Improving MAC Protocols for Wireless Industrial Networks via Packet Prioritization and Cooperation

Jörg Christian Kirchhof[1,2], Martin Serror[1], René Glebke[1], and Klaus Wehrle[1]
[1]Communication and Distributed Systems, RWTH Aachen University, Germany
[2]Software Engineering, RWTH Aachen University, Germany
Email: kirchhof@se-rwth.de, {serror, glebke, wehrle}@comsys.rwth-aachen.de

***Abstract*—Stations in Cyber-Physical Systems (CPSs) and especially Industrial Internet of Things applications often work towards a common goal, but not all their tasks may be equally important to reach this goal. Hence, stations need to prioritize traffic because network resources are limited. To increase the service quality by utilizing otherwise unused network resources, stations may also opt to use cooperation instead of contention. Many existing standards and academic approaches focus on implementing either cooperation or Quality of Service (QoS) mechanisms. In this paper, we evaluate how to leverage cooperation to improve QoS. Since stations in wireless industrial applications often communicate locally, we focus on the MAC layer. We identify a set of useful cooperation mechanisms that increase the packet delivery ratio, and then extend them by several packet prioritization strategies and evaluate in multiple simulated industrial scenarios. As a result, we provide a set of guidelines for protocol designers to combine different mechanisms depending on the requirements imposed by industrial applications. Moreover, we provide and evaluate an exemplary combination of mechanisms derived from our results aiming at high reliability and low latency.**

***Index Terms*—Wireless Communication, Cooperation, Quality of Service, Internet of Things**

I. Introduction

Cyber-physical systems (CPSs) in the Industrial Internet of Things (IIoT) promise to improve industrial processes in terms of flexibility, productivity, and costs by enabling new services through the comprehensive interconnection of sensors, controllers, and actuators [1]. Locally, this interconnection relies on wireless machine-to-machine communication, *i.e.,* the periodic exchange of small sensor and control messages [2]. Unfortunately, wireless communication is inherently characterized by unreliable transmission links and non-deterministic latency, where the former is often caused by fading and interference, and the latter by contention-based channel access. This is, however, in stark contrast to the stringent requirements of many safety- and mission-critical industrial applications [3, 4], where the stations cooperatively aim at a common goal, *e.g., to keep the industrial process going*. Moreover, other tasks of less priority, such as monitoring and logging, may equally rely on the shared wireless communication medium and thus use valuable transmission resources. Hence, wireless communications in the IIoT must meet the various requirements of *heterogeneous* industrial scenarios, *i.e.,* support quality of service (QoS), to allow seamless operation of the served industrial processes.

Many industrial and academic wireless communication protocols implement some form of QoS. In IEEE 802.15.4 [5], a de facto standard for wireless communication in the Internet of Things (IoT) [6], guaranteed time slots (GTSs) provide some applications with a fixed bandwidth. The evaluation in [7] shows that IEEE 802.15.4 can be used in non-critical scenarios but cannot provide the required latencies for mission-critical applications. WirelessHART [8] is a standard based on IEEE 802.15.4 that implements QoS by using priority queues. PriorityMAC [9] uses short contention-based frames to allocate subsequent longer frames to stations. WiDOM [10] adapts the collision resolution mechanism known from Controller Area Network (CAN) bus [11] by assigning each message a priority, where the message with the highest priority gets medium access. This, however, requires a full-duplex transceiver, which is typically not given in wireless communications.

In summary, existing QoS approaches either rely on a central entity to manage resource allocation or let stations compete for network resources. The first option introduces a single point of failure, whereas the latter does not reflect the interest in voluntary cooperation, albeit the stations are sharing a common goal. Motivated by these shortcomings, we investigate a *cooperative* approach where stations support each other to achieve QoS in the entire network. While some approaches are vulnerable to selfishly acting stations (*e.g.,* [9, 10]), our approach allows stations to voluntarily cooperate with one another even if some stations are not participating. By enabling cooperation, the service quality can be increased with efficient distributed scheduling strategies and by leveraging otherwise unused transmission resources.

A widely used form of cooperation is relaying. CoopMAC [12] and rDCF [13] allow stations to select a relay, *i.e.,* a cooperating station, that should forward their packets. CODE [14] is based on CoopMAC and rDCF but allows the selection of multiple relays. The work in [15] further investigates cooperation strategies between relays. Correspondingly, we analyze cooperation between multiple relays. However, the absence of global knowledge about the network prevents stations from calculating optimal relaying decisions. Thus, we introduce additional cooperation mechanisms requiring only limited knowledge from the stations.

978-1-7281-7374-0/20/$31.00

Motivated by a prototypical evaluation showing promising results for using cooperation combined with QoS [16], this paper aims to analyze how to leverage cooperation to improve QoS. Therefore, we first implement a basic time division multiple access (TDMA)-based wireless medium access control (MAC) protocol using the network simulator ns-3. We then extend this protocol by several cooperation mechanisms. By evaluating their performance, we identify a useful set of such mechanisms. The mechanisms are then extended by several local packet prioritization strategies, *i.e.,* strategies that select the next packets to be transmitted by a station. This represents our approach to enable QoS. The results of the evaluation of the packet prioritization strategies are used to derive a set of guidelines for communication protocol developers to adapt communication protocols depending on the requirements of the served industrial processes. This paper makes the following contributions:

- We propose multiple cooperation mechanisms and packet prioritization strategies serving as building blocks that can be incorporated into existing communication protocols to enable QoS through cooperation (Sec. II);
- We evaluate the cooperation mechanisms (Sec. III-A) and packet prioritization strategies (Sec. III-B), including guidelines on when to employ which mechanism; and
- As an example, we compose and evaluate a new protocol, the MCC protocol, aiming at mission-critical communication (MCC) using suitable building blocks (Sec. III-C).

II. Design

Before answering the question of how cooperation can be leveraged for QoS, we first want to evaluate which cooperation mechanisms can be useful. Since many industrial standards use some form of priority queues for implementing QoS, we choose to implement QoS by also using priority queues. We devise several *packet prioritization strategies* that sort the packets in the queues and select the next packet(s) to be transmitted and evaluate which strategies or combinations of strategies are useful. From that knowledge, we derive guidelines on how to design a MAC protocol for wireless industrial communication as well as an exemplary MAC protocol based on the guidelines. To prevent optimizing these guidelines on only one concrete setup, it is necessary to test the mechanisms in multiple scenarios. We use the network simulator ns-3 for evaluating the performance of the numerous cooperation mechanisms and QoS introduced in the following sections.

A. Base Protocol Design

To assess the impact of each mechanism, we first implement a common *base protocol* with only a minimal set of features. We then extend it by the mechanisms under test. We do not include all functionalities required for use with real hardware into the base protocol, but only those features necessary for testing the different mechanisms in our simulations. For example, the protocol does not define how stations can join or leave the network. We choose a round-robin TDMA scheme to circumvent the hidden station problem that non-TDMA based QoS mechanisms (such as [9, 10]) can suffer from. However, note that the presented cooperation and prioritization strategies are not bound to TDMA, but would also work with any other deterministic MAC protocol.

As industrial applications often impose strict deadlines, each packet in our protocol includes a deadline in its header. If the intended receiver of a message receives the message before its deadline expires, the receiver sends an acknowledgment (ACK). ACKs are not sent as standalone messages but piggybacked in the header of messages. If a station has no data to transmit, it sends an empty message to indicate its continuing operation. Stations discard messages from their transmission queues when they receive an ACK for that message or when message's deadline expires.

B. Cooperation Mechanisms

Cooperation mechanisms allow stations to sacrifice (parts of) their resources to improve the overall performance of the network. To support cooperation between stations, stations use received messages to predict their one-hop neighbors. If a station receives a message, it considers the transmitting station to be a neighbor. Since channel conditions may change over time, the neighbor list is not static. Instead, stations are removed from the neighbor list after 100 ms unless another message arrives from the same station, which resets the timer.

We consider the following cooperation mechanisms:

a) Relaying: Stations that receive a packet that was meant to be received by a different station may forward that packet. While this means they have to sacrifice parts of their bandwidth, they may have a better link quality to the intended receiver and, therefore, enable delivering packets that could otherwise not be delivered by the station that originally transmitted the packet. We allow multiple stations to relay the same packet and allow stations to relay packets received from a relay. While we allow multi-hop relaying, we do not implement routing. Higher layers could, however, add routing.

b) Aggregation: By using different modulation and coding schemes, stations can choose to send messages with higher bandwidth at the expense of a higher error rate. In our implementation, stations can choose to either use quadrature phase-shift keying (QPSK) with a coding rate of $\frac{3}{4}$ or 16-quadrature amplitude modulation (16-QAM) with a coding rate of $\frac{3}{4}$. 16-QAM provides double the data rate of QPSK, allowing stations to send two packets in one timeslot with only one physical (PHY) header. Thereby, stations can relay packets of other stations without sacrificing one of their TDMA timeslots, at the expense of a less robust transmission.

c) Negative acknowledgements (NACKs): In contrast to ACKs, NACKs inform stations that a packet has *not* been delivered and needs to be retransmitted. Since our base protocol is based on TDMA, stations can inform their one-hop neighbors if they did not receive a packet from them during their last timeslot but expected this based on the periodicity of the TDMA schedule.

d) Slot Yielding: Some stations which are placed at crucial positions of a network may experience higher loads than other stations which are placed at the edge of the network. Slot Yielding enables stations to pass on their next timeslot to another station, which they expect to have a higher load. Stations piggyback information about their queue lengths in packet headers and can select another station to whom to yield their slots based on the received queue lengths. In our case, stations will only give up their slot if they have less than three packets in their queue while at least one neighbor has more than 15 packets in its queue.

C. Packet Prioritization Strategies

To enable both cooperation and QoS, we complement the cooperation strategies with several packet *prioritization* strategies. These strategies sort the transmit queue in a station. Decisions which packet to transmit next depend solely on the prioritization strategy. Note that stations only remove packets from their transmit queues if either an ACK is received or the packet's deadline expires. We consider the following packet prioritization strategies:

a) First In – First Out (FIFO): The packet that is queued first is the first to be transmitted.

b) Last In – First Out (LIFO): The packet that is queued first is the last to be transmitted.

c) Self-Generated First: Stations favor packets created by themselves over packets created by other stations.

d) Earliest Deadline First: The packet with the soonest expiring deadline is the first to be transmitted.

e) Latest Deadline First: The packet with the latest expiration date is the first to be transmitted.

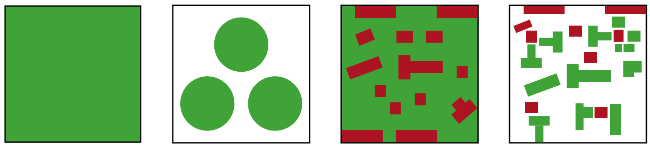
f) Highest Priority First: Packets may additionally be assigned a priority value in their headers. The packet with the highest priority value is the first to be transmitted.

g) Predicted Delivery First: If the intended receiver of a packet is a predicted one-hop neighbor (*cf.* Sec. II-B), the packet is favored over packets whose destination cannot be reached within one hop.

h) Least Transmission Attempts First: The packet which has been transmitted the least number of times is the first to be transmitted. All stations count the number of previous transmissions on a per-packet basis but do not exchange that count with other stations.

i) Signal-to-Noise Ratio (SNR)-based: Packets received with a lower SNR ($\lessapprox 10\,\mathrm{dB}$) are preferred for transmission, while packets that were received with a higher SNR ($\gtrapprox 20\,\mathrm{dB}$) are placed at the end of the transmit queue. This is based on the idea that stations may be placed in clusters: If a packet is received with a high SNR, a station assumes that the packet originated from a station close to itself (*i.e.,* in the same cluster). If received with a low SNR, it is assumed that the packet may be unknown to the other stations in the cluster.

j) En- and Discourage Requests: If a station predicts to be a one-hop neighbor of the intended receiver of a packet, it may inform other one-hop neighbors that further relaying this packet is unnecessary (discourage request). In contrast, if stations do *not* expect to deliver a particular packet, they may request their one-hop neighbors to prioritize this packet when choosing a packet to relay (encourage request).



(a) Random Placement (b) 3 Clusters (c) Factory Floor 1 (d) Factory Floor 2

Fig. 1. Evaluation scenarios. Each has a size of $50\,\mathrm{m} \times 50\,\mathrm{m}$. Green = Allowed station placement, Red = Obstacle, White = Free Space

III. Evaluation Study and Discussion

We evaluate the mechanisms described in the previous section in four scenarios (Fig. 1). Each station runs up to one application. We call stations that do not run an application *dedicated relays*. Each scenario consists of nine stations, where one acts as a central sink, four act as dedicated relays, and four run an application that generates one packet per superframe. As described in Section II-B, stations that run applications may also act as relays.

The *Random Placement* scenario is based on the findings of [17], stating that "the benefits of relaying are achievable even with randomly placed relayers, as long as enough of them are deployed." This is the only scenario in which stations are mobile, where mobility is implemented using ns-3's `RandomWaypointMobilityModel`. Since stations in industrial setups can often be assumed to be clustered, the *3 Clusters* scenario places stations randomly in one of three clusters. The two *Factory Floor* scenarios are adapted from the blueprint given in [18]. While the first scenario assumes stations are placed between obstacles, the second assumes stations are placed on machines.

Table II summarizes general simulation parameters and how the communication channel is modeled. The channel uses a constant speed propagation speed of the speed of electromagnetic waves in a vacuum. We use a log-distance [19] propagation loss model, where the reference path loss is calculated using Friis' formula [20] in the 5 GHz band. As error models, we use the `YansErrorRateModel` provided by ns-3. Additionally, we model block fading errors using the Gilbert-Elliot model, where the state change probabilities are set as in [21]. If the line-of-sight between two stations crosses an obstacle, we additionally subtract 15 dB from the received signal strength. This value matches ns-3's loss for external walls of type `ConcreteWithoutWindows`. Though our loss model does not account for effects like multipath propagation, it provides a rough estimate of the attenuation without adding a large computational overhead. The simulations for each combination of mechanisms and scenario are repeated 100 times with different random seeds. For comparability, the same 100 seeds were used for each combination of mechanism and scenario.

TABLE I

STATISTICAL EVALUATION OF COMPARING THE PACKET DELIVERY RATIOS OF COOPERATION MECHANISMS AGAINST FIFO USING A WILCOXON SIGNED-RANK TEST WITH BONFERRONI-HOLM CORRECTION AND A SIGNIFICANCE LEVEL OF 5%. GREEN CELL = SIGNIFICANTLY BETTER THAN FIFO

	Random Placement	3 Clusters	Factory Floor 1	Factory Floor 2
FIFO	$\mu = 0.87$	$\mu = 0.92$	$\mu = 0.26$	$\mu = 0.63$
FIFO + Relaying	$p <$ 2.2e-16; $\mu = 0.94$	$p <$ 2.2e-16; $\mu = 0.98$	$p =$ 9.518e-14; $\mu = 0.32$	$p <$ 2.2e-16; $\mu = 0.78$
FIFO + Aggregation	$p = 1.0$; $\mu = 0.82$	$p = 0.9556$; $\mu = 0.87$	$p =$ 5.656e-06; $\mu = 0.26$	$p = 0.0472$; $\mu = 0.62$
FIFO + Relaying + Aggr.	$p <$ 2.2e-16; $\mu = 0.96$	$p <$ 2.2e-16; $\mu = 0.99$	$p =$ 2.589e-14; $\mu = 0.36$	$p <$ 2.2e-16; $\mu = 0.84$
FIFO + NACKs	$p = 1.0$; $\mu = 0.83$	$p = 1.0$; $\mu = 0.89$	$p = 1.0$; $\mu = 0.26$	$p = 1.0$; $\mu = 0.61$
FIFO + Relaying + NACKs	$p = 0.9672$; $\mu = 0.87$	$p = 1.0$; $\mu = 0.90$	$p =$ 4.728e-05; $\mu = 0.31$	$p =$ 9.035e-13; $\mu = 0.72$
FIFO + Slot Yielding	$p = 1.0$; $\mu = 0.87$	$p = 1.0$; $\mu = 0.92$	$p = 1.0$; $\mu = 0.26$	$p = 1.0$; $\mu = 0.63$

TABLE II

SIMULATION PARAMETERS

Property	Value / Model
Transmission frequency	5.15 GHz
Propagation speed	$2.99792 \cdot 10^8 \frac{\text{m}}{\text{s}}$
Propagation loss	$L = -46.6777\,\text{dB} + 10 \cdot n \cdot \log_{10}(\frac{d}{d_0})$
Error rate	`YansErrorRateModel`
Error bursts	Gilbert-Elliot, $P_{bg} = 0.04$, $P_{gb} = 0.005$
Attenuation by obstacles	Reduce received signal power by 15 dB
`RxNoiseFigure`	7 dB (ns-3 default)
Reception threshold	-85 dBm (*cf.* [5, Clause 12.3.4])
TDMA slotlength	200 μs
Transmit Power	9 dBm
Packet Deadline	20 ms (= 100 slots)
Iterations	100 per mechan. / scenario combination
Number of Stations	9, *i.e.,* 1 sink, 4 dedicated relays, 4 apps
Packet generation freq.	One packet per superframe per app
Simulated Time	100 s

A. Cooperation Mechanisms

First, we evaluate the packet delivery ratios (PDRs), *i.e.,* the achieved reliability, of the different cooperation mechanisms. Table I shows the statistical results of comparing the PDRs of the mechanisms and some combinations of mechanisms against the PDRs achieved using FIFO using a Wilcoxon signed-rank test with Bonferroni-Holm correction and a significance level of 5%. Relaying improves the PDR significantly in all scenarios. While aggregation does not improve the PDR in most cases, combining aggregation with relaying improves the results compared to using only relaying. We attribute this to the fact that without relaying no station is overloaded as each application only generates one packet per superframe, which can be transmitted immediately. Combining aggregation with relaying gives stations the chance to either relay two packets within the timeslot and, hence, improves the performance compared to using only relaying. NACKs do not improve performance. Slot yielding neither improves nor reduces the performance, since the stations have similar loads without relaying. These results lead to the first guideline **(GL1)**: *Use relaying whenever stations should cooperate. Combining it with aggregation can improve performance if stations are overloaded. Use slot yielding when stations have different loads.* In the following experiments, we thus use the combination of relaying, aggregation, and slot yielding to evaluate the different packet prioritization strategies.

B. Packet Prioritization Strategies

Next, we compare the PDRs of different packet prioritization strategies against FIFO. Table III summarizes the statistical evaluation of our experiments. For comparison, this table also includes the results of the *MCC protocol*, which results from combining multiple packet prioritization strategies (*cf.* Sec. III-C). All strategies use relaying, aggregation, and slot yielding. One application generates packets with a higher priority than the other stations.

A disadvantage of LIFO is that packets might never get transmitted if they are not directly transmitted and the application continuously adds packets to the queue. While LIFO provides some of the worst PDRs, an additional latency evaluation however shows that those packets that actually are delivered using LIFO, are delivered with lower latency compared to the other prioritization strategies. From these findings, we derive our guideline **(GL2)**: *Use LIFO if low latencies are preferred over high PDRs.* Selfish behavior is advantageous if stations are so overloaded with higher priority traffic that they are completely prevented from delivering their own traffic. In our experiments, only one application has a higher priority than the other applications. Earliest Deadline First should be used if packets have very different deadlines. Latest Deadline First gives packets enough time to be relayed before their deadlines expire but penalizes packets with short deadlines. In our experiments, this effect is not visible, as packets have relaxed deadlines of 20 ms.

Highest Priority First overallocates resources to the application with the highest priority. This leads to a lower overall PDR but, in turn, also to lower latencies and a higher PDR for the application with the highest priority. We hence deduct guideline **(GL3)**: *Sorting by priority may lead to an overallocation of resources to high priority traffic. Combine this strategy with other strategies to limit its effects.* Predicted Delivery First performs exactly as FIFO because we used a central sink and hence, no packets were preferred over others. Least Transmission Attempts First outperforms FIFO in every scenario because it ensures that a large number of different packets is transmitted and, hence, prevents an overallocation of resources to one application. Overall, the SNR-based approach performs marginally better than FIFO. En- and Discourage requests perform better than FIFO as they

TABLE III

STATISTICAL EVALUATION OF COMPARING THE PACKET DELIVERY RATIOS OF PACKET PRIORITIZATION STRATEGIES AGAINST FIFO USING A WILCOXON SIGNED-RANK TEST WITH BONFERRONI-HOLM CORRECTION AND A SIGNIFICANCE LEVEL OF 5%. GREEN CELL = SIGNIFICANTLY BETTER THAN FIFO. ALL PACKET PRIORITIZATION STRATEGIES USED THE SAME COOPERATION METHODS: RELAYING, AGGREGATION, AND SLOT YIELDING.

	Random Placement	3 Clusters	Factory Floor 1	Factory Floor 2
FIFO	$\mu = 0.97869$	$\mu = 0.98713$	$\mu = 0.35619$	$\mu = 0.84314$
LIFO	$p = 1.0$; $\mu = 0.66085$	$p = 1.0$; $\mu = 0.65966$	$p = 0.002014$; $\mu = 0.38093$	$p = 1.0$; $\mu = 0.65379$
Self-Gen. First	$p = $ 3.289e-13; $\mu = 0.97907$	$p = $ 1.402e-05; $\mu = 0.98747$	$p = 0.9997$; $\mu = 0.35597$	$p = 0.8191$; $\mu = 0.84305$
Earliest DL First	$p = $ 8.961e-06; $\mu = 0.98033$	$p = 0.9996$; $\mu = 0.98258$	$p = $ 5.989e-07; $\mu = 0.36875$	$p = $ 1.876e-06; $\mu = 0.85347$
Latest DL First	$p <$ 2.2e-16; $\mu = 0.99464$	$p = 0.7622$; $\mu = 0.99438$	$p = $ 3.477e-12; $\mu = 0.43351$	$p = $ 1.989e-14; $\mu = 0.93192$
Highest Prio. First	$p = 1.0$; $\mu = 0.93676$	$p = 1.0$; $\mu = 0.94591$	$p = 0.8853$; $\mu = 0.34745$	$p = 1.0$; $\mu = 0.75728$
Pred. Deliv. First	$p = 1.0$; $\mu = 0.97869$	$p = 1.0$; $\mu = 0.98713$	$p = 1.0$; $\mu = 0.35619$	$p = 1.0$; $\mu = 0.84314$
Least Tx Att. First	$p <$ 2.2e-16; $\mu = 0.99560$	$p = $ 2.822e-07; $\mu = 0.99636$	$p = $ 1.346e-12; $\mu = 0.45832$	$p = $ 3.363e-16; $\mu = 0.97247$
SNR-based	$p <$ 2.2e-16; $\mu = 0.98376$	$p = $ 1.755e-05; $\mu = 0.98939$	$p = 0.04846$; $\mu = 0.36104$	$p = $ 1.503e-06; $\mu = 0.85012$
En- & Disc. Req.	$p <$ 2.2e-16; $\mu = 0.99703$	$p = $ 1.091e-06; $\mu = 0.99629$	$p = $ 3.07e-11; $\mu = 0.40675$	$p = $ 7.561e-16; $\mu = 0.92986$
MCC Protocol	$p <$ 2.2e-16; $\mu = 0.99911$	$p = $ 2.185e-07; $\mu = 0.99844$	$p = $ 1.19e-12; $\mu = 0.46645$	$p <$ 2.2e-16; $\mu = 0.97597$

limit superfluous transmissions. Relaying works best when stations have multiple options, which yields guideline **(GL4)**: *Use Least Transmission Attempts First to introduce different packets to the network. Also, allow stations to inform relays which packets they should (not) forward.*

We performed an additional fairness evaluation consisting of four stations that have to compete for the network resources of a single relay, which is also the only station that can reach the sink. We compare which portion of the delivered packets is generated by each of the four stations. Fairness is measured using Jain's fairness index (JFI) [22]. Using En- and Discourage Requests or LIFO leads to a lower fairness both when using aggregation (JFIs of 0.74 and 0.83 vs. 0.96 for FIFO) and when aggregation is disabled (JFIs of 0.41 and 0.47 vs. 0.81 for FIFO). Using aggregation increases fairness for every packet prioritization strategy with an average increase of 30%.

C. Combined Packet Prioritization Strategy

Following the above results, we design a combined packet prioritization strategy, the *MCC protocol*, with a focus on achieving high reliability. The MCC protocol uses the following packet prioritization strategies in descending order of priority: Discourage Requests Last, Predicted Delivery First, Self-Generated First, Least Transmission Attempts First, Encourage Requests First, Discourage High SNR, Latest Deadline First, Priority Per Slot Until Deadline, and FIFO. As an additional tie-breaker, we always choose the packet with the highest priority and the shortest deadline to be transmitted next. The protocol deliberately does not use any mechanisms other than those presented above to demonstrate their usefulness even without further optimization.

We evaluate the protocol in a traffic scenario with a high, bursty traffic load from the applications based on the Random Placement scenario. Table IV summarizes the traffic classes used in this scenario. All eight non-sink stations run an instance of an application with TC1. Additionally, four stations run a TC2 instance, two run a TC3 instance, and the remaining

TABLE IV

TRAFFIC CLASSES USED TO EVALUATE THE MCC PROTOCOL. SF = SUPERFRAME, F = FRAME

traffic class	Deadl.	Prio.	Packet Generation Frequency	Use Case
TC1	5 ms	4	1 Pkt / 2 Seconds	Safety-Critical
TC2	20 ms	3	1 Pkt / (SF + 1 F)	Uncritical Control
TC3	20 ms	2	1 Pkt / (SF + 1 F)	Uncritical Control
TC4	20 ms	1	1 Pkt / (SF + 1 F)	Uncritical Control

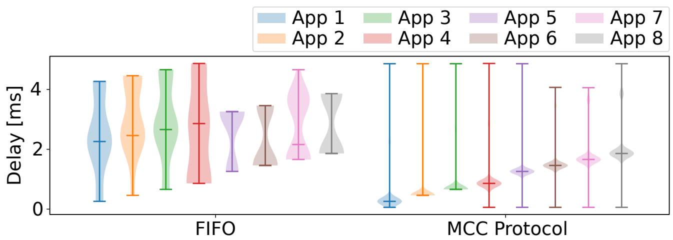


Fig. 2. Delay between packet generation and delivery observed in the burst pressure scenario for the safety-critical applications (*i.e.,* TC1).

two run a TC4 instance. Fig. 2 compares the latencies achieved by FIFO to the latencies achieved by our MCC protocol. The bellies in the violin plots are caused by the length of the timeslots. Fig. 3 compares the PDRs achieved for each instance of each application by FIFO and our MCC protocol. Our MCC protocol outperforms FIFO in every case, especially in the case of traffic with short deadlines (TC1).

In addition to the latency and reliability evaluation of our MCC protocol, we also conducted a scalability evaluation with up to 67 stations (Fig. 4). Precisely, the number of stations is given by $\{(2^n+m)\cdot 2+1 \mid n \in \{0,1,2,3,4,5\},\ m \in \{0,1\}\}$. The number of applications is adjusted for the number of stations so that there is always one dedicated sink and an equal number of stations running applications and stations acting as dedicated relays. Willig et al. [17], on whose work the

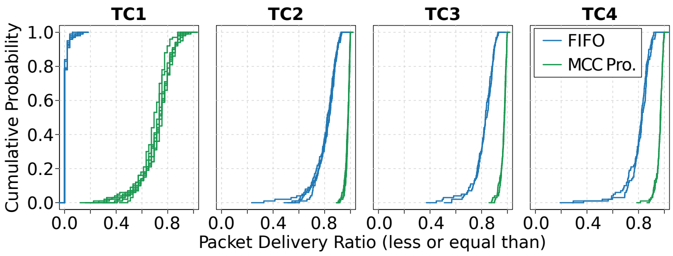


Fig. 3. Empirical distr. function of the packet delivery ratios observed in 100 runs of the bursty load scenario simulation. Every line describes the reliability of one application instance, every subplot shows one traffic class (TC).

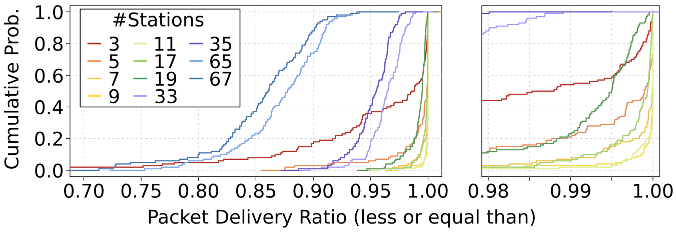


Fig. 4. Empirical distr. function of packet delivery ratios for different number of stations. Each number of stations simulated for 100 seeds. The right plot enlarges the region between 0.98 and 1.00 for better visibility.

random placement scenario is based, claim in their work that "four relayers can be considered as a 'critical mass', giving a reasonably high chance that there is at least one relayer having a good-quality channel to the central controller." Our scalability evaluation supports their claim. However, we also observe a saturation effect, which we attribute to the fact that more applications generate traffic and stations have more entries in their transmission queues due to more neighbors whose transmissions can be overheard. This leads to the following guideline **(GL5)**: *Provide enough relay stations, i.e., at least four.*

IV. Conclusion

This paper evaluates how cooperation can be leveraged on the MAC layer to increase the service quality of CPSs in wireless industrial scenarios by taking full advantage of scarce network resources. We propose a set of cooperation mechanisms and packet prioritization strategies that can be used to manage message priority queues, where each station takes local decisions based on limited knowledge about the network. By extending a basic TDMA-based MAC protocol, we analyze the performance of the cooperation mechanisms and packet prioritization strategies in a number of simulations inspired by industrial settings. Based on our results, we derive guidelines that can be used to develop new or improve existing wireless protocols. We demonstrate the viability of our guidelines by creating an exemplary MAC protocol for industrial settings, which we denote as mission-critical communication (MCC).

The evaluation of MCC shows that cooperation and packet prioritization can indeed achieve higher reliability and lower latencies compared to traditional, non-cooperative MACs. Future work should address evaluating the mechanisms using prototypical hardware and analyzing the influence of a dynamic TDMA schedule.

Acknowledgments

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) in the framework of the Priority Programme 1914 "Cyber-Physical Networking" – 315171171 – and under Germany's Excellence Strategy – EXC-2023 Internet of Production – 390621612. Simulations were performed with computing resources granted by RWTH Aachen University under project THES0461.

References

[1] J. Pennekamp, R. Glebke, M. Henze, *et al.*, "Towards an Infrastructure Enabling the Internet of Production," in *2019 IEEE Intl. Conf. on Industrial Cyber Physical Systems (ICPS)*, pp. 31–37, May 2019.
[2] 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.
[3] A. A. Kumar S., K. Ovsthus, and L. M. Kristensen., "An Industrial Perspective on WSNs–A Survey of Requirements, Protocols, and Challenges," *IEEE Comm. Surv. Tut.*, vol. 16, no. 3, pp. 1391–1412, 2014.
[4] M. Wollschlaeger, T. Sauter, and J. Jasperneite, "The Future of Industrial Communication: Automation Networks in the Era of the IoT and Industry 4.0," *IEEE Ind. Electr. Mag.*, vol. 11, pp. 17–27, Mar 2017.
[5] "IEEE Standard for Low-Rate Wireless Networks," *IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011)*, Apr 2016.
[6] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.
[7] G. Alderisi, G. Patti, O. Mirabella, and L. L. Bello, "Simulative assessments of the IEEE 802.15.4e DSME and TSCH in realistic proc. autom. scen.," in *IEEE Conf. on Ind. Inform. (INDIN)*, Jul 2015.
[8] "Industrial comm. networks - Wireless comm. network and comm. profiles - WirelessHART (IEC 62591:2016)," tech. rep., IEC, 2016.
[9] W. Shen, T. Zhang, F. Barac, and M. Gidlund, "PriorityMAC: A Priority-Enhanced MAC Protocol for Critical Traffic in Industrial Wireless Sensor and Actuator Networks," *IEEE Trans. on Ind. Informatics*, vol. 10, pp. 824–835, Feb 2014.
[10] N. Pereira, B. Andersson, and E. Tovar, "WiDom: A Dominance Protocol for Wireless Medium Access," *IEEE Trans. on Ind. Informatics*, vol. 3, pp. 120–130, May 2007.
[11] Robert Bosch GmbH, "CAN Specification, Version 2.0," 1991.
[12] P. Liu, Z. Tao, S. Narayanan, T. Korakis, and S. S. Panwar, "CoopMAC: A Cooperative MAC for Wireless LANs," *IEEE Journal on Selected Areas in Comm.*, vol. 25, pp. 340–354, Feb 2007.
[13] H. Zhu and G. Cao, "rDCF: A Relay-Enabled Medium Access Control Protocol for Wireless Ad Hoc Networks," *IEEE Trans. on Mobile Computing*, vol. 5, pp. 1201–1214, Sep 2006.
[14] K. Tan, Z. Wan, H. Zhu, and J. Andrian, "CODE: Cooperative Medium Access for Multirate Wireless Ad Hoc Network," in *IEEE Comm. Society Conf. on Sensor, Mesh and Ad Hoc Comm. and Networks*, Jun 2007.
[15] T. Jamal and P. Mendes, "Relay Selection Approaches for Wireless Cooperative Networks," in *IEEE 6th Int. Conf. on Wireless and Mobile Computing, Networking and Comm.*, pp. 661–668, Oct 2010.
[16] M. Serror, E. Wagner, R. Glebke, and K. Wehrle, "QWIN: Facilitating QoS in Wireless Industrial Networks Through Cooperation," in *IFIP Networking Conference*, June 2020.
[17] A. Willig and E. Uhlemann, "On Relaying for Wireless Industrial Communications: Is Careful Placement of Relayers Strictly Necessary?," in *IEEE Int. Workshop on Factory Comm. Sys.*, May 2012.
[18] L. Tang, K. C. Wang, Y. Huang, and F. Gu, "Channel Characterization and Link Quality Assessment of IEEE 802.15.4-Compliant Radio for Factory Env.," *IEEE Trans. on Ind. Inform.*, vol. 3, May 2007.
[19] T. Rappaport, *Wireless Communications: Principles and Practice*. New Jersey, USA: Prentice Hall PTR, 2nd ed., 2001.
[20] H. T. Friis, "A Note on a Simple Transmission Formula," *Proc. of the IRE*, vol. 34, pp. 254–256, May 1946.
[21] M. Nobre, I. Silva, and L. A. Guedes, "Reliability Evaluation of WirelessHART Under Faulty Link Scenarios," in *12th IEEE Int. Conf. on Ind. Informatics (INDIN)*, pp. 676–682, Jul 2014.
[22] R. K. Jain, D.-M. W. Chiu, and W. R. Hawe, "A Quantitative Measure of Fairness And Discrimination for Resource Allocation in Shared Comp. Systems," Tech. Rep. DEC-TR-301, Eastern Research Lab, 1998.