” in *Proceedings of the 30th Annual ACM Symposium on Applied Computing*. ACM, 2015, pp. 609–614.

[41] V. Jacobson, R. Frederick, S. Casner, and H. Schulzrinne, “Rtp: A transport protocol for real-time applications,” 2003.

[42] E. Kohler, M. Handley, and S. Floyd, “Datagram congestion control protocol (dccp),” Tech. Rep., 2006.

[43] T. Dreibholz, E. P. Rathgeb, I. Rungeler, R. Seggelmann, M. Tuxen, and R. R. Stewart, “Stream control transmission protocol: Past, current, and future standardization activities,” *IEEE Communications Magazine*, vol. 49, no. 4, 2011.

[44] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, “Internet of things: A survey on enabling technologies, protocols, and applications,” *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[45] K. Govindan and A. P. Azad, “End-to-end service assurance in iot mqtt-sn,” in *Consumer Communications and Networking Conference (CCNC), 2015 12th Annual IEEE*. IEEE, 2015, pp. 290–296.

[46] U. Hunkeler, H. L. Truong, and A. Stanford-Clark, “Mqtt-sa publish/subscribe protocol for wireless sensor networks,” in *Communication systems software and middleware and workshops, 2008. comsware 2008. 3rd international conference on*. IEEE, 2008, pp. 791–798.

[47] Z. Shelby, K. Hartke, and C. Bormann, “The constrained application protocol (coap),” 2014.

[48] L. Richardson, M. Amundsen, and S. Ruby, *RESTful Web APIs: Services for a Changing World*. ” O’Reilly Media, Inc.”, 2013.

[49] L. Daniel, M. Kojo, and M. Latvala, “Experimental evaluation of the coap, http and spdy transport services for internet of things,” in *International Conference on Internet and Distributed Computing Systems*. Springer, 2014, pp. 111–123.

[50] Z. Shelby and C. Bormann, “Block-wise transfers in coap,” 2015.

[51] M. Kovatsch, M. Lanter, and Z. Shelby, “Californium: Scalable cloud services for the internet of things with coap,” in *Internet of Things (IOT), 2014 International Conference on the*. IEEE, 2014, pp. 1–6.

[52] B. Konieczek, M. Rethfeldt, F. Golatowski, and D. Timmermann, “Real-time communication for the internet of things using jcoap,” in *Real-Time Distributed Computing (ISORC), 2015 IEEE 18th International Symposium on*. IEEE, 2015, pp. 134–141.

[53] M. Kovatsch, S. Duquennoy, and A. Dunkels, “A low-power coap for contiki,” in *Mobile Adhoc and Sensor Systems (MASS), 2011 IEEE 8th International Conference on*. IEEE, 2011, pp. 855–860.

[54] O. Bergmann, “libcoap: C-implementation of coap,” *URL: http://libcoap. sourceforge. net*, 2012.

[55] M. Kovatsch, “Coap for the web of things: from tiny resource-constrained devices to the web browser,” in *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*. ACM, 2013, pp. 1495–1504.

[56] J. Skodzik, P. Danielis, V. Altmann, and D. Timmermann, “Hartkad: A hard real-time kademlia approach,” in *Consumer Communications and Networking Conference (CCNC), 2014 IEEE 11th*. IEEE, 2014, pp. 309–314.

[57] J. Skodzik, P. Danielis, V. Altmann, B. Konieczek, E. B. Schweissguth, F. Golatowski, and D. Timmermann, “Cohart: A p2p-based deterministic transmission of large data amounts using coap,” in *Industrial Technology (ICIT), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1851–1856.

[58] B. Konieczek, J. Skodzik, P. Danielis, V. Altmann, M. Rethfeldt, and D. Timmermann, “Hartkad: A p2p-based concept for deterministic communication and its limitations,” in *2016 IEEE Symposium on Computers and Communication (ISCC)*, June 2016, pp. 1157–1162.