QWIN: Facilitating QoS in Wireless Industrial Networks Through Cooperation

Martin Serror, Eric Wagner, René Glebke, Klaus Wehrle
Chair of Communication and Distributed Systems, RWTH Aachen University, Germany
{serror, wagner, glebke, wehrle}@comsys.rwth-aachen.de

***Abstract*—For successfully establishing wireless communication in industrial environments, new approaches supporting the stringent requirements of industrial machine-to-machine communication are needed. Thereby, the main challenge is that different applications with distinct requirements compete against each other on the same wireless communication medium. Then again, an essential property of industrial scenarios is that the participating stations typically collaboratively work toward a common goal. In this paper, we thus investigate QWIN, a novel approach that leverages this cooperative nature by enabling the stations to share the scarce transmission resources. The stations hence offload their priority queues into the network and share them according to the quality-of-service requirements imposed by the overlying industrial applications. We implemented the cooperation mechanisms on prototypical hardware and evaluated them in a real-world testbed and by simulations. The evaluation reveals that our distributed decision approach effectively ensures that higher priority messages are conveyed more reliably within 1 ms, without reducing the reliability of lower priority messages.**

***Index Terms*—cooperative diversity, URLLC, wireless industrial networks, QoS, M2M communications, WARP**

I. Introduction

Wireless communication offers mobility and high data rates at low installation and maintenance costs. In the industrial domain, however, wired communication is still prevalent since, in contrast to wireless, it ensures high reliability and low latency [1]. Critical closed-loop control systems, which rely on Machine-to-Machine Communications (M2M), might require a communication latency down to $1\,\mathrm{ms}$ with a Packet Error Rate (PER) below 10^{-9} [2]. Although a few wireless communication protocols for industrial automation have emerged, e. g., WirelessHART and ISA100.11a [3], these protocols do not meet the aforementioned stringent communication guarantees that some industrial applications require. New communication protocols that fulfill these challenging requirements are therefore needed to benefit from the flexibility and cost reduction of wireless communication in the industrial domain [4]. The upcoming 5G standard refers to these requirements as Ultra-Reliable Low-Latency Communication (URLLC) [5].

Automation processes commonly consist of various stations carrying out a variety of tasks with distinct requirements, yet all stations typically share a single communication medium. While the exact requirements may vary from setup to setup, a safety-critical task, for example, usually has much more stringent requirements regarding communication latency and reliability than a simple task for logging purposes. A wireless communication protocol for industrial scenarios should, therefore, cater to the requirements of the respective tasks at hand to efficiently distribute the available wireless communication resources according to the actual needs.

Besides the communication requirements, there is another fundamental property of industrial networks: Although industrial networks typically consist of multiple heterogeneous devices, these devices collaboratively pursue a common goal, i. e., keeping the production process going. This feature opens up new opportunities for achieving URLLC when connected devices do not longer selfishly aim at maximizing the performance of their connections at the expense of others, but instead locally cooperate by sharing the available transmission resources according to the application requirements.

Previous research demonstrated that a promising approach for achieving URLLC is the use of cooperative diversity [6], [7], [8], in which wireless stations assist each other during the transmission process, e. g., when the link between a sender and a receiver is temporarily disturbed. In a simple form of cooperation, a third overhearing station with better link conditions relays the messages from sender to receiver in such situations. Cooperation mechanisms like relaying, however, require resources, e. g., wireless transmission time, which the stations have to share locally. Therefore, one must carefully decide how to assign the shared transmission resources depending on the individual application requirements.

In this paper, we thus propose *QoS in Wireless Industrial Networks (QWIN)*, a new approach supporting a wide range of distinct application requirements by leveraging cooperation. The main assumption for QWIN is that the participating stations aim at a common goal: the cooperative and distributed sharing of scarce transmission resources by offloading priority queues into the network. Instead of relying on a centralized resource scheduler, the stations indicate for each packet the respective priority and deadline. The other stations, which are overhearing ongoing transmissions, use relaying and other cooperation mechanisms to assist a sending station in conveying the packet to its destination. Since transmission time on the shared medium is limited, each station locally decides whether to transmit its own or relayed packets based on the priority and deadline as well as an estimation of the current Channel State Information (CSI) of the local links.

We evaluate the performance of our approach in a real-world testbed with Software-Defined Radios (SDRs) and by simulations using `ns-3` [9]. The results show that with cooperation, even though we rely on a simple local decision approach, the transmission reliability of high priority messages increases by orders of magnitude without reducing the reliability of low priority messages. Additionally, we show that when the number of stations increases, QWIN benefits from increased cooperative diversity and thus can maintain a certain Quality of Service (QoS). In particular, the contributions are:

1) We survey the diverse communication challenges in industrial settings (Sec. II) and present the related work addressing these requirements (Sec. III).
2) We propose QWIN, a simple and effective cooperative approach handling the stringent requirements of industrial applications in wireless networks (Sec. IV).
3) We empirically evaluate the effectiveness of QWIN in a prototypical testbed and through simulations (Sec. V).

Reliability

Latency	*More relaxed*	*More stringent*
More relaxed	General warehousing ($> 50\,\mathrm{ms}$, 10^{-2}) [13] ISA classes 4, 5 Supervision ($> 1\,\mathrm{s}$, "low") [14], [4] ISA classes 3, 4	Condition monitoring ($100\,\mathrm{ms}$, 10^{-5}) [13] ISA classes 4, 5 General field-level control (0.5–50 ms, 10^{-9}) [2] ISA classes 1, 2
More stringent	General maintenance ($> 20\,\mathrm{ms}$, 10^{-4}) [13] ISA classes 4, 5	Augmented reality ($10\,\mathrm{ms}$, 10^{-5}) [13] ISA class 3 Functional safety ($10\,\mathrm{ms}$, 10^{-9}) [13] ISA class 0 Motion control (0.25–1 ms, 10^{-9}) [13] ISA class 1

Fig. 1. Latency and reliability requirements of selected industrial applications.

II. Challenges

Industrial M2M scenarios impose challenging demands on communication infrastructures. The size of individual packets exchanged between machinery on the shop floor typically is in the range of a few bytes and thus lower than in home or office scenarios [10]. In contrast, the reliable control of complex and fast physical processes results in more strict limits regarding signal latencies and service interruptions than systems targeting human end users. In the following, we shortly elaborate on some fundamental properties of industrial scenarios and their implications on a potential wireless infrastructure.

Complexity of machinery: Industrial machinery may generate thousands of sensor readings and control signals per second that need to be transferred periodically [11]. Failures in the transmission process (e. g., due to fading or interference) may hence lead to wrong control decisions due to inconsistent assumptions about the state of the machinery within the controllers [12], or to states of unwanted behavior of actuators such as motors and valves. It is thus important that wireless communications in industrial settings acknowledges the complexity of the systems it caters for by: *(i)* keeping track of all signals the components of the system need to exchange, so that a communication schedule can be created; and *(ii)* allowing the prioritization of specific packets that are more important than others, e. g., emergency stop signals.

Process speed: The burden posed by the numerosity of the signals exchanged in industrial machinery increases with the speed in which they are generated. Typical signal update frequencies range from seconds for pressure sensors, over hundreds of milliseconds for temperature sensors [14], down to sub-millisecond ranges for vibration sensors, motor- or general machine control signals [11], [2]. Streaming data (which most of the exchanged signals in industrial systems are), in general, loses value when arriving late [15]. It is hence necessary that wireless industrial communication acknowledges the short-time stability of "uncontrolled" systems while keeping the communication as deterministic as possible.

Reliability and safety: Although the aforementioned room for variation exists, process and quality management, as well as occupational safety regulations, define further limits to the number of situations in which such variations may occur. While process monitoring for documentation or predictive maintenance purposes may call for maximum PERs of 10^{-4} to 10^{-5} with tolerable latencies between over $20\,\mathrm{ms}$ and $100\,\mathrm{ms}$ [13], processes related to the functional safety of the machinery—and hence the safety of the workforce—allow for at most $10\,\mathrm{ms}$ until appropriate actions are taken [13], with a maximum PER of 10^{-9} on average [2].

Fig. 1 provides a taxonomy for industrial applications concerning their latency and reliability demands. Multiple such classification schemes exist (e.g., [14], [2], [13]). The ISA-SP100 working group defined six generic classes based on the criticality of the respective applications [16], [17]. We include the ISA classes into our classification to show their relation. The varying dimensions and scenarios emphasize that when designing wireless industrial communications, it is not sufficient to optimize the behavior for just one use-case. Rather, it is important to keep in mind that industrial settings comprise several processes, both on- and offline, with humans in the loop and without, and that only the close cooperation of all these processes guarantees successful long term operation without interruption.

III. Related Work

Hence, the most challenging communication class for wireless industrial networks is URLLC, for its stringent requirements. Therefore, we first discuss related work on achieving URLLC and similarly challenging requirements for wireless communication. Afterward, we present the related work on supporting different wireless QoS classes.

A. URLLC

The authors of [18] propose to use best relay selection to achieve URLLC. They evaluate different selection schemes (namely periodic, adaptive, and reactive) regarding the achieved reliability. Their results indicate that the reactive scheme, where the best relay is calculated immediately after a direct transmission fails, achieves the highest reliability. This scheme requires, however, that instantaneous CSI is gathered and processed, typically leading to a higher transmission delay, which is not considered by the authors.

The authors of [19] propose to use network coding in the relaying process to reduce the number of required retransmissions per packet. By encoding k original packets into $k+m$ packets using Luby codes, it suffices to correctly receive any subset of k distinct packets to reconstruct the original k packets. Although the simulative results are encouraging, it would be interesting to see in a prototypical implementation if the number of retransmissions can be effectively reduced and how the encoding impacts the communication latency.

Occupy Cow [20] combines both, relaying and network coding, to achieve high reliability with low latency for industrial control. The analytic results are promising for URLLC. However, this approach makes strong assumptions regarding the time synchronization of the stations and there is a lack of prototypical evaluations to validate the results.

EchoRing [21] is a wireless token-passing protocol that achieves high reliability within a fixed latency of $10\,\mathrm{ms}$. An advantage of EchoRing is that it does not suffer from a single-point-of-failure because of its distributed coordination scheme. Nevertheless, the protocol is relatively inflexible regarding the provided communication guarantees, since it assumes that all traffic requires the same guarantees.

The Real-time Network Protocol [1] combines Carrier Sense Multiple Access (CSMA) and Time Division Multiple Access (TDMA) with different cooperative techniques that increase the reliability, such as relaying and piggybacking. The prototypical evaluation shows that a target latency of $100\,\mathrm{ms}$ achieves transmission reliability of $99.2\,\%$. This does, however, not suffice for URLLC applications.

The Bluetooth Low-Energy extension IO-Link [22] leverages frequency diversity to increase the transmission reliability by retransmitting a packet up to two times using different transmission channels. Although channel hopping, in general, effectively increases the communication reliability, the coordination between multiple coexisting channel-hopping systems is complex and thus not suited for industrial automation, typically consisting of various adjacent communication cells.

B. Wireless Traffic Prioritization

The WiFi extension IEEE 802.11e [23] offers QoS support at the Medium Access Control (MAC) layer. More specifically, it introduces a new Hybrid Coordination Function (HCF) supporting both a Contention Period (CP) and a Contention Free Period (CFP), where data traffic is handled according to its priority by four different access categories. In the CP, Enhanced Distributed Channel Access (EDCA) reduces the average time high priority messages have to wait for channel access using shorter inter-frame spaces. In the CFP, the HCF Controlled Channel Access (HCCA) extends the point coordination function, where the Access Point (AP) centrally schedules the channel access. Performance evaluations of IEEE 802.11e for industrial use cases in [24], [25] show that EDCA fails at providing low latency guarantees, especially with increasing network load. Likewise, the achieved latency in HCCA is too high for URLLC.

PriorityMAC [26] addresses critical traffic in wireless industrial networks by combining TDMA with different priority-based access approaches. The protocol supports four different traffic classes, where each class defines its own access method. To support emergency traffic, for example, stations may hijack slots from other stations with lower priority. Nevertheless, the highest priority class only achieves a PER of 10^{-4} with communication latencies between $6.9\,\mathrm{ms}$ and $14.2\,\mathrm{ms}$.

Similarly, EE-MAC [27] uses TDMA for deterministic channel access and supports emergency traffic upon request at a central coordinator. The analytical evaluation shows that EE-MAC effectively reduces the access delay of emergency traffic, while, on average, still causing a communication latency of several milliseconds due to the centralized approach.

The authors of PULS [28] also acknowledge that future wireless systems should support heterogeneous traffic consisting of strict and relaxed latency constraints. They tackle this problem by proposing an extensible wireless communication framework that performs scheduling at the software host and low-level MAC at the FPGA. Although PULS also targets deadlines down to $1\,\mathrm{ms}$, the measured loss ratio of $1\,\%$ does not suffice for URLLC. However, the concepts of QWIN could be integrated into PULS to increase the reliability further.

To sum up, recent related work partially addresses the challenges of URLLC and wireless traffic prioritization. A combination supporting a wide range of industrial applications with varying communication requirements (cf. Sec. II), however, is still missing. Hence, we are convinced that a cooperative approach, where the participants share the scarce transmission resources according to the application requirements, contributes towards flexible and efficient use of the wireless spectrum for industrial communication.

IV. DESIGN

In this section, we present *QWIN*, comprising different cooperation mechanisms at the MAC to achieve QoS in wireless industrial networks. We begin with the general assumptions that we make (Sec. IV-A). Then follows a description of the characteristics a deterministic MAC needs to exhibit to support our approach (Sec. IV-B), a description of our proposed cooperation mechanisms (Sec. IV-C), and finally, the scheduling process of QWIN (Sec. IV-D).

A. General Assumptions

We assume that the set of wireless stations, for which we would like to enable QoS, collaboratively works towards a common goal, e. g., maintaining a production process in a factory. Within this process, the stations have different tasks with different communication requirements (cf. Sec. II), where a single station may be responsible for more than one task, i. e., it generates messages of different priority classes and deadlines. For simplicity, messages that arrive past their deadline are considered useless for the respective overlying

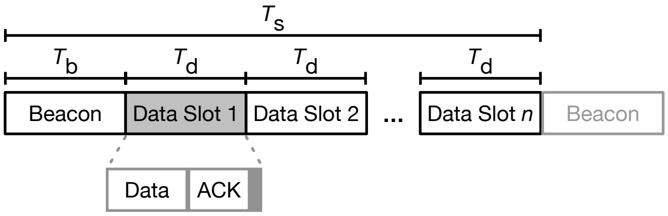

Fig. 2. TDMA superframe of the proposed MAC protocol. Each station is assigned one data slot in which it can transmit one data frame. The successful reception is then immediately acknowledged by the receiver. At the end of each data slot, there is a short guard time.

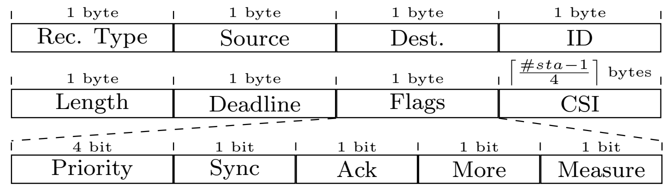

Fig. 3. Header of the data frame. The MAC header has a minimum length of 8 bytes. The length of the payload is set in the header, at most, to 255 bytes.

industrial application and thus discarded. In order to achieve the common goal, the stations may assist each other in the transmission process, which we refer to as *cooperation*.

Furthermore, we assume that the wireless stations have rather simple hardware characteristics, e. g., only a single transceiver, due to cost and size constraints. The stations are clustered into small communication cells according to their tasks, and all stations of the same cell are within communication range to each other. The stations communicate directly to each other, where the primary source for packet losses is fading and interference. Due to the broadcast nature of wireless communication, such stations may thus overhear transmissions from the other stations, opening further opportunities for cooperation. Although our approach would, in general, also work in multi-hop scenarios, this would require the integration of routing, which we do not consider in this work.

To facilitate the stringent communication guarantees that some of the automation tasks impose, the considered cooperation mechanisms should be implemented as close as possible to the Physical Layer (PHY). Only a few assumptions regarding the PHY need to be made: The packet-based PHY should provide error correction and measure the signal strength of received packets in order to assess the current link quality. Therefore, for the practical implementation, we selected a standard IEEE 802.11n PHY, which, however, may be replaced by any PHY that matches the criteria mentioned earlier.

B. TDMA-based MAC

The cooperation mechanisms of QWIN occur at the MAC layer, which needs to provide deterministic channel access for ensuring latency guarantees. For simplicity and as a proof of concept, we rely on a TDMA-based MAC, which offers determinism and is especially suited for typical industrial settings with periodic, small-sized data traffic (cf. Sec. II).

For the TDMA implementation, one of the stations assumes the role of the Central Coordinator (CC), responsible for assigning data slots to the connected stations. To coordinate medium access, we introduce a basic superframe structure, which we depict in Fig. 2. The superframe begins with a beacon slot broadcasting control information to the participating stations. Then follow n data slots where each belongs to one of the stations, including the CC, which may also transmit its data. Therefore, the fixed length of the superframe, e. g., $T_\mathrm{s} = 1\,\mathrm{ms}$, depends on the number of participating stations in the communication cell, the maximum supported length of the payload, and the transmission latency offered by the PHY.

1) Beacon Slot: For time synchronization and scheduling, the CC periodically sends a beacon at the beginning of each superframe. Stations receiving the beacon then adjust their internal clocks to the clock of the CC. Given the fixed duration of the superframe, a station can reset its timers even when it did not receive a beacon. Therefore, it suffices to receive a beacon once in a while to compensate for the clock drift.

The structure of the beacon frame consists of a message type, a sequence number, and a fixed assignment of data slots to stations. Changes in the schedule are announced repeatedly in the superframes preceding a change, so that all stations are informed even when they missed some beacons.

2) Data Slot: The data slot enables each station to send at least one data frame within the superframe. The length of the data slot suffices to transmit a data frame and for the receiving station to reply with an Acknowledgement (ACK). A data frame consists of a header and a payload. The header, shown in Fig. 3, includes the message type, the source, and the destination. Further, it specifies a message ID and the length of the payload. Then follows a deadline counter, which indicates the number of superframes the payload is still valid. That is, after each superframe, this counter is decremented, and eventually, the message is discarded. The header also includes a priority class field indicating the priority of the data. Finally, the remaining bits of the header are reserved for flags and CSI.

The `Sync` flag indicates whether the sending station successfully received a beacon for the current superframe. It allows overhearing stations, which missed that beacon, to resynchronize their clocks based on the known transmission schedule and superframe duration. Similarly, the sending station sets the `Ack` flag if it overheard an ACK frame during the last data slot (cf. Sec. IV-C3). The `More` flag indicates a piggybacked message (cf. Sec. IV-C2). The last flag may be used for measurement or debug purposes.

When the intended receiver successfully receives a data frame, it will immediately reply with an ACK frame. This frame only includes a message type, the original sender, and the message ID of the received data frame.

C. Cooperation Mechanisms

Next, we explain the cooperation mechanisms that we integrate into the TDMA-based MAC to increase the reliability of the data transmissions. In combination with local scheduling

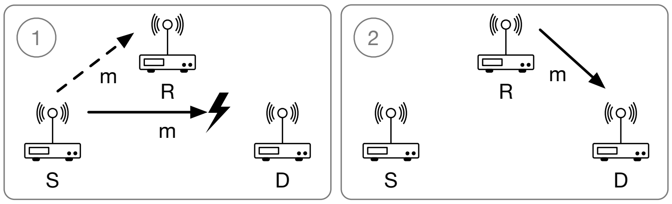

Fig. 4. Example for relaying: Sender S transmits a message m to destination D. Since the link between S and D is in a deep fade, D does not receive m. Relay R, however, overhears m. Afterward, R relays m to D.

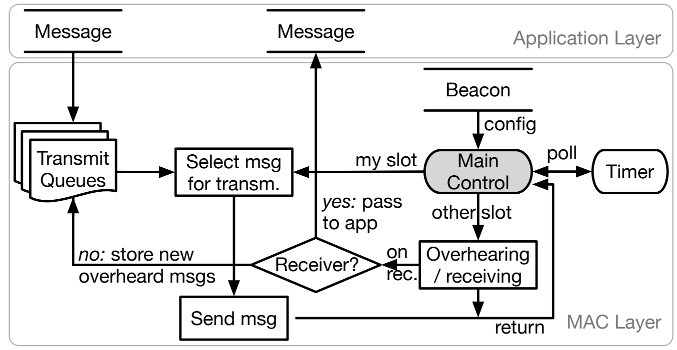

Fig. 5. General procedure of QWIN. The main control polls a timer to determine whether the station's data slot started. If so, it triggers the transmission of a message from the transmit queues. Otherwise, it enters into the overhearing / receiving mode. The transmit queues contain their own as well as overheard messages from other stations for relaying.

decisions of the stations (cf. Sec. IV-D), these cooperation mechanisms facilitate QoS for wireless industrial networks.

1) Relaying: A fundamental cooperation mechanism in wireless communications is relaying, describing the process of sending a message via a third cooperating station (a so-called relay) to the destination, instead of sending it directly from sender to destination. This type of spatial diversity is particularly useful when the direct link between the sender and the destination is in a deep fade, as shown in Fig. 4. The benefits of relaying depend strongly on the position of the relay. However, when the number of potential relays increases, the probability of a successful transmission also increases [29].

In our approach, we thus enhance the transmission reliability of a message by allowing it to be relayed multiple times by different stations, where any station within the communication cell may act as a relay. When a station acts as a relay, it can not send an own message anymore since the length of its data slot only suffices to transmit a single message. Consequently, each potential relay needs to include a relaying decision into its message scheduling process, for which we provide details in Sec. IV-D. Under certain conditions, however, a station may transmit two messages within one data slot, e. g., an own and a relayed message, which we explain in the next section.

2) Piggybacking: To further extend cooperation between the stations, we include the well-known concept of piggybacking into QWIN. Piggybacking, in general, refers to a method for increasing throughput by combining ACKs with regular data transmissions to avoid having to send an explicit ACK for a previously received packet. In our case, we use piggybacking to combine regular data transmissions with relayed transmissions. Hence, a station does not need to sacrifice its own data slot to act as a relay but instead sends an own message while also relaying a message from another station. In our TDMA-based protocol, the length of the superframe, as well as the length of the individual data slots, are fixed to achieve a deterministic medium access. Therefore, these two messages need to be transmitted within the same time that is normally used to transmit a single message, which can be achieved by selecting a weaker Modulation and Coding Scheme (MCS). To give an example, switching from Binary Phase-shift Keying (BPSK) with coding rate ½ to Quadrature Phase-shift Keying (QPSK) with coding rate ½ roughly doubles the data rate when both messages share the same PHY header. However, this also means that the transmission reliability decreases. Therefore, we only use piggybacking when (i) the estimated link quality to the main destination is strong enough that, despite using a weaker MCS, the transmission is likely to succeed and (ii) two messages of the same priority await transmission. Due to timing reasons, only the receiver of the first message will send an ACK within the data slot. The receiver of the second message has to save the ACK for the next time it is allowed to send an ACK (cf. Sec. IV-C3).

Since wireless links, in general, are not symmetric [30], each station estimates the current qualities of incoming links using the Received Signal Strength Indicator (RSSI) of received or overheard packets. Then each station includes the CSI of the $n-1$ links into the header of its next transmission (cf. Sec. IV-B2), allowing the other stations to overhear these recently measured link qualities. Consequently, stations with empty transmission queues need to send a "dummy" message, i. e., without payload, to allow the other stations to measure current link qualities and to receive the last measurements from the other stations.

3) Multiple ACKs: ACKs are used to signal the sender *and* the potential relays that a message successfully arrived at the destination. However, ACKs, just as regular data transmissions, are subject to fading and interference and might not arrive at their destinations. Missed ACKs lead to unnecessary retransmissions and thus waste transmission resources, which are consequently unavailable for needed retransmissions.

To mitigate this, we apply relaying to ACKs. Therefore, we extend the ACK frame such that two messages can be acknowledged. Whenever a receiver sends an ACK frame, it also includes the last pending ACK or the last overheard ACK. Moreover, each station sets the so-called `Ack` flag in the data frame header (cf. Fig. 3), when it overheard an ACK during the previous data slot. This informs other stations that the message of the last slot in the superframe was indeed acknowledged.

D. Scheduling Decisions

Fig. 5 illustrates the general procedure of QWIN. The selection of messages for transmission is mainly based on the given priority class and available deadline, which are both included in the message header (cf. Sec. IV-B2). The selection thus occurs in a distributed way, i. e., each station locally decides which message it will transmit next based on its transmit queues and a well-defined set of rules (not depicted in Fig. 5). In the following, we first describe how the stations manage the transmit queues, and then we explain how a station selects a message for transmission.

1) Transmit Queue Management: A station maintains for each priority class its own priority queue. Whenever there is a new message to transmit, this message is stored in the priority queue that corresponds to the message's priority class. Moreover, whenever the station overhears a message that is destined for another station, this message is also stored in the respective queue since the overhearing station may act as a relay. To avoid a priority queue overflow, messages that successfully arrived at their destinations and messages with elapsed transmission deadlines are deleted from the queues. The length of each priority queue depends on the available memory of the station, the respective priority, and the deadline. Furthermore, the expected number of stations within the communication cell may also be taken into account for the queue length because of the considered cooperation. If a new message arrives for a full priority queue, the oldest message is removed from that queue. The rationale for this is that, in industrial scenarios, we consider newer information to be more valuable than older information. (cf. Sec. II).

2) Transmission Priorities: In each data slot, the assigned station has to select a message out of all queued messages for transmission. Therefore, the station first determines the transmission queue with the highest priority that is not empty. Then it selects from its own messages in that queue the one with the shortest remaining deadline for transmission. If only overheard messages are present, it selects the one with the shortest deadline for relaying. The station thus favors its own over relayed messages because its messages might not have been sent out yet, while other stations might also have overheard relayed messages. Finally, based on the message receiver, the station checks whether the current link quality is strong enough to apply piggybacking (cf. Sec. IV-C2). If yes, it also selects a second message in the same way as before.

V. Evaluation

To evaluate the effectiveness of QWIN, we implemented a prototype on SDRs and measured its performance in a real-world testbed and through simulations. We first describe the evaluation setup and parameters. Then, we discuss the results.

A. Setup

For the implementation, we selected the Wireless Open Access Research Platform (WARP) v3 [31], which is an SDR consisting of Field Programmable Gate Arrays (FPGAs), two radio interfaces, and several I/O ports. The developers of WARP offer a reference design implementation for IEEE 802.11 [32], where the main parts of the MAC layer are realized on two MicroBlaze CPUs. We reused the provided PHY of the 802.11 Reference Design, which corresponds to

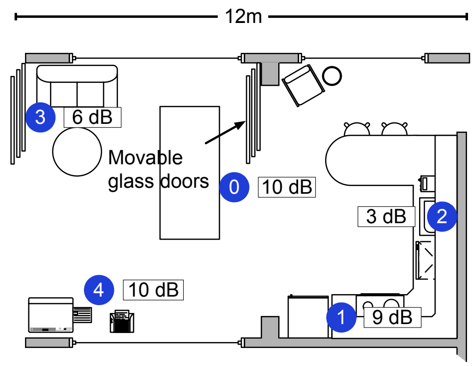

Fig. 6. Evaluation setup: The circled numbers represent the positions of the WARP boards, where ID 0 assumes the role of the CC. Next to each board, we indicate the selected antenna attenuator value.

TABLE I
Traffic Classes for Evaluation

Class	Description	Deadline	Occurrence
P0	safety-critical	$1\,\mathrm{ms}$	$10\,\%$
P1	closed-loop control	$10\,\mathrm{ms}$	$50\,\%$
P2	condition monitoring	$100\,\mathrm{ms}$	$20\,\%$
P3	supervision	∞	$20\,\%$

IEEE 802.11n with a single antenna while implementing the MAC layer according to the design presented in Sec. IV.

Regarding the evaluation setup, we placed five WARP boards into the social room of our research institute, as shown in Fig. 6. Each numbered circle corresponds to the position of a station with the respective ID, where station 0 assumes the role of the CC. The antennas are mounted at the ceiling and in line-of-sight to each other. However, the movable glass doors in the middle of the room might block the line-of-sight of some links. To artificially increase the distances between the boards, we additionally connect attenuators between the transceivers and the antennas. The respective attenuator values are shown in the boxes next to the stations. We set the length of the superframe $T_\mathrm{s} = 1\,\mathrm{ms}$, which corresponds to the latency bound of the traffic class with the highest priority. Since the beacon requires $T_\mathrm{b} = 65\,\mu\mathrm{s}$, the duration of a single data slot (including guard times) is then $T_\mathrm{d} = 187\,\mu\mathrm{s}$ for five stations.

At the beginning of a superframe, each station randomly generates a packet for one of the given traffic classes and with a random receiver. Table I shows the definitions of the selected traffic classes and their occurrence probabilities. In the considered example, we do not impose any deadline for priority class P3. Nonetheless, stored packets of P3 are deleted when the corresponding buffer is full (starting with the oldest packet). We set the length of each transmit buffer to 5, i. e., each station stores up to five packets of each priority class. Furthermore, each station uses the same transmit power, which we set to a moderate value of $-2\,\mathrm{dBm}$ to reduce interference to (assumed) adjacent communication cells. Table II lists the used evaluation parameters.

TABLE II
EVALUATION PARAMETERS

Parameter	Value
#stations	5
Superframe duration (T_s)	1 ms
Data slot duration (T_d)	187 µs
MAC header size (D_h)	8 B
Size of payload (D_{pl})	28 B (P0), 56 B (P1-P3)
Transm. bandwidth (B)	20 MHz
Center frequency (f_c)	5700 MHz
Transm. power (P_{Tx})	−2 dBm
Size of transmit queues	5 packets per priority class
Selected MCS	BPSK ½ (P0), QPSK ½ (P1-P3)

For the evaluation, we are mostly interested in the measured PER of different QoS classes from the application perspective, where packets that do not arrive within their deadline are considered to be useless, i. e., "lost". Consequently, for successful transmission, it does not matter how often the original packet was retransmitted or relayed, as long as the destination correctly receives it within the given deadline.

B. Empirical Measurement Results

In the following, we evaluate different aspects of QWIN. The measurements are based on the setup presented in Sec. V-A. For each evaluation, we thus only mention eventual changes we made to the original setup. We measured over a total of more than 300 hours. The measurements were carried out during the night to ensure a better reproducibility. Fig. 7 – Fig. 11 show the average PER for the different priority classes and all transmitted packets (total), as well as the respective 95 % confidence intervals.

1) Protocol Comparison: First of all, we are interested in the performance benefits of our cooperative system compared to approaches with less or no cooperation. Therefore, we compare QWIN to the following two variants:

Basic QoS. This protocol deactivates all cooperation mechanisms. It thus corresponds to a TDMA-based MAC, where each station uses its transmit queues for prioritizing its own messages, but does not use them for messages from others.

Relaying Only. Here, we extend Basic QoS by allowing relaying of messages by other stations. Piggybacking and multiple ACKs (cf. Sec. IV-C), however, are not enabled, which allows us to assess the performance improvements of relaying alone for achieving QoS in industrial settings.

We performed the measurements of the three protocol variants according to the descriptions in Sec. V-A. The results, depicted in Fig. 7, show that for each protocol variant, the average PER decreases for higher priority traffic. For Basic QoS, the differences between the priority classes are only about an order of magnitude because only the sender can improve the PER with retransmissions (no cooperation). For Relaying Only, substantial performance improvements between P0 and P1 compared to P2 and P3 are visible, showing that the high priority classes especially benefit from the relaying approach.

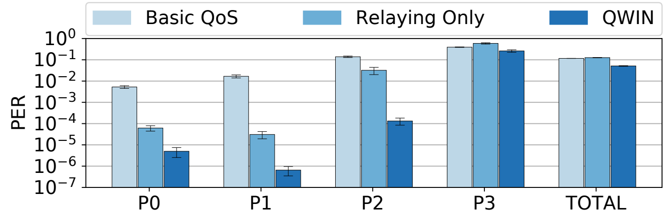

Fig. 7. Average PERs of the different protocol variants, by priority classes, and also showing the total average PER.

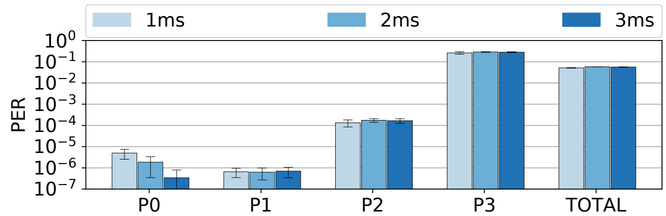

Fig. 8. Average PER for the different priority classes when increasing the latency of P0 (from 1 ms to 3 ms).

Since QWIN additionally relays ACKs (avoiding unnecessary retransmissions) and applies piggybacking, the average PER goes even further down. Note that for Relaying Only and QWIN, the average PER of P0 is higher than for P1. We attribute this to P0's low latency requirement of 1 ms, which coincides in our setting with the superframe length and thus limits the opportunities for relaying P0 messages.

2) Latency Requirements: To substantiate this observation, we now consider a more relaxed latency requirement for P0 messages. More specifically, we increase for QWIN the P0 latency bound of 1 ms to 2 ms, and even to 3 ms, to investigate the effects on the PER in this priority class and the impact on other classes. Otherwise, we conduct the measurements in the same way as before and with the same parameterization.

The results, depicted in Fig. 8, show that when the latency bound of P0 increases, the average PER of P0 decreases since the gained time is used for relaying. Although messages from P1, P2, and P3 do not benefit from P0's increased scheduling flexibility, neither does the average PER for these priority classes increase. Therefore, an effective way to improve the reliability for P0 is to reduce the slot length, which corresponds to increasing the relaying opportunities, since the stations are then scheduled more often. This can be achieved, e. g., by using a specialized low-latency PHY [33].

3) Superframe Utilization: Depending on the automation task, not every station might have a new message to transmit in each superframe. In this case, the opportunities for cooperation also increase, since these stations can still use their data slots to transmit messages from others, mainly because QWIN requires that each station regularly transmits a packet (cf. Sec. IV-C2). Therefore, in this evaluation, we measure the PER per priority class for different superframe utilizations. We thus introduce a probability with which each station generates a new message at the beginning of a superframe, which we vary between 80 %,

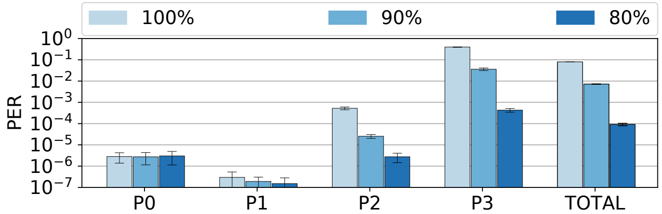

Fig. 9. Average PER for the priority classes when reducing the superframe utilization (from 100 % to 80 %).

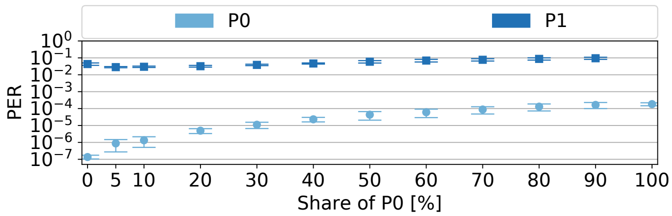

Fig. 10. Average PER for different shares of P0 messages. The two leftmost data points represent a measurement where only one P0 message is present per superframe.

90 %, and 100 %. The remaining evaluation parameters are the same as before (cf. Table II).

The results in Fig. 9 show that, in contrast to the evaluation presented in Sec. V-B2, reduced utilization of the superframe leads for the priority classes P1, P2, and P3 to a lower PER, since the opportunities for relaying increase, while the average PER of P0 remains roughly the same. We attribute this result to the observation that in a system that operates at full capacity, messages of lower priority are seldom relayed since the resources are needed for higher priority messages. In turn, reduced utilization of the superframe also enables the relaying of lower priority messages.

4) Share of P0 Traffic: So far, we set the safety-critical traffic (P0) to an occurrence probability of 10 %. However, depending on the automation task, a higher or a lower rate of safety-critical traffic might be present in the network. To investigate how QWIN handles different shares of safety-critical traffic, we simplify the definition of the priority classes (cf. Table I), by defining only two distinct classes, i. e., P0 with a 1 ms deadline and P1 with a 10 ms deadline. The first one represents safety-critical (high-priority) traffic, while the second one represents all other traffic of lower priority.

Fig. 10 shows the average PER for different shares of P0. Note that the two leftmost data points represent a measurement where the share of P0 is almost at 0 %, and the share of P1 is consequently almost at 100 %. To be more precise, we constructed a case where, in each superframe, only a single message of P0 is present in the network, i. e., when one station generated a P0 message, all other stations only generate P1 messages for this superframe. With this, we want to assess the achieved reliability when safety-critical messages do not compete against each other for resources in the network.

The results show that with an increasing share of P0, the average PER of P0 increases by several orders of magnitude

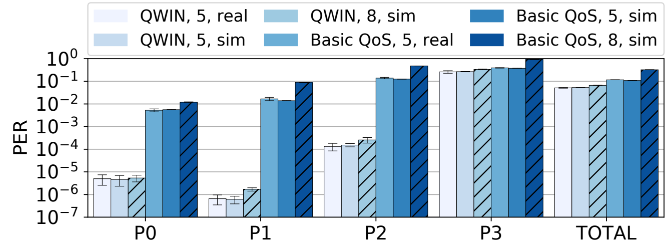

Fig. 11. Simulation results for QWIN and Basic QoS for 5 and 8 stations. As a reference, the plot also includes the real-world measurements for 5 stations (cf. Fig. 7). Note that hatches additionally highlight the results for 8 stations.

since more high-priority messages compete for the relaying. The PER of P0, however, does not reach the PER of P1 due to the more robust MCS used for P0 messages. For P1, the PER also increases when the share of P0 messages rises, because less P1 messages can be relayed when more P0 messages are present. Interestingly, the average PER of P1 slightly increases when the share of P1 messages approaches 100 %. Although this distribution enables more opportunities for relaying P1 messages, the relaying of P1 messages is less successful than relaying P0 messages, because of P0's more robust coding.

5) Increasing the Number of Stations: To evaluate the performance of QWIN when we increase the number of stations, we rely on the discrete-event simulator `ns-3` [9]. More specifically, we use a code-transparent extension [34], allowing us to simulate the same code in `ns-3` that we developed for the WARP boards. For comparability, we recreate the setup described in Sec. V-A and Table II using `ns3::NakagamiPropagationLossModel` with `ns3::LogDistancePropagationLossModel` to model the wireless channel. Increasing the number of stations within a fixed superframe duration of 1 ms, however, leads to smaller data slot durations for each station, which can be compensated, e. g., by using a less robust MCS. In the simulation, we hence increase the number of stations to 8 and, at the same time, use QPSK ½ for P0 and 16-QAM ½ for P1-P3, such that all eight transmissions fit into 1 ms.

Our simulation results for QWIN and Basic QoS are shown in Fig. 11. As a reference, we include simulation and real-world measurements (cf. Fig. 7) for five stations in the plot. Comparing simulation and real-world measurements, we see that for both QWIN and Basic QoS, the simulation results closely match the prototypical setup results. Moving to the results for eight stations, we note that although the coding is less robust, the PER of QWIN only marginally increases for the different priority classes. For Basic QoS, in turn, the PER increases about one order of magnitude. These results indicate that with QWIN, to a certain extent, the increased cooperative diversity when bringing more stations into the system mitigates the performance losses of weaker coding.

VI. Conclusion and Future Work

In this paper, we investigate QWIN, a cooperative approach to achieve QoS in wireless industrial networks. The key idea is that the participating stations cooperatively share the limited wireless spectrum, e. g., by using relaying and piggybacking, to fulfill the stringent requirements of different industrial applications. The advantage of this approach is that it works with heterogeneous devices with limited hardware characteristics and that it supports a wide range of requirements. The evaluation of QWIN, which we carried out in a real-world testbed and with simulations, shows that the transmission reliability of high priority messages increases by orders of magnitude without reducing the reliability of low priority messages. Moreover, when increasing the number of stations in the system, QWIN benefits from a higher cooperative diversity, which, to a certain extent, mitigates the reliability losses of weaker MCSs. These preliminary results thus show that cooperation in the context of URLLC, where the devices work towards a common goal, should be further investigated.

QWIN, nevertheless, can only be seen as a proof of concept for using cooperation to achieve wireless QoS. We thus anticipate that additional improvements, e. g., regarding the scheduling and the PHY [33], will further improve the results. Moreover, additional application requirements, such as security, also influence the latency and must be addressed accordingly in the communication stack [35], [36].

As a next step, it would be interesting to evaluate the concepts of QWIN in existing industrial deployments with sophisticated schedulers. Here, QWIN benefits from more relays, but it also has to take dynamic traffic and multi-hop communication into account. Altogether, we showed that cooperation allows us to effectively share the limited wireless spectrum in scenarios where the participants collaboratively aim at a common goal, fulfilling tasks of varying priority. Our approach hence further contributes towards reliable wireless communication for the industrial domain.

Acknowledgment

This research was supported by the research training group "Human Centered System Security" sponsored by the German federal state of North Rhine-Westphalia. The authors would like to thank the German Research Foundation (DFG) for the kind support within the REFLEXES project (ID 315171171) in Priority Programme 1914 "Cyber-Physical Networking".

References

[1] J. Silvo, L. M. Eriksson, M. Björkbom, and S. Nethi, "Ultra-reliable and Real-time Communication in Local Wireless Applications," in *IEEE IECON*, Nov. 2013.

[2] A. Frotzscher, U. Wetzker, M. Bauer *et al.*, "Requirements and current solutions of wireless communication in industrial automation," in *IEEE ICC Workshops*, Jun. 2014.

[3] S. Petersen and S. Carlsen, "WirelessHART Versus ISA100.11a: The Format War Hits the Factory Floor," *IEEE Industrial Electronics Magazine*, vol. 5, no. 4, pp. 23–34, Dec. 2011.

[4] M. Raza, N. Aslam, H. Le-Minh *et al.*, "A Critical Analysis of Research Potential, Challenges, and Future Directives in Industrial Wireless Sensor Networks," *IEEE Comm. Surv. Tut.*, vol. 20, no. 1, pp. 39–95, 2018.

[5] C. She, C. Yang, and T. Q. S. Quek, "Radio Resource Management for URLLC," *IEEE Comm. Magazine*, vol. 55, no. 6, pp. 72–78, Jun. 2017.

[6] C. Dombrowski and J. Gross, "EchoRing: A Low-Latency, Reliable Token-Passing MAC Protocol for Wireless Industrial Networks," in *IEEE European Wireless Conference*, May 2015.

[7] V. N. Swamy, S. Suri, P. Rigge *et al.*, "Cooperative Communication for High-reliability Low-latency Wireless Control," in *IEEE ICC*, Jun. 2015.

[8] M. Serror, S. Vaaßen, K. Wehrle, and J. Gross, "Practical Evaluation of Cooperative Communication for Ultra-Reliability and Low-Latency," in *IEEE WoWMoM*, Jun. 2018.

[9] The ns-3 Consortium, "ns-3 Discrete Event Network Simulator," [Online] https://www.nsnam.org.

[10] B. Galloway and G. P. Hancke, "Introduction to Industrial Control Networks," *IEEE Comm. Surv. Tut.*, vol. 15, no. 2, pp. 860–880, 2013.

[11] R. Glebke, M. Henze, K. Wehrle *et al.*, "A Case for Integrated Data Processing in Large-Scale CPS," in *IEEE HICSS*, Jan. 2019.

[12] A. Willig, K. Matheus, and A. Wolisz, "Wireless Technology in Industrial Netw." *Proc. of the IEEE*, vol. 93, no. 6, pp. 1130–1151, Jun. 2005.

[13] I. Aktas, A. Bentkus, F. Bonanati *et al.*, "Position Paper: Wireless Technologies for Industrie 4.0," VDE, Tech. Rep., 2017.

[14] J. Akerberg, M. Gidlund, and M. Bjorkman, "Future Research Challenges in Wireless Sensor and Actuator Netw. Targeting Ind. Automation," in *IEEE INDIN*, Jul. 2011, pp. 410–415.

[15] M. Stonebraker, U. Çetintemel, and S. Zdonik, "The 8 Req. of Real-Time Stream Processing," *SIGMOD Record*, vol. 34, no. 4, 2005.

[16] DIN, "Industrial networks – Wireless communication network and communication profiles – ISA 100.11a," Standard EN 62734, Jul. 2015.

[17] P. Zand, S. Chatterjea, K. Das, and P. Havinga, "Wireless Ind. Monitoring and Control Networks: The Journey So Far and the Road Ahead," *J. Sensor and Actuator Netw.*, vol. 1, no. 2, pp. 123–152, 2012.

[18] N. Marchenko, T. Andre, G. Brandner *et al.*, "An Experimental Study of Selective Cooperative Relaying in Industrial Wireless Sensor Networks," *IEEE Trans. on Ind. Inform.*, vol. 10, no. 3, pp. 1806–1816, Aug. 2014.

[19] S. Girs, E. Uhlemann, and M. Björkman, "Increased Reliability or Reduced Delay in Wireless Industrial Networks Using Relaying and Luby Codes," in *IEEE ETFA*, Sep. 2013.

[20] V. N. Swamy, P. Rigge, G. Ranade *et al.*, "Network Coding for High-reliability Low-latency Wireless Control," in *IEEE WCNCW*, Apr. 2016.

[21] C. Dombrowski, S. Junges, J. Katoen, and J. Gross, "Model-Checking Assisted Protocol Design for Ultra-reliable Low-Latency Wireless Networks," in *IEEE SRDS*, Sep. 2016.

[22] R. Heynicke, D. Krush, G. Scholl *et al.*, "IO-Link Wireless Enhanced Sensors and Actuators for Industry 4.0 Netw." in *AMA Conf.*, Jun. 2017.

[23] IEEE, "Standard for Information Technology–Local and Metropolitan Area Networks–Part 11: Wireless LAN MAC and PHY Specifications - MAC QoS Enhancements," IEEE Standard 802.11e-2005, Nov. 2005.

[24] H. Trsek, J. Jasperneite, and S. P. Karanam, "A Simulation Case Study of the new IEEE 802.11e HCCA mechanism in Industrial Wireless Netw." in *IEEE Conf. on Emerging Tech. and Factory Autom.*, Sep. 2006.

[25] G. Cena, L. Seno, A. Valenzano, and C. Zunino, "On the Performance of IEEE 802.11e Wireless Infrastructures for Soft-Real-Time Ind. Applications," *IEEE Trans. on Indust. Inform.*, vol. 6, no. 3, Aug. 2010.

[26] W. Shen, T. Zhang, F. Barac, and M. Gidlund, "PriorityMAC: A Priority-Enhanced MAC Protocol for Critical Traffic in Industrial WSANs," *IEEE Trans. on Indust. Inform.*, vol. 10, no. 1, pp. 824–835, Feb. 2014.

[27] M. Raza, H. Le-Minh, N. Aslam, S. Hussain, and W. Ellahi, "A Control Channel Based MAC Protocol for Time Critical and Emergency Communications in Industrial WSNs," in *IEEE C-CODE*, Mar. 2017.

[28] S. Yau, P. Hsieh, R. Bhattacharyya *et al.*, "PULS: Processor-Supported Ultra-Low Latency Scheduling," in *ACM Mobihoc*, Jun. 2018.

[29] A. Willig and E. Uhlemann, "On Relaying for Wireless Industrial Communications: Is Careful Placement of Relayers Strictly Necessary?" in *IEEE Int'l Work. on Factory Comm. Systems*, May 2012.

[30] N. Baccour, A. Koubâa *et al.*, "Radio Link Quality Estimation in WSNs: A Survey," *ACM Trans. on Sensor Netw.*, vol. 8, no. 4, Sep. 2012.

[31] Rice University and Mango Communications, "The WARP Project," [Online] https://warpproject.org/, 2019, last accessed 2019-03-25.

[32] Mango Communications, "802.11 Reference Design," [Online] https://mangocomm.com/802.11, 2019, last accessed 2019-03-25.

[33] H. Ji, S. Park, J. Yeo *et al.*, "URLLC in 5G Downlink: Physical Layer Aspects," *IEEE Wireless Comm.*, vol. 25, no. 3, pp. 124–130, Jun. 2018.

[34] M. Serror, J. C. Kirchhof, M. Stoffers, K. Wehrle, and J. Gross, "Code-transparent Discrete Event Simulation for Time-accurate Wireless Prototyping," in *ACM SIGSIM-PADS*, May 2017.

[35] A.-R. Sadeghi, C. Wachsmann, and M. Waidner, "Security and Privacy Challenges in Industrial IoT," in *ACM DAC*, Jun. 2015.

[36] J. Hiller, M. Henze, M. Serror *et al.*, "Secure Low Latency Communication for Constrained IIoT Scenarios," in *IEEE LCN*, Oct. 2018.