Harnessing Cooperative Anycast Communication for Increased Resilience in Wireless Control

René Glebke, Jan Scheiper, Stefan Lenz, Mirko Stoffers and Klaus Wehrle

***Abstract*— Closing control loops over wireless channels is a challenging task due to the inherent unreliability of the wireless medium. Interference caused by equipment or obstruction due to movement may quickly render proven good channels to fail temporarily, causing both sensor and controller signals to not reach their intended recipients. We argue that this notion of single "intended recipients" for wireless signals is at odds with the nature of control systems in which multiple plant components work towards a common goal. Furthermore, the associated unicast-based communication mechanisms ignore the potential benefits of the broadcast nature of the wireless medium. We hence develop a novel anycast-based communication system for industrial control, in which sent signals are potentially interesting to a multitude of recipient nodes. Our system enables to share replicas of the controller functionality among these nodes in order to improve the resilience of the control process via spatial diversity. Through simulation experiments, we show that our system can maintain a high quality of control despite deteriorating channel conditions, while at the same time requiring only a low coordination overhead.**

I. Introduction

Many industrial control processes require the flexibility of wireless communication but do not tolerate a low level of reliability. In this paper, we explore an opportunity to increase the reliability in wireless control scenarios, which, to the best of our knowledge, has not been investigated so far. We base our approach on two key observations.

Firstly, industrial control processes build on a set of sensors, actuators, and controllers, which – unlike devices in many other communication systems – all follow the same purpose and aim to collaboratively fulfill the control task. The contrary would be, for example, a website, where a specific end user wants to see that site and other end users do not care about the experience of that specific user. On the other hand, all devices in an industrial control process belong to the same owner, and this owner is interested in a smoothly operating control process, but not so much in the specific functionalities of the different devices.

Secondly, the nature of wireless communication is that messages are not directed to a single receiver, but can be overheard by every receiver in radio range. While this can be prevented by encrypting the data in a way that only the intended receiver can decrypt it, this seems not necessary when all devices are owned and controlled by the same entity. Nevertheless, even in industrial scenarios, wireless receivers are usually instructed to discard all incoming data frames that are not specifically addressed to them.

We summarize that industrial control processes based on wireless communication use a unicast mechanism where the sender of a data frame (e.g., a sensor) chooses a recipient (e.g., a controller) to send the signal to. At the same time, the broadcast nature of the wireless channel makes many more devices receive the content, yet all but one drop it. We argue that devices owned by the same entity and installed for a common purpose should not drop valuable information but use it to increase the overall reliability of the control process.

To this end, we propose a *cooperative anycast mechanism*. In this approach, information sources no longer decide upon an intended recipient but explicitly broadcast the information on the wireless channel, so that *any* device in radio range can and shall receive the information. We propose to install more than one device able to emit actuation signals – controller *replicas* – in wireless scenarios, and devise a protocol that enables re-requesting sensor data in cases of channel failures, as well as sending actuation signals of increasing qualities, both with a low coordination overhead. As replicas reside at different locations, their communication channels are uncorrelated, making it increasingly more likely for such cooperative message exchanges to lead to the successful generation and reception of actuation signals as the number of replicas increases. With this new paradigm, we leverage the properties we observed before in order to increase the reliability of control processes.

An alternative to our anycast approach would be relaying, which is, in essence, a method to reach a set of *fixed and previously-known* recipients. In contrast, with anycast we have multiple recipients, and any of them can eventually steer actuators to perform the proper control responses. We hence do not need relaying but only *completions* in case nodes have not received the required data to generate actuation signals.

To estimate the benefit of our approach, we perform a small case study simulating a factory floor operating under harsh conditions. We find that our cooperative system is able to successfully create and transmit control signals in 98.5% of the control loop rounds, while single or non-cooperative controllers would not be able to fulfill the task at all.

Structure: In Sec. II, we outline the problem of wireless control in more detail, discuss related work, and introduce our idea of shared responsibilities in wireless control. In Sec. III, we then describe our approach to an anycast-based wireless control system. We provide results from a simulative prototype as well as a discussion of limitations and possible improvements in Sec. IV, before concluding in Sec. V.

The authors are with the Chair of Communication and Distributed Systems, RWTH Aachen University, 52074 Aachen, Germany. Email: {glebke, scheiper, lenz, stoffers, wehrle}@comsys.rwth-aachen.de

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 315171171, 432169785.

II. A Case for Anycast Communication in Wireless Control

In communication systems, a device or a set of devices operated by one entity usually provides services to other devices, which may be under control of the same or another stakeholder. For example, web clients operated by end-users may request website data from some cloud provider, and the connection between the two runs via the devices of one or multiple Internet service or transit network providers. The stakeholders in such a scenario have inherently *individual* demands regarding the properties of the communication system, ranging from ease-of-use over scalability to monetary returns. We observe that communication in industrial control processes is orthogonal to this paradigm: Here, different devices (sensors, actuators, and controllers) *commonly* serve a control task; neither of the devices has any selfish interest, as an overall goal is to maintain control over the plant.

For the benefit of increased flexibility, many industrial control scenarios nowadays employ wireless communication. While this comes at reduced reliability, we on the other hand observe a beneficial property of wireless communication, which is aligned to the cooperative nature of control processes, but is commonly neglected: The broadcast nature of wireless channels enables every device within range to receive data that was initially directed to someone else.

When serving control tasks over a network, it is hence worthwhile to cooperatively optimize the likelihood for the controllers to succeed in eventually delivering proper actuator responses to sensor inputs, instead of focusing on the data communication between two nodes for the sake of selfish interest. In the following, we discuss how this has been tackled so far, before we derive the case for a novel direction.

A. Related Work

The unreliability of the wireless medium when targeting control applications has triggered vast interest by both the communication and the control systems communities. A fairly recent survey in [1] categorizes almost 250 works among various axes, highlighting both control and communication aspects. Options and design criteria the study found range from totally disjoint "black-box" designs of the control and communication aspects to tightly integrated systems that dynamically adapt their behavior during run-time based on feedback from either side, with varying degrees of realism also in the assumptions regarding wireless communication in the neighborhood or the type of transmitted data.

From a control perspective, the main challenges associated with signaling over a wireless (or otherwise unreliable) link are related to the unpredictable and potentially unbounded delays that signal losses cause, while the plant continues to operate. Techniques originally developed for wired networks such as the one in [2] monitor the network behavior and select among controllers specifically designed for the prevalent conditions (gain scheduling). Such approaches can be partially adapted to wireless networks [3], but suffer from the problem that observing the network from within the network itself is hard-to-impossible if the links suffer intermittent failures. Information on the network state directed at controllers hence suffer from fate sharing with the information transmitted for the control loops, so that controllers can only issue "post-hoc" reactions. Relying on network state information in large-scale wireless scenarios with centralized controllers is therefore hard. To mitigate these issues, responsibility for specific areas of a plant can be broken up into inherently *distributed* controllers such as in [4], which uses state estimation techniques to enable coordinated controller responses. However, even when controllers are made responsible only for their limited physical vicinities, the problem that channel degradations impair communication to and from these (sub)-controllers remains.

From a communication system point of view, control applications exhibit extremely demanding requirements. While the general size of transmitted data items may be low [5], extremely tight delay and packet error rate bounds (see e.g., [6] for an overview) paired with interference and obstruction caused by the equipment are challenging issues. As intermittent line-of-sight obstruction is problematic for maintaining reliable communication patterns, research has made attempts at diversifying the communication paths by using multiple channels at the same time [7], in order to increase the chance of at least one of the signals to arrive at the destination. Works such as the one in [8] in turn explore path diversification by explicitly allowing nodes to cooperate by e.g., transmitting data on behalf of other nodes (relaying). However, such approaches are dominated by the *end-to-end principle*, i.e., the nodes remain oblivious to the meaning of the data they are transmitting. The expected knowledge about transmitted data is increased in works such as Chaos [9], which lets nodes within a network perform all-to-all data sharing with simple on-path aggregation functionality but in turn heavily relies on physical-layer phenomena that may not be available in all communication systems [10]. By combining relaying, network coding and simultaneous transmissions, Occupy CoW [11] achieves promising information dissemination results in multi-hop networks, but assumes perfect synchronization between nodes, which may not be achievable in all cases. Furthermore, although including certain en-route data handling mechanisms, also the likes of Chaos and Occupy CoW do not explicitly consider that the nodes in the network might be interested in not just jointly computing functions, but jointly taking action.

Takeaway: Most of the currently-employed strategies for control and communication in industrial domains can be categorized as either *(a)* physically and logically centralizing information and decisions, or *(b)* dividing an overarching problem into sub-problems distributed among the nodes but again with single nodes responsible for information collection and sub-decision making. Thus, even when the decision process involves more than one entity, there remains an aspect of inherent non-involvement of the majority of entities besides providing input to or acting on signals from the controlling node(s). This is a stark contrast to the inherently inclusive natures of both control problems and of wireless networks, leading to a waste of readily-available resources.

B. From Distributed To Shared Responsibility

Based on our observations of both an *inherent interest* for involvement of nodes within industrial environments in the control process as well as an underutilized *passive ability* to do so in wireless communication scenarios, we propose to leverage these two facts by including a novel concept of *shared responsibility* to the design of future control and communication strategies within industrial environments. Instead of relying on the communication system to achieve information concentration on a limited number of controller nodes (distribution), our approach is to allow all capable and willing nodes to actively involve themselves in the decision process by hosting *replicas* of the control algorithms, with all the controller replicas in the system running *concurrently*.

In the remainder of this document, we show that in this setting, the broadcast nature of the wireless medium can cater these replicas simultaneously with either no (wrt. sensor signals) or very limited coordination overhead (wrt. actuation signals), whilst increasing the likelihood for successful reception of all signals required by the replicated control algorithms compared to systems with centralized or distributed controller designs. We also show that our approach only requires limited effort by control system designers and operators for integration, making it amenable to a variety of use cases. With an adjustable trade-off between minimal actuation interval on the one hand, and the level of controller replication on the network nodes on the other, it further allows to refine the control strategy iteratively, with only minimal changes to both the control algorithms and the communication strategy in each iteration.

We shall note here that the concept of redundancy through controller replication is not new. Indeed, hosting copies of controllers on potentially physically separated entities is a well-known approach in safety-critical systems design, where the provision of "warm" or "hot standby" spares that run in parallel to the primary system are popular options for providing failover guarantees. However, our motivation for the creation of such failover systems comes from a different angle. Redundant components in safety engineering are employed primarily to compensate for component failures once the latter have been properly detected, and it is also assumed that failures occur from active usage of a component, i.e., through wear-out [12]. Thus, the considered failures are of a permanent nature caused by physical stress, which also affects otherwise reliable wired control architectures. In our case, we employ replicated controllers to counteract inherently *transient* failures caused by the intermittent unavailability of wireless communication opportunities between the involved nodes, and thus offer an approach to increase the resilience of networked control systems that is orthogonal to existing approaches.

III. An Anycast-based Wireless Control System

We now present an overview of the control system design aspects that our approach of shared responsibility entails, followed by a description of a prototypical anycast-based wireless communication system that implements our idea. Our design is guided by a set of principles and assumptions, which we describe before.

A. Design Principles and Assumptions

Designing a wireless communication and control system is a highly complex task, as evidenced by the plethora of work that has already been invested into these topics over the last decades. In order to assess the principal viability of our idea, we aim at a *middle ground* in aspects of system complexity and control system/wireless system integration. We hence make the following assumptions and abstractions, which we believe hold for the majority of scenarios due to well-defined administrative oversight and sufficient equipment in industrial control settings:

Network isolation: We assume that all nodes in the wireless network are owned by the same entity and thus willing to cooperate. Security considerations are out of scope of this research paper and should be tackled, e.g., by encrypting/signing frames such that only friendly hosts can decrypt content and insert valid control messages and sensor values. We further assume that the communication channel can be exclusively used by our network, i.e., the plant owner should configure other networks at the same site to use different channels. Like with any reliable wireless communication system we need to assume absence of malicious interferers, which must be guaranteed by other measures.

Clear roles: For the sake of simplicity, we assign each node in the network the logical role of either a sensor, an actuator, or a controller replica. Nevertheless, a physical device can have more than one role; we will then consider it as one logical device per role.

Sufficient energy: We assume that all nodes have sufficient energy resources to participate in the system. Energy-saving approaches are out of scope of our design.

Monolithic controller: While physical entities may be subdivided into several logical ones, each replica of the control algorithm must be able to control the entire plant.

Time-triggered control strategy: To maintain a low coordination overhead, we require all nodes within our network to adhere to a time-triggered communication principle, i.e., nodes never send nor expect information outside slots assigned to them in a clearly-defined schedule. While this does not mean that we require each node to send each time its assigned slot has come, it does mean that we only support round-based and not event-triggered control strategies.

Fixed schedule: In our prototype, we assume that the control and communication schedules are devised in an offline process (cf. Sec. III-B), and that these schedules remain fixed during the operation of the plant.

Leeway in the time domain: As we outline below, a certain controllable variation within the length of the control and communication schedules is our main lever for controlling the shared responsibility between the controller replicas. As a consequence, both the controller and its design method as well as the components of the plant must allow for a certain level of variation in signal delay, which, however, is limited by the length of the round.

One node, one shot: In our prototype, we assume that each logical node has exactly one slot within the schedule for transmitting its own freshly-created sensor signal. A node may transmit for as long as it wishes during that assigned slot and even repeat its signal if time permits, but it cannot assume any further transmission opportunity until the schedule has completed its round. Furthermore, although theoretically possible in our approach, we do not include "slot sharing" strategies such as implicit relaying or information piggybacking [6] in our prototype.

B. Shared Responsibility and Anycast-Aware Controllers

Harnessing the opportunities provided by shared responsibilities and an anycast-based communication regime in controller design is a straightforward process, which engineers can both apply to existing controllers and integrate within the development process of entirely new solutions. Irrespective of the system representation or controller generation method they use, given our assumptions from Sec. III-A, our approach requires an analysis of the plant's properties under the following questions:

(1) *Sensor priority:* "Given a stateless, round-based controller, and the same set of possible sensor inputs in each round, which sensors should be given preference when assuming that only a limited number of signals can reach the controller in time per round?"
(2) *Actuator priority:* "Given (1), which actuation signals from the controller should be given preference when assuming that only a limited number of signals can reach the actuators in time per round?"
(3) *Signal sizes:* "What is the size of each signal (from sensors/to actuators) in the system?"
(4) *Leeway:* "What is the maximum allowable duration for the tightest part of the control loop?"
(5) *Replication number:* "How many nodes in the system can be outfitted with replicas of the controller?"

Answering these questions yields a *superframe* as depicted in Fig. 1 as the overall structure of each communication round. We start each superframe with a short optional beacon for time synchronization, which we describe later in Sec. III-C. We then assign each sensor from (1) a unique slot at the start of the superframe, and similarly guarantee every actuator from (2) at least one actuation slot at the end. Each slot lasts long enough to send each signal as determined by (3), as we also detail in Sec. III-C. Actuator nodes perform or modify their physical manipulations at the end of each superframe.

Depending on the maximum possible duration from (4), we then fill the superframe with an intermediate *completion phase*. In this phase, each controller replica from (5) is given the chance to compensate possibly missed sensor signals by sending out an *inquiry* asking the other replicas to transmit *response*s containing a subset of the missing signals. Depending on the priorities from (1) and (2), each replica may choose to inquire different sensor data in each round. Abiding to our one-shot design principle from Sec. III-A, each replica may be assigned at most one slot for sending inquiries, but the superframe may contain several response

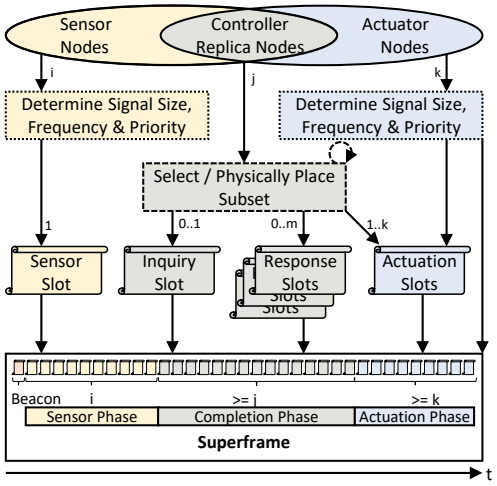

Fig. 1. In our round-based anycast communication scheme, following an optional synchronizing beacon, each sensor (left) is assigned a unique transmission slot. Controller replica nodes (middle) can use an optional "completion phase" to either gather missing sensor signals, or to send actuation signals. Each actuator node (right) is guaranteed at least one actuation slot per round (superframe). Dashed boxes indicate involvement of the control system designer before operation.

slots following an inquiry. The rationale behind allowing only a single inquiry per replica is that it is more important to get a response from any replica than informing every replica about the missing value. Letting other replicas inquire their desired sensor data leverages the inherent path diversity (cf. Sec. II) of our wireless medium much more efficiently.

Once a controller replica has gained sufficient information to generate actuation signals, it sends them out in their assigned actuation slots. The perception of sufficiency may be based on the priorities from (1) and (2) but can, depending on the control process, also go beyond mere priorities. Yet, the decisions must be deterministic to guarantee that the plant behaves the same way irrespective of which replica sends an actuation signal. As several replicas may send out actuation signals during a superframe, these signals can include a *confidence marker* that represents the level of certainty gained from the number of successfully received sensor values. Actuator nodes can use this information to decide whether a control signal arriving later in the superframe originated from a more "confident" controller replica; signals from less confident replicas are then evicted. As actuators wait for the end of each superframe to start their manipulations, this heuristic allows the plant to be controlled at the highest quality of control achievable within the current superframe.

Besides the inherent distribution of signals to multiple interested recipients, the joint efforts aiming at the completion of sensor signals for replicas is a core principle of our approach. Within the confines of the superframe duration as determined by (4), it enables us to effectively trade an increased delay (due to more inquiry/response slots) for an increase in reliability and resilience against transient failures of the wireless channel. Moreover, it can be expected that depending on the priorities (1) and (2), those sensor signals

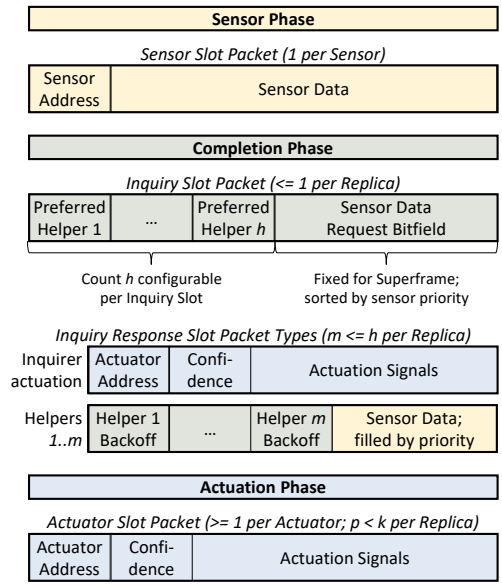

Fig. 2. Packet formats used within our anycast communication scheme differ in each phase. In the completion phase, the currently active replica can choose between sending an inquiry packet or an actuation packet.

of higher importance to the control system are inquired both earlier and more often than lower-priority signals if networking conditions are suboptimal. Consequently, replicas with inquiry/actuation slots later in the superframe have both an increased chance of reception of these more important signals when earlier replicas inquire them, as well as the opportunity to request lower-priority signals in their own slots and thus create actuation signals with increasing confidence.

We next provide details on a prototypical wireless system that can support our idea of shared controller responsibilities.

C. A Wireless Anycast Communication Protocol

The last decades have seen the development of numerous wireless communication systems for control scenarios, especially in industrial settings. The most widely-used [1] standards for the two lowest layers (physical, medium access / logical link control) in these settings are IEEE 802.15.4 (LR-WPAN) and variants of IEEE 802.11 (WiFi). LR-WPAN variants such as ISA100.11a, WirelessHART or Zigbee aim to cater low-rate, low-power scenarios and hence tightly integrate energy saving mechanisms into their operation schemes. While the data rates seen in many current wireless control scenarios are moderate at most [5], especially the integration of additional sensor equipment may quickly lead to requirements which are orders of magnitudes larger [13]. Additionally, it can be assumed that such plants that require very tight and reliably-operated control loops are either wall-powered because of their size and strength (e.g., industrial robots), or at least spend a considerable amount of battery power on their physical manipulators (e.g., drones). Given that many of these systems operate in safety-critical environments, we believe that trading energy usage of the communication system for increased reliability is worth the consideration. We hence base our prototypical system on the 802.11n WiFi standard without dedicated access points, which emphasizes throughput and reliability more than energy efficiency.

Traditional 802.11 networks operate using a carrier sense multiple access/collision avoidance (CSMA/CA) scheme, in which a distributed coordination function (DCF) allows nodes in a network to access the wireless medium if it is sensed idle, and otherwise employs a backoff mechanism without a coordinating entity in which each node contending for the medium evokes randomized, exponentially increasing timers before attempting to send again. To enable our superframe-based mode of operation, we alter the 802.11 DCF function such that the carrier sensing mechanism is removed and a node can always access the medium without waiting time. To further increase the time within each superframe that we can use for the transmission of sensor/actuation data, we also deactivate the 802.11 ready-to-send/clear-to-send (RTS/CTS) mechanism in which the (unicast) sending and receiving stations agree on a specific duration for which the channel shall be used for the next transmission. Our prototype thus employs a 802.11-based communication scheme that retains the channel characteristics as well as the general frame structures but gives us full control over which node sends when under the constraints of the timing characteristics of transceivers as defined by the standard (e.g., the time it takes a node to switch from reception to transmission mode).

To enable true anycast communication in our superframe-based scheme, we make one final adjustment to 802.11n. We leverage the *multicast/broadcast* capabilities of the standard, which allows us to send unacknowledged messages to an unspecified group of recipients. For this, we set the lowest bit of the highest byte of the destination address of all frames we send to 1, using the remainder of the address space for uniquely representing each sensor and actuator signal. However, the standard's requirement to send all such frames with the lowest possible data rate [14] would severely limit our achievable throughput rates. We hence change the rate selection algorithm so that it always calculates the *lowest possible* rate to send a frame for a signal based on point (3) from our list of questions in Sec. III-B. The rates are fixed during superframe construction for sensor, inquiry and actuation signals based on the amount of time the respective slots are assigned in the superframe. For inquiry response slots, the rate may vary depending on the amount of data that is sent by the respective node. As more robust coding schemes yield lower data rates and hence, increased resilience against channel disturbances, we allow nodes in the completion phase to weigh off sending more inquired data against sending less data more reliably.

Given this technical basis, the message exchange between the involved nodes in our system is straightforward. Once the internal clock of a node has determined that one of its assigned slots has started, it encapsulates the generated data in an 802.11 multicast frame and starts transmitting, using the payload packet formats depicted in Fig. 2. Upon reception, each controller replica node determines whether the received signal is of interest to it (e.g., whether it has already received

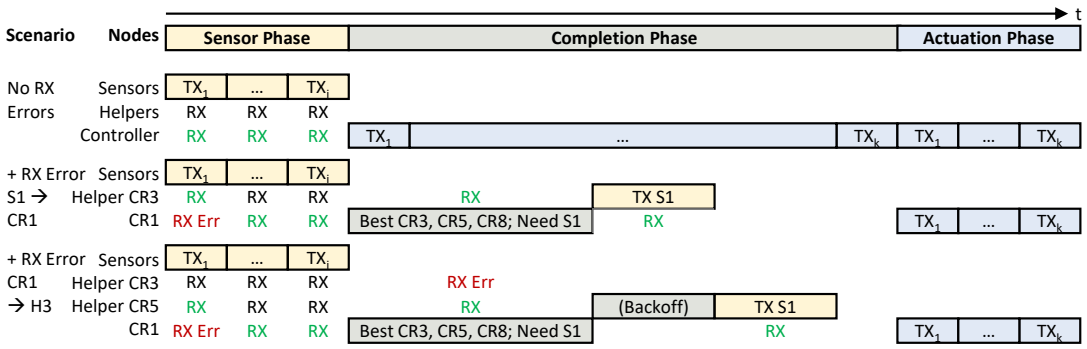

Fig. 3. Three example runs of our cooperative anycast-based control scheme in simplified settings (beacon phase not shown, single inquiry response slot): Ideal case (top), loss of one signal (middle), double signal loss (bottom). Green "RX" contribute to the successful generations of actuation signals.

some inquired sensor information in the superframe) and stores data it deems interesting in an internal buffer that is reset at the beginning of each superframe. Actuator and controller replica nodes evict actuation signals with lower confidence markers upon reception. When one of its assigned actuation slots has come, a replica calculates the required control signals and starts transmitting.

If a replica has been assigned slots within the completion phase of the superframe, it chooses between one of two options in both the inquiry and the related response slots. The first option is to send an actuation signal in the inquiry slot. It does so if the slot's duration permits it given our rate selection algorithm and if the replica deems the actuation signal it could create at this point sufficient. The replica then repeats the transmission of the created actuation signal also in the subsequent response slots. The second option is to send an actual inquiry packet within the inquiry slot. For this, it generates a packet containing a bitfield which, sorted by the priority given by point (1) in Sec. III-B, for each sensor value indicates whether it is interested in receiving that value (1) or not (0). It also generates a list of "preferred helpers" by concatenating the source addresses of a number of controller replicas that it believes can best fulfill its wishes. In our prototype, the helper order is determined by the signal strength indicators (RSSI) of signals received within a fixed number of previous superframes, based on the notion that high RSSI values indicate a good reception from a helper and thus a high chance of receiving eventual responses.

Upon the reception of an inquiry packet, if included within the list of preferred helpers, a controller replica enters a *back-off phase* based on its position within the helper list, with the most preferred helper having the shortest phase. We borrow our backoff times from the IEEE 802.11e standard, which provides a method for (semi-)deterministic access in WiFi networks via eight different priority classes each assigned a specific backoff time, taking into account propagation delays for networks adhering to the generic 802.11 standard. After the backoff period has elapsed, the respective helper performs carrier sensing, i.e., it checks whether the wireless channel is free, and if so, determines whether it has data that would satisfy parts of the received inquiry. If so, it transmits a response packet consisting of a subset of the values requested in the inquiry slot by iteratively including the sensor value with the highest priority that is locally available until the maximum allocated number of bytes to send within the respective slot is reached. Less preferred helpers and all other nodes meanwhile continue listening to the channel, handling response packets as if they were simple concatenations of sensor values. This behavior repeats until the last response slot for the inquiry has finished, with helpers that have sent in a previous response slot yielding to less preferred helpers. By combining the predetermined (and hence, implicit) signal priorities, explicit helper priorities and carrier sensing, this approach allows both the inquiring node and the preferred helpers to select and transmit data within the completion phase in a coordinated fashion with minimal overhead.

If a controller replica receives an actuation signal by a remote replica with a confidence marker higher than the currently achievable local one, the replica can decide to save that signal and re-send it during its completion/actuation phases in lieu of an "own" signal. Thus, even replicas with insufficient information can use their assigned superframe slots to support the propagation of actuation signals.

In order to guarantee that nodes do not miss the beginning of an assigned slot or the end of a backoff period due to clock drift, we allow the transmission of an optional beacon at the beginning of each superframe, which contains the sender's current internal time; nodes receiving the beacon adjust their internal clocks to this value. If a node has not received a beacon for a number of superframes, it temporarily pauses all transmissions until receiving a beacon again, so it does not disturb operation of those nodes that are still synchronized. Beacon transmissions rotate among the active controller replicas to ensure that even nodes not reliably receiving from one specific replica are eventually re-synchronized.

In the following, we present a series of example runs to illustrate the working mechanisms of our approach.

D. Example Runs

We depict the observable packet transmissions (TX) and receptions (RX) within one (simplified) superframe round for three different scenarios in Fig. 3. In the first idealized scenario (top), all signals transmitted during the sensor

phase (yellow) are received by all nodes. Each controller replica (CR) can thus send actuation signals (blue) in both the completion and the actuation phases, maximizing the chance for eventual reception of the signals by the actuator nodes.

In the second scenario (middle), replica CR1 fails to receive the most important sensor value S1 and is unable to generate a sufficient actuation signal at the beginning of the completion phase. It hence sends an inquiry for S1 to the (in this case, three) other replicas it hears best (in order, CR3, CR5, CR7). The most preferred helper CR3 receives the inquiry, can contribute S1 and thus schedules a transmission in the response slot following the inquiry. CR1 receives the answer and can subsequently generate actuation signals in the actuation phase at the end of the superframe.

In the last scenario (bottom), CR3 fails to receive the inquiry by CR1, but CR5 receives it and initiates a backoff phase to not interfere with an eventual transmission attempt by CR3. At the end of the backoff phase, CR5 transmits the data and CR1 receives, yielding a similar situation at the end of the completion phase as in the second scenario.

IV. Evaluation and Future Work

In the following, we first present the results of a preliminary simulative evaluation of our system. We then shed light on current limitations and avenues for future work.

A. Simulative Evaluation

We implement a prototype of our system based on the IEEE 802.11 model of version 4.3.2 of the INET framework [15] for the OMNeT++ simulator (version 6.0pre11) according to our description in the previous section. We then generate the artificial environment shown in Fig. 4 based on a shop floor described in [16], install a Rayleigh Fading model for path loss ($\alpha = 2$), and set background noise to -94 dBm as suggested by measurements in [17]. We distribute ten nodes, each transmitting with a power of 100 mW.

In our scenario, fresh signals from all sensor-equipped nodes (1-8) are required by the controller replicas (1-5) to generate signals for the actuators (9-10) in each round. We establish a communication regime in which each slot is 100 μs long, followed by a guard space of 50 μs to allow for signal propagation. Following the beacon, the sensors send their values of 16 bytes (nodes 1-5), respectively 250, 100 and 50 bytes (nodes 6, 7, 8). Next, the controller replicas can each request up to 350 bytes within the completion phase; we restrict the inquiries to a single potential helper with the highest RSSI. Finally, each controller replica can send an actuation signal of 16 bytes, which is the same for both actuators. Nodes send in order of numeration in each phase, i.e., node 1 always starts, while node 5 has the last chance for both an inquiry and actuation. This regime yields a duration of 3.6 ms for the superframe. Our simulation has a length of 360 seconds, equaling 100,000 superframe iterations.

With the relatively low number of nodes and high crowding, our scenario is very tough in terms of reception quality. Indeed, our simulation logs reveal that in a non-cooperative setting, the control task would fail, as *none* of the replicas

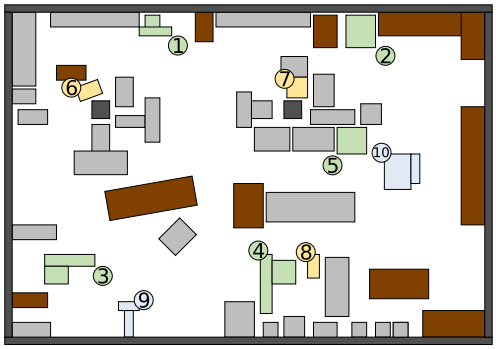

Fig. 4. Evaluation Scenario. 5 nodes with sensors and controller replicas (1-5, green), 3 pure sensors (6-8, yellow) and 2 actuators (9-10, blue) are distributed on a factory floor of approx. 16m x 10m. Line-of-sight is partially obstructed by objects of varying heights and materials (concrete (dark gray), metal (light gray), wood (brown)). Antennas are placed 1.5m above ground.

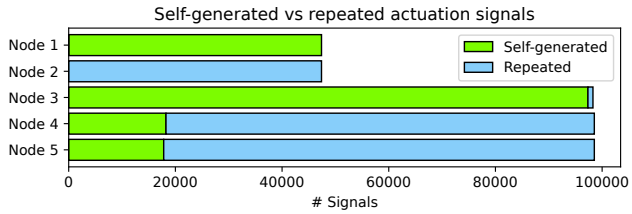

Fig. 5. Overhearing during the actuation phase allows Node 2 to repeat signals generated by Node 1, while Node 3 is able to generate signals itself in the majority of cases; Nodes 4 and 5 profit from all previous slots.

are able to receive the required information on their own in *any* of the superframe iterations. Especially, the very isolated position of node 6 (upper left) causes its signals to exclusively reach node 1, which in turn regularly does not receive from nodes 3 and 5 because of large distance and crowding, respectively. During the completion phase, node 1 is hence regularly asked to redistribute the information from node 6, which it combines with other available information. Similarly, the relatively central and exposed position of node 4 makes it a regular candidate to help out nodes 1, 3 and 5.

This results in the situation that in the majority of cases, as visible in Fig. 5, node 3 is able to successfully create an actuation signal. While nodes 4 and 5, which appear subsequently in the superframe, are not always able to generate signals themselves, they profit from the high generation rate of node 3, whose signals they are able to repeat. The effect of overhearing is even more important for node 2, which cannot generate any signals on its own but can repeat the signal created by node 1 in almost all iterations.

In Fig. 6, we show the observed actuation signal generations and receptions. Within the 100,000 rounds, our actuators receive at least one signal per superframe in 98.5% of the iterations, and 3 or more signals in 98.2% and 53.1% of the cases, respectively. Since the signal paths are uncorrelated, these results underline the ability of our cooperative approach to deliver actuation signals with an increased probability even under harsh conditions, compared to approaches in which only a single node takes responsibility.

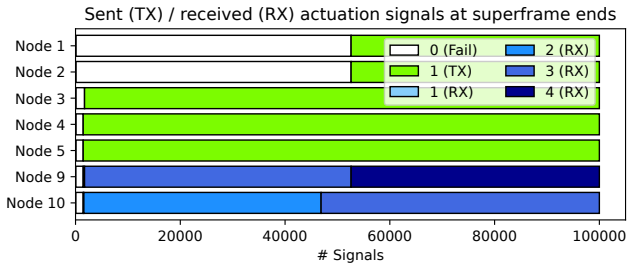

Fig. 6. Our cooperative strategy enables 3 of 5 controller replicas to produce (TX) actuation signals to a significant degree, and actuator nodes to be supplied (RX) with predominantly 2 actuation signals or more.

B. Limitations and Future Work

While our approach increases the chances of signals to reach controller replicas and actuators, it – as *every* wireless system – remains susceptible to the effects of fading and loss.

First, in the form presented in this paper, our system cannot guarantee state- and command consistency, i.e., when controller replicas receive different sets of sensor signals and hence create different actuation signals, actuators might receive conflicting commands. Achieving consensus over unreliable connections such as wireless channels is generally hard, and while recent approaches such as Wireless Paxos [18] appear promising, the question of applicability given the induced overhead and reliance on physical-layer effects remains. The consensus problem may however also be mitigated (although not solved) using means provided by our system, as controller replicas can monitor the actuation signals and confidence markers sent by other replicas. It would thus be an idea to research control algorithms that include special cases for situations in which the majority of witnessed remote confidences is low. Own high-confidence (and thus potentially more aggressive) signals could then be dampened when it is uncertain that the signals successfully spread to all affected actuators. When detecting contradicting commands, replicas could also generate special "correction" signals to propagate within the remainder of the current and potentially later superframes. The trade-off between older / conservative but conflict-resolving versus new but potentially conflict-prolonging signals would be interesting to investigate.

A second type of limitations is connected to the tight and ultimately fixed schedule in our system, which is optimized for the classical control loop and has no notion of communication in the reverse direction of this loop. However, allowing such messaging could enable actuators to report on the reception qualities from different replicas, which can be leveraged to further mitigate the inconsistency issue mentioned above. Also, while based on well-proven standards, our backoff mechanism in the completion phase might experience issues related to the *hidden node problem* when short slot- and guard times are required and the requested helpers are close to the inquiring node, but far enough apart to not reliably hear eventual other responses that are being transmitted, which may lead to signal collisions at the inquiring node. We are hence investigating less rigid communication schemes in an effort to trade slight prolongations of the superframe in favor of "maintenance communication" for on-line reconfiguration.

V. Conclusion

In this paper, we evaluate the applicability of cooperative anycast communication to increase resilience against channel disruptions in critical wireless industrial control scenarios. Our core idea is to leverage the broadcast nature of the wireless medium and the inherent unselfish interest of nodes in the control process to build a novel system in which responsibilities for control are shared concurrently among the nodes while maintaining a low coordination overhead.

To this end, we first formulate a set of questions that guides the integration of our approach into existing and emerging control scenarios. We then develop a prototypical communication protocol implementing our idea based on the well-established IEEE 802.11n standard. Via simulation, we show that our approach is able to close an artificial control loop under conditions that would cause systems without shared responsibilities among the nodes to fail.

We believe that our idea provides a promising perspective for wireless control and communication systems design orthogonal to existing work, and hope to soon test our approach and suggested future work in a real-life control scenario.

References

[1] P. Park *et al.*, "Wireless Network Design for Control Systems: A Survey," *IEEE Comm. Surv. Tut.*, vol. 20, no. 2, pp. 978–1013, 2018.
[2] S. Hirche *et al.*, "Performance Oriented Control over Networks – Switching Controllers and Switched Time Delay –," in *IEEE 45th Conference on Decision and Control*, 2006, pp. 4999–5005.
[3] S. Gallenmüller *et al.*, "Enabling Wireless Network Support for Gain Scheduled Control," in *EdgeSys*, 2019, pp. 36–41.
[4] J. Chen *et al.*, "Distributed Collaborative Control for Industrial Automation With Wireless Sensor and Actuator Networks," *IEEE Trans. Ind. Electron.*, vol. 57, no. 12, pp. 4219–4230, 2010.
[5] A. Frotzscher *et al.*, "Requirements and current solutions of wireless communication in industrial automation," in *IEEE International Conference on Communications Workshops (ICC)*, 2014, pp. 67–72.
[6] M. Serror *et al.*, "QWIN: Facilitating QoS in Wireless Industrial Networks Through Cooperation," in *IFIP Netw.*, 2020, pp. 386–394.
[7] M. Rentschler and P. Laukemann, "Performance analysis of parallel redundant WLAN," in *IEEE ETFA*, 2012.
[8] M. Serror *et al.*, "Practical Evaluation of Cooperative Communication for Ultra-Reliability and Low-Latency," in *IEEE WoWMoM*, 2018.
[9] O. Landsiedel *et al.*, "Chaos: Versatile and Efficient All-to-All Data Sharing and in-Network Processing at Scale," in *ACM SenSys*, 2013.
[10] M. Zimmerling *et al.*, "Synchronous Transmissions in Low-Power Wireless: A Survey of Communication Protocols and Network Services," *ACM Computing Surveys*, vol. 53, no. 6, 2020.
[11] V. Narasimha Swamy *et al.*, "Real-Time Cooperative Communication for Automation Over Wireless," *IEEE Transactions on Wireless Communications*, vol. 16, no. 11, pp. 7168–7183, 2017.
[12] M. Bozzano and A. Villafiorita, *Design and Safety Assessment of Critical Systems*. Boca Raton, FL, USA: CRC Press, 2010.
[13] R. Glebke *et al.*, "A Case for Integrated Data Processing in Large-Scale Cyber-Physical Systems," in *HICSS*, 2019, pp. 7252–7261.
[14] F. Wu *et al.*, "Multicast Rate Adaptation in WLAN via NDN," in *ICCCN*, 2018.
[15] INET / OMNeT++ Contributors, "INET Framework," 2022. [Online]. Available: https://inet.omnetpp.org/
[16] L. Tang *et al.*, "Channel Characterization and Link Quality Assessment of IEEE 802.15.4-Compliant Radio for Factory Environments," *IEEE Transactions on Industrial Informatics*, vol. 3, no. 2, pp. 99–110, 2007.
[17] B. Fu *et al.*, "Wireless Background Noise in the Wi-Fi Spectrum," in *WiCom*, 2008.
[18] V. Poirot *et al.*, "Paxos Made Wireless: Consensus in the Air," in *EWSN*, 2019, pp. 1–12.