IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 11 October 2022, accepted 26 October 2022, date of publication 31 October 2022, date of current version 8 November 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3218410

RESEARCH ARTICLE

Semantics- and Task-Oriented Scheduling for Networked Control Systems in Practice

ONUR AYAN, (Graduate Student Member, IEEE), POLINA KUTSEVOL, (Graduate Student Member, IEEE), HASAN YAĞIZ ÖZKAN, AND WOLFGANG KELLERER, (Senior Member, IEEE)
Chair of Communication Networks, Technical University of Munich, 80333 Munich, Germany

Corresponding author: Onur Ayan (onur.ayan@tum.de)

This work was supported in part by the Deutsche Forschungsgemeinschaft (DFG) through German Research Foundation under Project 315177489.

ABSTRACT Networked control systems (NCSs) are feedback control loops closed over a communication network. Emerging applications, such as telerobotics, drones, and autonomous driving, are the most prominent examples of such systems. Regular and timely information sharing between the components of NCSs is essential to fulfilling the desired control tasks, as stale information can lead to performance degradation or even physical damage. In this work, we consider multiple heterogeneous NCSs that transmit their system state over a shared physical wireless channel to a gateway node. We conduct a comprehensive experimental study on selected MAC protocols using software-defined radios with state-of-the-art (SotA) solutions designed to increase information freshness and control performance. As a significant improvement over the SotA, we propose a novel contention-free algorithm that is able to outperform the existing solutions by combining their strengths in one protocol. In addition, we propose a new metric called normalized mean squared error that maps the age of information to a dimensionless quantity that captures the expected value of a control system's next transmission. We demonstrate its adoption and effectiveness for wireless resource scheduling in a case study involving multiple inverted pendulums. From our experimental study and results, we observe that value-aware prioritization of the sub-systems contributes to minimizing the adverse effects of information staleness on control performance. In particular, as the number of devices increases, the benefit of control awareness to the quality of control stands out when compared to protocols that focus solely on maximizing information freshness.

INDEX TERMS Age of information, networked control systems, semantics of information, task-oriented communications.

I. INTRODUCTION

6G wireless systems are envisioned to be a disruptive generation of cellular networks whose design is tailored to the performance requirements of the supported applications [1]. In particular, connected robotics and autonomous systems are one of the key driving application domains in 6G wireless systems. Emerging applications such as autonomous cars, autonomous robotics, and drone-delivery systems are the most prominent examples of connected robotics. Since such systems rely on regular and timely information sharing between multiple sensors, actuators, and controllers, the co-existence of such applications in a network requires a careful end-to-end co-design of communication, control, and computing.

From a system theoretic perspective, connected robotics applications can be classified as *networked control systems (NCSs)*, i.e., feedback control loops that are closed over a communication network. In contrast to conventional control theory, NCSs accommodate at least one link in the feedback loop that is not ideal, hence directly affecting the *quality of control (QoC)*. In particular, as the scarcity of network resources increases, e.g., due to a large number of users in the network, those components located at the other end of the imperfect communication link may retain outdated information due to delays and packet loss. As a result, the

The associate editor coordinating the review of this manuscript and approving it for publication was Kuo-Ching Ying.

control performance degradation is inevitable, which may lead to the destabilization of the system, physical damage to the environment, and even injuries to human operators.

One of the main differences between NCSs and traditional communication systems is that their information exchange takes place to complete a particular task in the physical environment. For example, the communication between the sensors of an autonomous car and its electronic control unit aims to prevent potential accidents or improve the driving experience. In such an environment, the overall system performance cannot be measured by the number of bits correctly transmitted over a noisy channel, as it is mostly done in today's communication networks, but rather by the efficiency of achieving the underlying communication purpose [2]. This calls for a careful reconsideration of the existing networking infrastructure and the protocol stack, as the network design cannot be further considered agnostic to the meaning behind the transmitted information.

An effort towards this vision has been made through the introduction of the metric *age of information (AoI)* [3]. The AoI is a metric quantifying information freshness in a real-time monitoring scenario and is defined as the time that has elapsed since the generation of the most recent information at the monitor. In other words, a piece of information is said to be *fresh* when the time difference between current time and its generation time is relatively small. Moreover, the value in transmitting a packet has been measured by its contribution to the reduction in AoI upon its reception by the monitor.

The ability of AoI, which is an application layer metric by definition, to combine end-to-end delay and packet loss into one metric makes it appealing to time-sensitive applications and cross-layer network design. However, the AoI lacks the ability to capture the value of information beyond its timeliness aspect, as it is independent of the characteristics of the application, e.g., whether the communication takes place to control an autonomous car or a room temperature. Therefore, one could intuitively argue that having the same AoI does not necessarily indicate the same level of urgency or importance of providing the destination with a new update. Although receiving fresh information at the monitor is generally desired, the value of an update should not only be measured by the level of its timeliness but also by its task-specific value.

The value of a piece of information beyond its freshness is addressed by the emerging notion of *semantics of information (SoI)*, defined as the significance of data relative to their transmission purpose in [4]. SoI have drawn great attention by the research community and is envisioned to play a vital role in future networked systems, especially in task-oriented communication systems. SoI become particularly relevant in scenarios with limited resources, e.g., for wirelessly connected NCSs, where multiple devices share the same channel. As the network resource scarcity increases with the growing number of connected machines, it becomes crucial to identify and prioritize the most relevant information to efficiently utilize the resources for improved control performance. This particular challenge is envisioned to be addressed already in the 6G cellular networks by including the significance and effectiveness aspects of information in the network design [5].

Adopting an SoI- and task-oriented approach in decision-making introduces new challenges to the network and protocol design in practice. Suppose a wireless network shared by multiple control applications, in which the medium access is managed by a centralized scheduler. In such a setting, having the control performance improvement as the primary goal in mind, the scheduler does not only need to measure the channel quality or keep track of the offered throughput, but also it needs to be provided with a certain application-specific real-time evaluation mechanism to prioritize the "most urgent" transmission. However, the imperfect and random nature of the wireless channel, approximation and hardware effects, as well as timing errors, may potentially conceal the performance gain promised by the theoretical research. Hence, the additional effort spent on the protocol design may be completely superfluous.

A. MAIN CONTRIBUTIONS

In a preliminary version of this work, we have shown that if the control-task-specific value of packets are considered for the resource allocation, a higher control performance is achieved when compared to an entirely freshness-based strategy [6]. Including but not limited to [6], the vast majority of the existing theoretical research papers studying the centralized scheduling problem based on AoI and beyond relies on various assumptions ranging from perfect slot synchronization to the exact knowledge of the success probability of an upcoming transmission [7]. On the contrary, this work is free from such assumptions and is entirely based on real-world measurements.

In this work, we extend our preliminary theoretical results through an extensive experimental study focusing on medium access control (MAC) protocols for NCSs. To that end, in a network comprising multiple feedback control loops of heterogeneous type, we study the relationship between various resource allocation mechanisms and the resulting control performance. Particularly, the core question that this work is centered around is: *Despite all the aforementioned challenges and non-idealities introduced by the real network, is it still possible to achieve a notable QoC improvement in practice through SoI-based scheduling policies that heavily rely on theoretical and control-specific system modeling?*

In order to answer this question, we have set up a network comprised of up to 15 real-time feedback control loops closed over a shared single-hop wireless link. The information exchange between source-destination pairs has been implemented using software-defined radios (SDRs).[1] By leveraging our theoretical results from [6], we implement an enhanced version of our previously proposed SoI-aware

[1]The SDRs have been programmed in C++ using the GNU Radio software radio framework [8].

centralized scheduling policy and compare the resulting key performance indicators (KPIs) such as the mean AoI, mean squared estimation error, and linear-quadratic-Gaussian (LQG) cost, a well-established KPI metric from the control theory.

The operation of the proposed scheduler is based on the polling principle, i.e., the most urgent transmission is triggered by a scheduling grant message centrally. The scheduling decisions are made in a control- and channel-aware manner. That is, a scheduling decision depends on the estimated link reliability, as well as the growing uncertainty at the receiver in the absence of fresh information. To obtain the uncertainty with growing AoI, the dynamics of control applications are taken into account; hence the task-oriented decision-making is enabled through the consideration of SoI. To the best of our knowledge, this is the first work implementing a wireless scheduling policy on real hardware that considers the control dynamics or SoI beyond information freshness.

Our experimental study consists of the comparison of multiple contention-based and contention-free MAC protocols. We show that our proposed solution is able to outperform its closest competitor, which is an AoI-aware polling-based protocol agnostic to the value of a packet beyond its freshness, by up to 21% concerning control performance. This reveals that in task-oriented communications, the network design should not adopt a task-agnostic approach neglecting the application and task-specific SoI. On the contrary, the consideration of the value and meaning behind the transmitted bits play a significant role if improving applications' performance is at the forefront.

The remainder of this paper is outlined as follows. Section II overviews the related literature in two main blocks, i.e., theoretical and systems research. Section III introduces the considered theoretical control and network model, on which our experimental setup is based. Section IV elaborates on the adverse effects in the control, i.e., application layer, arising from the network-induced information staleness. In section V, we give an overview of the implemented MAC protocols from the SotA that are used for benchmarking. Additionally, in the same section, we introduce our proposed polling-based scheduling policy. Section VI presents the most relevant details of our practical setup and implementation. Moreover, section VII evaluates the results and key findings. Finally, section VIII concludes the paper.

II. RELATED WORK

Theoretical Research: There are extensive theoretical research papers on minimizing AoI in various contexts. Works [7], [9], [10], [11] focus on finding age-optimal scheduling policies in single-hop networks, whereas [12], [13], [14], [15] consider multi-hop topology. Moreover, [16], [17], [18] derive AoI performance when different queuing disciplines are applied. They show that replacing outdated packets with newly generated ones in the transmission queue is beneficial w.r.t. information freshness.

Adoption of properties of information as in the case of AoI has given rise to the adoption of metrics beyond AoI such as non-linear age, ''value of information'' (VoI), ''age of incorrect information'' and ''deviation of information'' [19], [20], [21], [22], [23], [24] in single user scenarios. The notion of non-linear aging has been refined for control applications by using system-dependent parameters [6]. In their work, the authors define the VoI as a function of control system parameters and employ it for the wireless resource allocation problem. They show that it outperforms the AoI w.r.t. control performance, although the network-wide information freshness is decreased. Moreover, [25] focuses on improving control performance for queuing systems in which an optimal distribution of the total available service rate is found. [26] is another example of the adoption of AoI-based functions in NCSs domain. It suggests an optimal sampling policy for a single-user scenario that minimizes AoI-based functions. However, [6], [25], [26] assume either zero end-to-end delay or base their evaluation on constant service rate. Additionally, [27] studies the centralized scheduling for NCSs. It proposes a greedy scheduling protocol based on the estimation error caused by the communication network. However, they assume global knowledge of the instantaneous error at the scheduler, which is not a feasible assumption regarding the proposed algorithm's practical feasibility. All the works mentioned above provide valuable insights into AoI and NCS domains. However, they do not capture important system-related complications of practical deployment.

Systems Research: The vast majority of previous work on AoI and NCSs, including but not limited to the papers mentioned in the previous subsection, has been theoretical. One of the main reasons for this has been the significance of the demanded effort for modifying the communication stack, which has hindered the validation of proposed solutions on hardware.

[28], [29], [30] are the first examples of practical AoI research, which measure AoI performance in real-life connections without any modifications in the communication stack. [31] is the first work known to us towards AoI-aware customization in which the authors propose a transport layer protocol for increased information freshness. However, their implementation is limited to rate control, and lower layers are transparent to the source and thus are left unmodified.

The recent increasing popularity of softwarization in networking, especially the introduction of SDR platforms in wireless research, has lowered the barrier to go beyond transport layer customization on real hardware. [32], [33], [34], [35] propose customized solutions using SDRs for improved network-wide information freshness. To the best of our knowledge, these are the only AoI research papers that propose a customized MAC layer solution using real-world equipment. In [33], the authors propose an AoI-threshold-based random access protocol for wireless networks that reduces the mean AoI compared to the well-known slotted ALOHA protocol. [32] proposes a contention-free wireless MAC protocol implemented on SDR platforms and

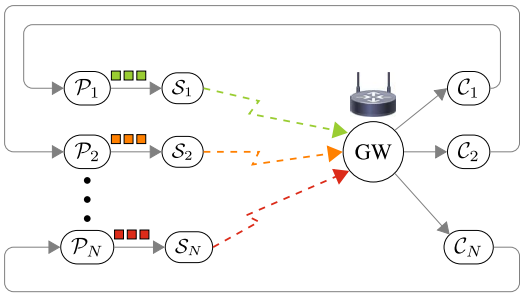

FIGURE 1. The considered scenario with N feedback control loops closed over a shared wireless link. Each SDR $\mathcal{S}_i$ is responsible for transmitting status update packets of plant $\mathcal{P}_i$ to the gateway (GW) node from where it is forwarded to the corresponding controller $\mathcal{C}_i$. Solid arrows represent ideal links between components of a feedback loop.

shows that their framework outperforms a standard WiFi network w.r.t. information freshness. The authors of [35] develop an SDR testbed with single transmitter and multiple receivers and compare age performance of conventional and age-minimizing MAC scheduling policies for push- and pull-based communication scenarios. [34] considers multiple NCSs sharing a wireless communication network. It compares the network-wide AoI and control performance when different queuing disciplines such as last come first serve (LCFS) and first come first serve (FCFS) are employed. The authors conclude that in their considered scenario, the LCFS discipline performs significantly better as the resource scarcity of the network increases. However, they assume control systems of homogeneous type, and their work is limited to a performance comparison between different queuing disciplines. In contrast to our work, it does not suggest any customized task-oriented MAC protocol for improving the AoI or control performance.

A. NOTATION

$\mathbb{N}_0$ denotes the natural numbers including zero. The positive natural numbers are denoted by $\mathbb{N}^+$. Throughout the paper, matrices are denoted by capital letters in bold font, i.e., $\boldsymbol{M}$, whereas small letters are used for vectors, i.e., $\boldsymbol{v}$. Transpose of a matrix $\boldsymbol{M}$ is given as $\boldsymbol{M}^T$. Moreover, $\boldsymbol{M}^p$ is the p-th power of a matrix $\boldsymbol{M}$.

III. SCENARIO AND BACKGROUND ON REMOTE CONTROL

A. NETWORK

We consider N heterogeneous feedback control loops closed over a shared wireless channel. Each loop consists of a plant and a controller. The plant $\mathcal{P}_i$ of the i-th sub-system is the entity that is to be controlled by the controller $\mathcal{C}_i$, whereas the controller's goal is to drive $\mathcal{P}_i$ to a desired state.

The controller is able to observe the plant state via the shared wireless channel, where each packet containing a single status update is transmitted by an SDR $\mathcal{S}_i$ to a gateway (GW) from where it is forwarded to $\mathcal{C}_i$. In this work, we consider the controller-to-plant link to be ideal. The camera-based control of an inverted pendulum can be named as a practical example of such a topology.[2] While the camera observes the system remotely and transmits real-time state measurements over a wireless network, the plant and the controller are co-located. As a result, the topology can be viewed as N source nodes contending for channel resources to transmit their status updates over the shared single-hop wireless communication link. Fig. 1 depicts the resulting network topology that is considered throughout the following sections.

From the theoretical AoI research [16], we know that under the assumption that the status is Markovian, having received an update, the controller does not benefit from receiving an older observation. Thus, older packets are considered as obsolete and non-informative. Consequently, in our framework and in the following analysis, we assume that each $\mathcal{S}_i$ discards any older packet upon generating new state information. This queuing discipline is referred to as LCFS in the literature and has been proven to outperform practical implementations employing FCFS [32], [34].

B. CONTROL

In our setup, we employ digital representation of control sub-systems running as independent processes parallel to their communication counterparts, i.e., SDRs. The behavior of the control loops is modeled as discrete-time linear time-invariant (LTI) systems. That is, the system state of $\mathcal{P}_i$ varies over time in discrete steps with a constant period of $T_{i,s}$. In other words, two consecutive discrete time steps t and $t+1$ are $T_{i,s}$ seconds apart in continuous time.

The system state of each sub-system i follows the state-space representation:

$$\boldsymbol{x}_i[t+1] = \boldsymbol{A}_i\boldsymbol{x}_i[t] + \boldsymbol{B}_i\boldsymbol{u}_i[t] + \boldsymbol{w}_i[t]. \tag{1}$$

Here, $\boldsymbol{x}_i \in \mathbb{R}^{n_i}$ and $\boldsymbol{u}_i \in \mathbb{R}^{m_i}$ are column vectors denoting the plant state and control input, respectively. $\boldsymbol{A}_i \in \mathbb{R}^{n_i \times n_i}$ is the time-invariant system matrix, which defines the relationship between the current system state $\boldsymbol{x}_i[t]$ and the next state $\boldsymbol{x}_i[t+1]$. Moreover, $\boldsymbol{B}_i \in \mathbb{R}^{n_i \times m_i}$ is the time-invariant input matrix, which defines the effect of the control input on the next system state. The noise vector $\boldsymbol{w}_i \in \mathbb{R}^{n_i}$ is considered to be independent and identically distributed (i.i.d.) according to a zero-mean Gaussian distribution with diagonal covariance matrix $\boldsymbol{\Sigma}_i \in \mathbb{R}^{n_i \times n_i}$, i.e., $\boldsymbol{w}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_i)$.

It is important to emphasize that every $T_{i,s}$ seconds, the system state $\boldsymbol{x}_i[t]$ is updated according to (1) from $\boldsymbol{x}_i[t]$ to $\boldsymbol{x}_i[t+1]$. The state is considered constant between the two consecutive update instances, also referred to as *sampling* instances in the literature. However, we assume that $T_{i,s}$, i.e., the *sampling period* is selected small enough such that (1) sufficiently approximates the continuous time behavior of the

[2] An inverted pendulum is explained in Sec. VII-C in detail. Fig. 13 depicts an inverted pendulum.

real control system. This is a well-established approach in control theory textbooks to model continuous-time systems in discrete time [36]. In order to simplify the following analysis and the protocol design, we have selected the sampling period to be equal among all control sub-systems, i.e., $T_{i,s} = T_s,\ \forall i$.

The input signal $\boldsymbol{u}_i[t]$ is calculated based on the observation history available at $\mathcal{C}_i$. However, as the state information is delivered via the shared wireless $\mathcal{S}_i$-to-$\mathcal{C}_i$ link, packet collisions and the wireless resource scarcity lead to the fact that only a subset of the generated packets at each source is successfully delivered to $\mathcal{C}_i$. In addition, those are delivered with a non-negligible end-to-end delay caused by the communication stack between the plant and controller processes. As a result, the information at the controller is outdated, and the network-induced information staleness at $\mathcal{C}_i$ leads to inaccurate control inputs, degrading the overall control performance.

To reduce the adverse effects of information staleness, each controller $\mathcal{C}_i$ estimates the current state $\boldsymbol{x}_i[t]$ remotely. In particular, given the freshest information $\boldsymbol{x}_i[\nu_i(t)]$ generated at sampling period $\nu_i(t)$ and received by $\mathcal{C}_i$ until the beginning of the t-th sampling period, the controller of sub-system i estimates the current state based on its expected value as:

$$\begin{aligned}\hat{\boldsymbol{x}}_i[t] &\triangleq \mathbb{E}\left[\boldsymbol{x}_i[t] \mid \boldsymbol{x}_i[\nu_i(t)]\right] \\ &= \boldsymbol{A}_i^{\Delta_i[t]}\boldsymbol{x}_i[\nu_i(t)] + \sum_{q=1}^{\Delta_i[t]} \boldsymbol{A}_i^{q-1}\boldsymbol{B}_i\boldsymbol{u}_i[t-q], \end{aligned} \tag{2}$$

with $\Delta_i[t] \triangleq t - \nu_i(t), \quad t, \nu_i(t) \in \mathbb{N}_0,\ \forall i$. The model of the remote state estimation is taken from [25], in which the authors provide the proof of (2). Similar to their work, $\Delta_i[t]$ is defined as the number of elapsed sampling periods since the generation of the freshest state information available at $\mathcal{C}_i$, thus *age of information (AoI)* in the unit of T_s. The age model is discussed in detail later in Sec. IV-A.

We assume that the controller design is done independently of the network and prior to the deployment of control loops. Therefore, we select the commonly used *linear quadratic regulator (LQR)* for controller design that aims to minimize the infinite horizon quadratic cost function:

$$F_i \triangleq \limsup_{T\to\infty} \mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1}(\boldsymbol{x}_i[t])^T\boldsymbol{Q}_i\boldsymbol{x}_i[t] + (\boldsymbol{u}_i[t])^T\boldsymbol{R}_i\boldsymbol{u}_i[t]\right]. \tag{3}$$

The matrices $\boldsymbol{Q}_i$ and $\boldsymbol{R}_i$ are symmetric positive semi-definite weighting matrices of appropriate dimensions that are used to penalize the state error and control effort, respectively. Throughout the paper, we assume the set-point to be zero. Therefore, the state $\boldsymbol{x}_i$ is essentially its deviation from the desired value. In control theory textbooks, F_i is referred to as the *linear-quadratic-Gaussian (LQG)* cost function.

The controller obtains the control input by following a linear, time-invariant *control law* [36]:

$$\boldsymbol{u}_i[t] = -\boldsymbol{L^*}_i\hat{\boldsymbol{x}}_i[t], \tag{4}$$

where $\boldsymbol{L^*}_i \in \mathbb{R}^{m_i\times n_i}$ is the optimal state feedback gain matrix. The calculation of $\boldsymbol{L^*}_i$ follows by solving the discrete time algebraic Riccati equation:

$$\boldsymbol{P}_i = \boldsymbol{Q}_i + \boldsymbol{A}_i^T\left(\boldsymbol{P}_i - \boldsymbol{P}_i\boldsymbol{B}_i\left(\boldsymbol{R}_i + \boldsymbol{B}_i^T\boldsymbol{P}_i\boldsymbol{B}_i\right)^{-1}\boldsymbol{B}_i^T\boldsymbol{P}_i\right)\boldsymbol{A}_i, \tag{5}$$

with the solution:

$$\boldsymbol{L^*}_i = \left(\boldsymbol{R}_i + \boldsymbol{B}_i^T\boldsymbol{P}_i\boldsymbol{B}_i\right)^{-1}\boldsymbol{B_i}^T\boldsymbol{P}_i\boldsymbol{A}_i. \tag{6}$$

In simple words, the operation of the controller can be summarized as follows: After each estimation step performed according to (2), the controller uses $\hat{\boldsymbol{x}}_i[t]$ to determine the control input following the control law in (4). The resulting $\boldsymbol{u}_i[t]$ is then applied to $\mathcal{P}_i$ during the next sampling period t. The freshest packet that has been received until the end of the t-th sampling period is then used for the estimation of $\boldsymbol{x}_i[t+1]$, and the following control input is obtained analogously.

We would like to mention that $\boldsymbol{L^*}_i$ is the optimal matrix minimizing the LQG cost F_i without the consideration of the network. However, as the authors of [37] show in Corollary 1, the controller with the conditional state estimation as in (2) and the optimal feedback matrix $\boldsymbol{L^*}_i$ obtained by solving the standard LQG problem leads to the optimal control law as in (4) if the network is prone to delays and dropouts. The effects of the network imperfections are reflected in the estimation process.

IV. INFORMATION STALENESS AND EFFECTS ON CONTROL PERFORMANCE

A. AGE OF INFORMATION

As described in Sec. III, each controller obtains a remote state estimate based on the freshest information available. However, especially in a real network, as in our considered scenario, it is common to observe network-induced delays originating from processing and transmission. In addition, part of the generated data is either discarded in the transmission queue or "lost" in the channel due to bad link quality or simultaneous access. These combined effects lead to information staleness and, consequently, inaccurate state estimation. In that case, the controller's actions become suboptimal, which leads to increased state deviation from the equilibrium. This makes the controller put more effort into driving the state back to the desired value. As a result, the control cost F_i given in (3), which is characterized jointly by the state error and the control effort, increases.

As in (2), let $\boldsymbol{x}_i[\nu_i(t)]$ be the most recent information available at $\mathcal{C}_i$ that denotes the system state at $\nu_i(t)$, where $\nu_i(t) < t$ holds.[3] From Sec. III we know that the state of our plant process only changes with discrete and constant intervals over time. Therefore, since our goal is to quantify the age of the freshest information, we are interested in the

[3] $\nu_i(t)$ is always smaller than t because in our implementation, the calculation of $u_i[t]$ happens directly subsequent to sampling. As it is infeasible to have "almost zero" delay in a practical setup, in our mathematical model, we do not allow the equality case, i.e., $\nu_i(t) < t,\ \forall t$.

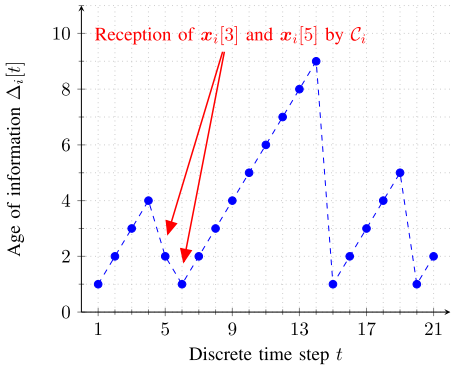

FIGURE 2. Example evolution of discrete time AoI of a sub-system i, i.e., $\Delta_i[t]$ recorded during a real-world measurement using our experimental platform. We observe four reduction of AoI, namely at $t = \{5, 6, 15, 20\}$. Note that the dashed line does not represent the evolution of AoI in continuous time.

difference between the current time step t and the generation time step $\nu_i(t)$, i.e., AoI, in units of T_s.

For better understanding, we present the evolution of AoI during a 20 sampling periods long measurement. Fig. 2 shows the AoI at one of the controllers monitoring the state of its respective plant via a shared channel. The figure plots $\Delta_i[t]$ over $t = \{1, 2, \ldots, 21\}$ where the initial AoI is one, i.e., $\Delta_i[1] = 1$ or $\nu_i(1) = 0$. We observe that AoI drops four times during our measurement, at $t = \{5, 6, 15, 20\}$ due to a successful update. First, we see a linear increase up to $\Delta_i[4] = 4$ with a slope of 1, indicating no new reception prior to the beginning of the fourth sampling period, i.e., $\nu_i(t) = 0$ for $t \leq 4$. Until $\mathcal{C}_i$ begins with the calculation of $\boldsymbol{u}_i[5]$, one or more packets have been successfully decoded by the controller, with the freshest packet containing the system state $\boldsymbol{x}_i[3]$. In other words, two sampling periods have elapsed until the new update has successfully been used by $\mathcal{C}_i$ when $\boldsymbol{u}_i[5]$ is obtained and AoI drops to 2. During the next sampling period, $\mathcal{C}_i$ receives $\boldsymbol{x}_i[5]$ which leads to $\nu_i(6) = 5$. Similarly, it follows that $\nu_i(t) = 5$ for $6 \leq t \leq 14$ and $\nu_i(t) = 14$ for $15 \leq t \leq 19$. It is important to emphasize that the dashed line connecting the round markers at discrete time steps of t does not represent the continuous-time behavior of $\Delta_i[t]$ as it is only defined at sampling instances.

B. ESTIMATION ERROR

Let us consider two real-time processes that are sampled with the same sampling frequency of $T_s = 10$ milliseconds, e.g., the temperature of an office room which may not vary a lot over long time periods and the location of an unmanned aerial vehicle (UAV) that is highly mobile. Moreover, suppose that we are monitoring the states of these two plants via a communication network, and we can transmit the latest system state only once in every 1000 packets, i.e., once per 10 seconds.

Suppose we are dealing with different classes of applications as in this toy example. In that case, we can intuitively see that the AoI cannot capture the uncertainty growing at the monitor over time between two consecutive status updates. In other words, the value of transmitting the next packet when the AoI reaches 1000 differs for the considered applications as they are unlike in state dynamics.

One way of capturing the uncertainty at the destination that monitors heterogeneous sources is to use the estimation error. The *estimation error* is defined as the difference between the actual system state and the estimated system state, i.e.:

$$\begin{aligned} \boldsymbol{e}_i[t] &\triangleq \boldsymbol{x}_i[t] - \hat{\boldsymbol{x}}_i[t] \\ &= \sum_{d=1}^{\Delta_i[t]} \boldsymbol{A}_i^{d-1} \boldsymbol{w}_i[t-d]. \end{aligned} \tag{7}$$

The closed form equation for $\boldsymbol{e}_i[t]$ can be obtained by subtracting (2) from (1). The mean squared error (MSE), which can be derived from the estimation error, is widely used in the literature to quantify estimation performance. It can be obtained by taking the expectation of a quadratic form as:

$$MSE_i[t] \triangleq \mathbb{E}\left[(\boldsymbol{e}_i[t])^T \boldsymbol{e}_i[t]\right]. \tag{8}$$

In [6], we derive the MSE as a function of AoI for the same model of an NCS as in this work:

$$MSE_i[t] = \sum_{d=1}^{\Delta_i[t]} \mathsf{tr}\left((\boldsymbol{A}_i^T)^{d-1} \boldsymbol{A}_i^{d-1} \boldsymbol{\Sigma}_i\right), \tag{9}$$

with the trace operator $\mathsf{tr}(.)$. Here, $\boldsymbol{A}_i$, $\boldsymbol{\Sigma}_i$ are defined as in (1). (9) maps the instantaneous AoI $\Delta_i[t]$ to MSE that strongly depends on control system parameters such as the system matrix and noise covariance matrix. Note that these parameters are time-invariant and $\Delta_i[t]$ is the only time-dependent variable in the equation.

$\boldsymbol{e}_i[t] \in \mathbb{R}^{n_i}$ is a multi-variate random variable (RV) defined as the deviation of the system state from its expectation. The first property of $\boldsymbol{e}_i[t]$ is that it is a zero-mean multivariate RV, i.e., $\mathbb{E}[\boldsymbol{e}_i[t]] = \boldsymbol{0}$, with $\boldsymbol{0}$ being a column vector of length n_i that contains only zeros. This can easily be shown by taking the expectation of the right hand side (RHS) of (7) and applying the linearity property of expectation. Moreover, $\boldsymbol{e}_i[t]$ is a normally distributed multi-variate RV since each addend in (7) is a linear transformation of the multivariate normal RV $\boldsymbol{w}_i[t-d] \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_i)$ with $1 \leq d \leq \Delta_i[t]$. In fact, each addend follows a normal distribution with the covariance matrix $\boldsymbol{\Sigma}_d = \boldsymbol{A}_i^{d-1} \boldsymbol{\Sigma}_i (\boldsymbol{A}_i^{d-1})^T$.

Proof: Given any $d \geq 1$, the d-th addend of (7) is $\boldsymbol{y}_d[t] = \boldsymbol{A}_i^{d-1} \boldsymbol{w}_i[t-d]$ with $\boldsymbol{y}_d[t] \in \mathbb{R}^{n_i}$ and $\mathbb{E}\left[\boldsymbol{y}_d[t]\right] = \boldsymbol{0}$. The covariance $\boldsymbol{\Sigma}_d$ can be written as:

$$\begin{aligned} \boldsymbol{\Sigma}_d &\triangleq \mathbb{E}\left[(\boldsymbol{y}_d - \mathbb{E}\left[\boldsymbol{y}_d\right])(\boldsymbol{y}_d - \mathbb{E}\left[\boldsymbol{y}_d\right])^T\right] \\ &= \mathbb{E}\left[\boldsymbol{A}_i^{d-1} \boldsymbol{w}_i[t-d](\boldsymbol{w}_i[t-d])^T (\boldsymbol{A}_i^{d-1})^T\right] \\ &= \boldsymbol{A}_i^{d-1} \mathbb{E}\left[\boldsymbol{w}_i[t-d](\boldsymbol{w}_i[t-d])^T\right] (\boldsymbol{A}_i^{d-1})^T \\ &= \boldsymbol{A}_i^{d-1} \boldsymbol{\Sigma}_i (\boldsymbol{A}_i^{d-1})^T \end{aligned}$$

■

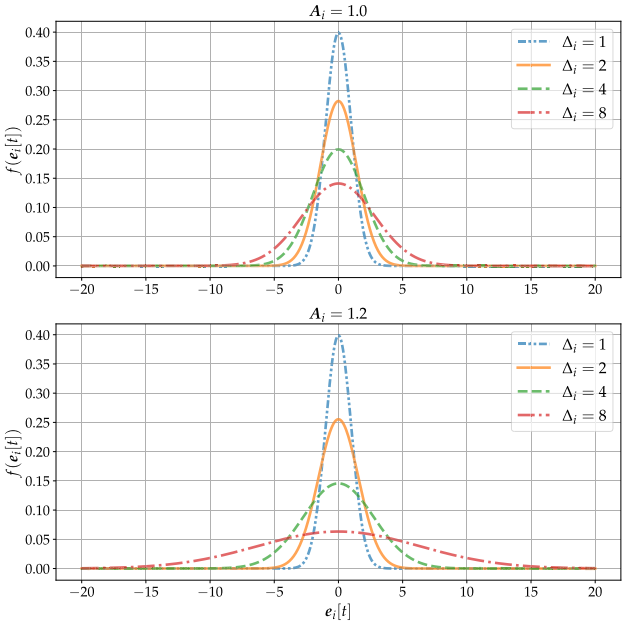

FIGURE 3. The probability density function of the estimation error $f(e_i[t])$ for varying AoI values. The distribution is characterized by $e_i[t] \sim \mathcal{N}(0, \Sigma_e)$ with $\Sigma_e = \sum_{d=1}^{\Delta_i[t]} A_i^{d-1} \Sigma_i (A_i^{d-1})^T$. Here, $A_i = \{1.0, 1.2\}$ and $\Sigma_i = 1.0$ are used.

The overall estimation error $\boldsymbol{e}_i[t]$, which is comprised of d independent addends, i.e., $\{\boldsymbol{y}_d[t] : 1 \leq d \leq \Delta_i[t]\}$, is characterized by the multivariate normal distribution $\boldsymbol{e}_i[t] \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_e)$. Since we are able to sum up the covariance matrices as the individual addends are independent RVs, it holds that:

$$\boldsymbol{\Sigma}_e = \sum_{d=1}^{\Delta_i[t]} \boldsymbol{A}_i^{d-1} \boldsymbol{\Sigma}_i (\boldsymbol{A}_i^{d-1})^T. \tag{10}$$

Here, it is important to emphasize that an increase in Δ_i leads to a new positive semi-definite addend on the RHS. Note that if $\boldsymbol{A}_i$ is a scalar, this corresponds to an increase in the variance of $\boldsymbol{e}_i$'s distribution. Let us illustrate this with a numerical example that considers a scalar loop with $\boldsymbol{A}_i = 1.2$ and $\boldsymbol{\Sigma}_i = 1.0$.

Fig. 3 depicts the probability density function (PDF) of estimation error for different control systems when the AoI ranges from one to eight. The figure shows how the PDFs become more stretched as information staleness at the estimator increases. Put differently, if we consider the estimation error as the deviation of the estimated state from the actual system state, the uncertainty of our estimation about the remote state grows with the increasing Δ_i. It is important to mention that this uncertainty does not grow at the same speed for every control application as information gets outdated. In fact, the sub-system with $\boldsymbol{A}_i = 1.2$ depicted at the bottom has a much wider distribution of the squared error at $\Delta_i = 8$ than the one with $\boldsymbol{A}_i = 1.0$ shown at the top. The figure can be interpreted as an illustration of how the significance of transmitting the next status update relates to the freshness property of information and to its context, i.e., who is sending and receiving the information, what is the purpose of conveying this information, etc. In our toy example illustrated in the figure, the context of communication is defined by the goal of uncertainty reduction at two destinations that are monitoring two remote processes with distinct system dynamics.

Although the estimation error is not a direct measure of control performance, it strongly affects the accuracy of control inputs. That is, with the growing uncertainty at each controller $\mathcal{C}_i$, the applied control inputs become sub-optimal due to the deviation between $\boldsymbol{x}_i[t]$ and $\hat{\boldsymbol{x}}_i[t]$. Consequently, $\boldsymbol{u}_i[t]$ is not able to drive the state towards the reference value correctly. This causes an increase in the overall control cost F_i since the state and the control effort grows. This phenomenon is shown in Fig. 4, where the relationship between a wrong state estimate and an imperfect control input is illustrated.

In Fig. 4, we see an example interplay between the AoI, system state, estimation, and the control input. The controller has outdated information and expects $\boldsymbol{x}_i[t']$ to be correctly driven to the equilibrium point of $\boldsymbol{x}_i = 0$. As the AoI increases further, the controller does not take any immediate action due to the lack of recent information, i.e., $\boldsymbol{u}_i[t] = 0$ for $t \in [t', t' + 2]$. Upon receiving a new update, the controller improves its estimation and applies a non-zero control input at $t = t' + 3$ to drive the state back to zero. Both the state deviation and the following control effort contribute to the LQG cost and lead to a degradation in the control performance.

C. TASK-ORIENTED COMMUNICATIONS AND PROBLEM STATEMENT

The optimal state feedback gain matrix $\boldsymbol{L}^*{}_i$ from (4) is calculated by assuming ideal communication links between the components of a feedback control loop. However, this contradicts our considered scenario, in which the state observations are sent over a physical wireless link. Therefore, to limit the deviation of controller design from optimality w.r.t. the LQG cost, the network should aim to reduce the error between the actual and estimated states induced by the imperfect communication links within sensor-controller pairs.

As the network consists of multiple control sub-systems and the available bandwidth is limited, we need to identify the most relevant transmissions and fit them into the available network resources to improve performance. Considering the fact that in our setup, the source-destination pairs are represented by sensors and controllers, this would correspond to selecting the highest instantaneous uncertainty reduction at the controller in case of a successful transmission, hence scheduling the user with the highest MSE.[4]

On the contrary, the control theory uses LQG cost F_i as a metric to quantify the success level in accomplishing the control goal. Although it is challenging to formulate the

[4]Note that the uncertainty reduction happens only if the transmission is successful. This requires the consideration of packet success probability. In Sec. V-B we discuss in detail how the link reliability is incorporated into scheduling decisions in our setup.

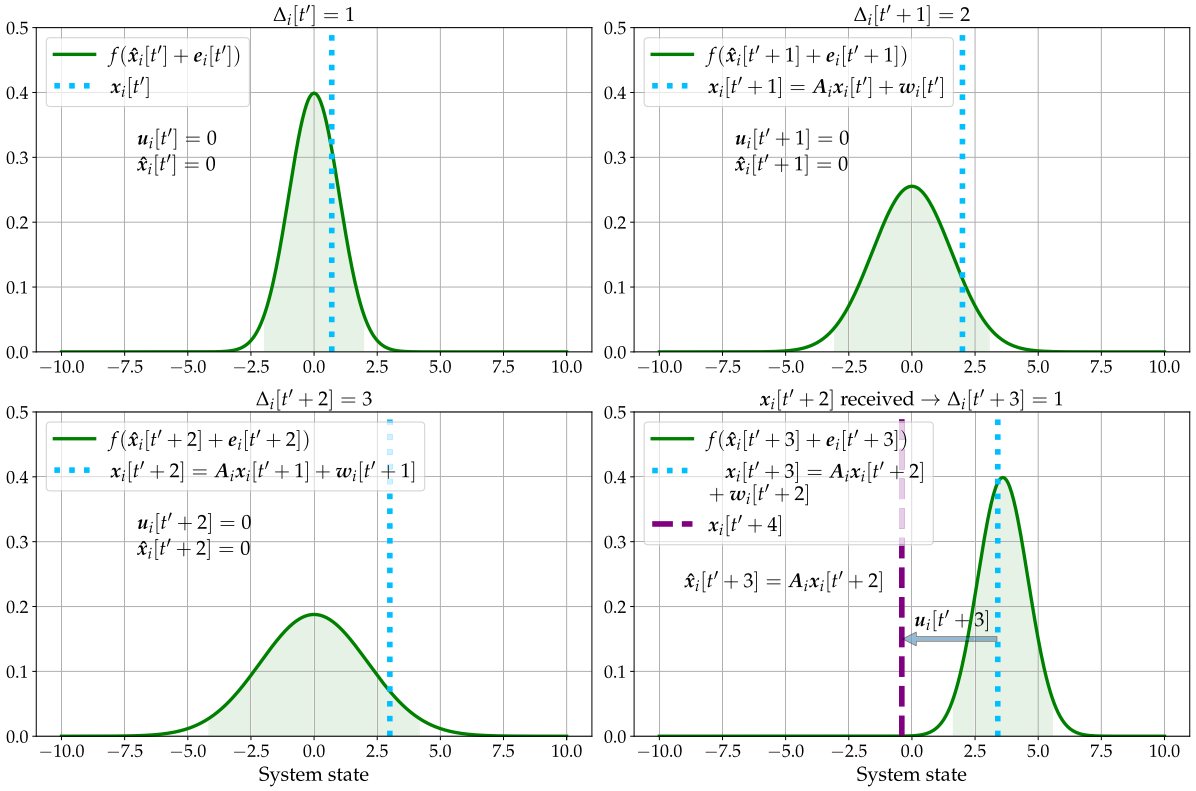

FIGURE 4. An example snapshot of the system state $x_i[t]$, control input $u_i[t]$ and estimated state $\hat{x}_i[t]$. The figures illustrate how the state drifts away from the reference value due to missing status updates about recent changes. Please notice that the distribution of the estimation error is more stretched as Δ_i increases.

exact relationship between MSE and LQG cost analytically, they are strongly intertwined, as discussed in the previous section. Having that said, our approach exploits the indirect relationship between estimation and control performances. In other words, by reducing the overall MSE in the network, we expect to reduce the LQG cost and thus improve the quality of control.[5] Therefore, our final goal is to implement a customized wireless MAC protocol $\boldsymbol{\pi}$ on SDRs to minimize the average LQG cost per control sub-system, i.e.:

$$\boldsymbol{\pi} = \arg\min_{\boldsymbol{\pi}} \lim_{T\to\infty} \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N} F_i(\boldsymbol{\pi})\right], \tag{11}$$

with:

$$F_i(\boldsymbol{\pi}) \triangleq \frac{1}{T}\sum_{t=0}^{T-1} (\boldsymbol{x}_i[t])^T \boldsymbol{Q}_i \boldsymbol{x}_i[t] + (\boldsymbol{u}_i[t])^T \boldsymbol{R}_i \boldsymbol{u}_i[t]. \tag{12}$$

As in (3), $F_i(\boldsymbol{\pi})$ is the linear quadratic cost when $\boldsymbol{\pi}$ is employed. Section V presents two examples of such wireless MAC protocols using MSE in the context of control-oriented communications. While the first is an existing protocol from the literature, the second is a new solution firstly proposed in this work.

[5]This expectation is based on the results of a previous work [25] that studies an FCFS discrete-time queue in a simulation-based setup.

V. MAC PROTOCOLS FOR REAL-TIME NCS

This section introduces various selected MAC protocols we have implemented and tested in our experimental framework. First, we explain three existing contention-based protocols in V-A. Next, we briefly present three centralized solutions: 1) Round Robin scheduling, 2) WiFresh from [32], and 3) Maximum Error First from [6]. In subsection V-B4, we will introduce a new protocol that combines the core ideas from V-B2 and V-B3, consolidating the strong sides of both methods. As we are will show in section VII, our solution outperforms the other methods concerning control performance.

A. CONTENTION-BASED PROTOCOLS

1) ALOHA

The simplest MAC protocol we have implemented in our experimental framework is the pure ALOHA proposed in [38]. It is based on the simple idea of transmitting any incoming data packet when it is ready to send. We know from the basics of wireless communications that this results in high packet loss if the network traffic load is high.

2) SLOTTED ALOHA (SA)

Slotted ALOHA, which has originally been proposed in [39], is based on the idea that time is divided into equally long time slots, and each user transmits with a constant channel access

probability (CAP) p_i when a slot begins or backs off with $1 - p_i$ probability.

SA has recently been studied in the context of AoI in [33], [40].[6] In particular, in [33], the authors use SDRs programmed with GNU Radio similar to this work. As derived in [40], by using SA each loop achieves a mean AoI $\bar{\Delta}_{SA}$ given as:

$$\bar{\Delta}_{SA} = \frac{1}{p(1-p)^{N-1}}, \tag{13}$$

where $N \geq 3$ is the number of users in the network. As proven in the same work, the age-optimal CAP p^* for SA is given as $p_i^* = p^* = 1/N$, $\forall i$. Throughout the paper, we assume that when SA is employed, the optimal channel access probability is selected.

The nodes should continuously transmit the most recent state information for (13) to be valid,. By adopting the LCFS queuing discipline, we ensure that this requirement is fulfilled. Moreover, the slot frequency and the frequency of the aging process should coincide, which is the case for our work.

Time synchronization among SDRs, which is necessary for time-slotted protocols such as SA, is realized through periodic transmission of beacon packets. Further details for synchronization are given in section VI-B.

3) AGE-DEPENDENT RANDOM ACCESS PROTOCOL (ADRA)

Chen et al. proposed the ADRA protocol in [40] as an optimized age-dependent stationary randomized MAC policy for large-scale networks. It is a threshold-based policy in which each user accesses the shared medium with a predetermined CAP $p = p_i$, $\forall i$ only if its instantaneous AoI is not below a certain threshold value $\delta_i = \delta$, $\forall i$, i.e.:

$$p_i[t] = \begin{cases} 0, & \text{if } \Delta_i[t] < \delta \\ p, & \text{if } \Delta_i[t] \geq \delta \end{cases} \tag{14}$$

Evidently, from (14), each source SDR $\mathcal{S}_i$ needs to know the instantaneous AoI at the receiver to decide whether it is eligible for data transmission. However, as the sensors do not have the perfect knowledge of the reception history at the receiver, the instantaneous AoI has to be estimated remotely. To overcome this issue, which is not directly addressed in theoretical works, our framework uses acknowledgment (ACK) packets transmitted by the GW device upon a successful reception. The instantaneous AoI estimation at the sensors is based on the assumption that every unacknowledged packet is lost on the sensor-to-GW link. In case of an unreliable control channel with a high loss ratio of the ACK packets or when they arrive with a significant delay, the ADRA protocol would overestimate the AoI at the receiver, thus leading to more frequent and redundant transmissions, hence increased network load.

[6] In [33], [40] authors refer to slotted ALOHA as "Age-independent random access (AIRA)".

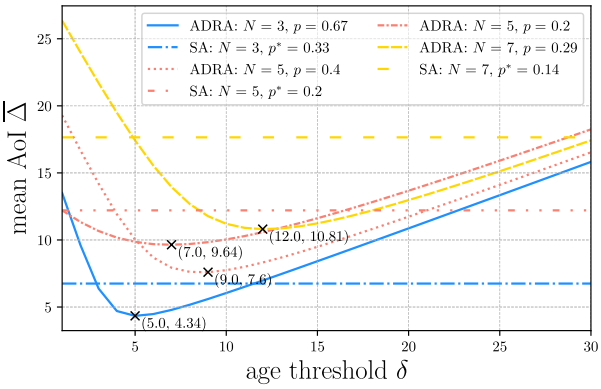

FIGURE 5. Network-wide mean AoI $\bar{\Delta}$ is plotted against age-threshold δ for selected number of users, $N = \{3, 5, 7\}$. p denotes the channel access probability for the ADRA protocol. The horizontal lines show the minimum achievable AoI for slotted ALOHA with the age-optimal CAP $p^* = 1/N$.

In their work, the authors derive the network-wide mean AoI for ADRA protocol as:

$$\bar{\Delta}_{ADRA} = \frac{\delta}{2} + \frac{1}{pq} - \frac{\delta}{2(\delta pq + 1 - pq)}, \tag{15}$$

with the successful status update probability q. To obtain the value for q we refer to the original paper. Moreover, the optimal values for δ and p can be obtained numerically. As suggested by the authors, we used the bisection method to find the optimal δ^* and p^* values. When comparing the ADRA protocol to others, like in the SA case, we have used the optimal values for δ and p during our measurements. Fig. 5 depicts the network-wide mean AoI $\bar{\Delta}_{ADRA}$ and $\bar{\Delta}_{SA}$ for various number of users $N = \{3, 5, 7\}$ and varying δ up to 30. It is evident from the figure that when the right configuration is selected, the ADRA protocol outperforms the slotted ALOHA w.r.t. the mean AoI.

B. CONTENTION-FREE PROTOCOLS

1) ROUND ROBIN (RR)

The RR is a well-known scheduling policy from the literature that prioritizes each user in a fixed order. Therefore, it is neither a channel-aware nor an application-aware scheduling algorithm. In our RR implementation, the users are prioritized in the same order as their unique control loop ID i. Given that at any time slot $t \in \mathbb{N}^+$ only a single source node $i \in \{1, 2, \ldots, N\}$ is scheduled, the next node to schedule can be obtained by the simple rule:

$$i^*[t] = \arg\min_i \{t + N - i \mod N\}, \tag{16}$$

with the modulo operator mod . There is always a single user i that makes $t + N - i \mod N = 0$, where $i^*[1] = 1$, $i^*[2] = 2$, etc. In our framework, we enforce synchronization among users with the help of beacon packets as in SA and ADRA protocols. Therefore, each source node i can track the current time slot index t and thus detect the next allocated slot by (16).

With constant number of users, RR results in periodical prioritization of every user. That is, every source node is

scheduled once in every N slots. If the destination can successfully decode all transmitted updates, the discrete-time AoI of each user experiences a decrease from N to 1 with a periodicity of N slots. In other words, every time when AoI reaches $\Delta_i[t'] = N$, it is followed by a reset to $\Delta_i[t'+1] = 1$ in the subsequent slot. As a result, the long-term mean AoI of each source node is equivalent to its mean AoI throughout N slots, which can be derived as a sum of arithmetic sequence as follows:

$$\bar{\Delta}_{RR} = \frac{1}{N}\left(\frac{N}{2}(1+N)\right) = \frac{N+1}{2}. \tag{17}$$

Despite its simple operation, RR comes with some drawbacks in practical deployment. In addition to its dependence on time synchronization, the RR may cause underutilization of the network resources. In particular, the RR allocates a certain amount of resource units, e.g., a time slot, exclusively to a user. This implies that the remaining portion of the resource is wasted if the transmission takes shorter than the allocated slot. Especially in connected robotics and remote monitoring scenarios, where the data packets are small, finding the suitable slot duration to accommodate exactly a single transmission becomes a challenge. Our results in section VII reveal the performance loss caused by shorter transmissions than a slot duration. However, in this work, we do not tackle the slot duration adaptation problem and choose a fixed length throughout our measurements.

2) WIFRESH

One of the most prominent examples of practical AoI research is WiFresh [32], which is a polling-based protocol. Similar to this work, the authors consider multiple sources transmitting via SDRs to a base station (BS). The BS tracks the AoI of each source process and asks for a status update packet by sending a poll request. Additionally, it estimates the channel reliability $r_i^{ch}(t)$ between a source device i and the BS by the following equation:

$$r_i^{ch}(t) = \frac{RX_i^D(t)+1}{TX_i^P(t)+1}, \tag{18}$$

where $RX_i^D(t)$ and $TX_i^P(t)$ denote the number of successfully received data packets and transmitted poll packets in the last 0.5 seconds, respectively. The next source node to poll is then determined by the max-weight policy as:

$$i^*(t) = \arg\max_i \{r_i^{ch}(t)\tilde{\Delta}_i(t)\}, \tag{19}$$

with $\tilde{\Delta}_i(t)$ being the estimated age of the freshest information about source node i. The necessity for the AoI estimation arises because the source and destination nodes are not co-located, and the sampling instances of the source nodes are unknown to the GW. Therefore, the GW has to estimate the AoI remotely by tracking the elapsed sampling periods since the latest reception. Our approach is similar to the one considered in [32]. We would like to mention that this is an example challenge of system research that is revealed only prior to deployment and may be hidden for purely theoretical works.

Moreover, please notice the round brackets we have used for the variables in (19), which has the following reason: WiFresh is a polling-based protocol that operates asynchronous to the time-slotted model we have introduced before. In other words, the GW does not have any notion of a network time slot, and, therefore immediately begins with the following polling procedure once the outstanding poll packet has been responded to by a data packet.

The channel awareness of WiFresh makes it suitable for environments where the nodes are highly mobile, thus experiencing time-varying link quality. In addition, as it does not rely on synchronization among users, one can argue for its lower complexity than SA or ADRA. It is clear that, in contrast to random access protocols, the packet success ratio is expected to be much higher as simultaneous channel access is avoided by virtue of the centralized polling mechanism.

3) MAXIMUM ERROR FIRST SCHEDULER (MEF)

The MEF scheduler has been proposed in [6] for two-hop cellular networks where the users are feedback control loops. [6] suggests employing the MSE from (9) as the scheduling metric in a time-slotted resource allocation problem. As a result, at each time slot t, the next user to schedule is determined as:

$$i^*[t] = \arg\max_i \{MSE_i[t]\}. \tag{20}$$

The MEF scheduler is an example of control-aware scheduling policies for wireless NCS that has only been studied in the context of theoretical research. To the best of our knowledge, there has not been any previous work that implements the algorithm in an experimental setup. Therefore, the following design choices have been made in order to implement the MEF scheduler in our framework:

- The GW broadcasts a beacon packet every 20 time slots that contains the transmission schedule during that period.
- Only the source node $i^*[t]$ that has been scheduled for transmission at time slot t accesses the channel.
- The GW neglects the packet loss probability and allocates each of those 20 slots in advance as if all transmissions were to be successful.

We provide more details on beacon packets and time synchronization later in subsection VI-B.

4) OUR PROPOSED POLLING-BASED MEF SCHEDULER (pMEF)

The main difference between MEF and WiFresh is that MEF considers control system-dependent parameters implicitly through MSE. On the other hand, WiFresh does not operate in a slotted fashion in contrast to the MEF scheduler. This feature allows WiFresh to reduce the amount of idle time between two consecutive transmissions if the response to a poll packet comes earlier than the beginning of the next

slot.[7] Moreover, the MEF scheduler was initially proposed as a channel-unaware scheduling policy, as evident from (20). Therefore, we propose to combine the strengths of both schedulers in a polling-based, channel- and control-aware scheduler, that determines the next source node to schedule as:

$$i^*(t) = \arg\max_i \{r_i^{ch}(t) MSE_i(t)\}, \tag{21}$$

with $r_i^{ch}(t)$ as in (18). The MSE is obtained by substituting the instantaneous estimated AoI $\tilde{\Delta}_i(t)$ into (9). As in the WiFresh case, the round brackets are used to emphasize asynchronous operation of the scheduler to sampling process in contrast to time-slotted implementations such as MEF or RR.

C. A NEW METRIC FOR CONTROL-AWARE SCHEDULING: THE NORMALIZED MSE (nMSE)

The MSE, as defined in (9), has been used for control-aware scheduling in previous works [6], [25]. However, by definition, it is strongly system-dependent. Hence its unit varies from one control application to another. As a result, when making scheduling decisions that consider system parameters, as in the case of the MEF scheduler or our proposed pMEF scheduler, one cannot employ the MSE in its raw form in systems design. In other words, it may not capture the urgency of transmission for different applications. More precisely, the scheduling decision based on raw MSE would correspond to the comparison of multiple numbers in different units and orders of magnitude.

As a solution to this problem, we propose and employ the *normalized mean squared error (nMSE)*, which is defined as:

$$\|MSE_i(t)\| \triangleq \frac{MSE_i(t)}{MSE_{\Delta_i=1}} \tag{22}$$

where $MSE_{\Delta_i=1} \triangleq MSE_i(t)|_{\Delta_i(t)=1}$. In simple words, we divide the MSE of each control sub-system by the MSE when the AoI is one. It is important to mention that the normalization factor, i.e., $MSE_{\Delta_i=1}$, is equal to the trace of the covariance matrix, which is the only addend in the RHS of (9) when $\Delta_i(t) = 1$. Similar to the MSE, nMSE is zero when $\Delta_i(t) = 0$ and is strictly increasing with Δ_i since the denominator takes a positive value.[8]

The normalized MSE can be seen as an adaptation of the MSE to the so-called "age-penalty" or "non-linear aging" from the existing literature [41], [42]. The concept of non-linear aging has been proposed to represent the information losing its usefulness with varying speed over time. In those works, the authors investigate well-known non-linear functions of AoI, such as $f(\Delta) = e^{a\Delta}$ and $f(\Delta) = \Delta^a$, with $a \geq 0$. In contrast to such system-independent penalty functions, the nMSE is a way of defining control-aware age-penalty functions as it depends on AoI, the system matrix, and the noise covariance matrix. Moreover, it captures the growth of the mean squared estimation error relative to the value that it takes if the information has been generated in the previous sampling period. Through normalization, we are able to unify heterogeneous control applications in a dimensionless quantity. Section VII-C presents a case study utilizing the nMSE for wireless resource management, where the MSE is not directly applicable.

By definition, the pMEF scheduler depends on the system dynamics as it utilizes the instantaneous nMSE, which is a normalized version of the MSE. More specifically, the centralized scheduler requires the knowledge of the system-dependent parameters $\boldsymbol{A}_i$ and $\boldsymbol{\Sigma_i}$ to be able to obtain $MSE_i(t)$ given the AoI as in (9). Nevertheless, as those parameters are time-invariant since we are dealing with LTI systems, a single information exchange prior to operation is sufficient.

VI. DESIGN AND IMPLEMENTATION

A. HARDWARE AND SOFTWARE

Our experimental setup consists of $N \in \{2, 3, \ldots, 15\}$ plant processes programmed in Python programming language. Each plant process $\mathcal{P}_i$ generates periodic packets, which are forwarded to $\mathcal{S}_i$ using a UDP socket.[9] Once the SDR receives the packet, it traverses through multiple packet processing blocks programmed in C++ with GNU Radio.

Our testbed is composed of eight computers running Ubuntu 20.04.3 LTS operating system. Ettus Research's USRP™ B200mini-i and B205mini-i SDRs are used as the source and destination for wireless data transmission. Fig. 6 shows a photo of our experimental testbed during measurements involving twelve control sub-systems. Note that there are twelve SDRs responsible for the transmission of status update packets and an additional SDR serving as GW. In contrast to [33], we have not directed the data flow of multiple source processes into a single SDR.

In our framework, we have a clear separation of the application layer and the wireless communication stack. Specifically, the status update packets are generated and written to a local UDP socket that is read by the GNU Radio signal processing blocks. The wireless network behind the UDP socket is entirely transparent to the application, i.e., the control system. Similarly, at the GW, the interfacing between the GNU Radio process and the controller processes is done by employing local UDP sockets. By choosing a clear separation between the wireless networking stack and the application layer, we aim to simplify the integration of any internet protocol-based application into our framework, thereby removing the barrier to its adoption.

An automation script is used to reduce the influence of a human operator on the results when the measurements are started, repeated, and stopped. Additionally, we ignore the first and last five seconds of each thirty-seconds-long

[7] In section VII, we discuss the effect of this property of polling on the AoI and control performances in detail.

[8] The strictly increasing property of MSE had been shown in previous works, e.g., in [6].

[9] Each plant process $\mathcal{P}_i$ and SDR $\mathcal{S}_i$ run on the same machine.

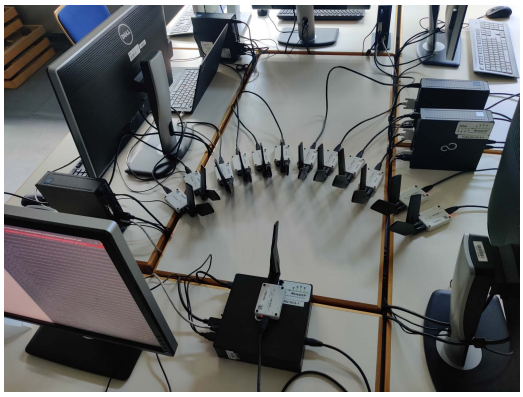

FIGURE 6. A photo of our testbed while taking measurements with twelve control sub-systems.

measurement run in the data collection to avoid transitional effects of the startup and completion phases.

B. SYNCHRONIZATION

Time synchronization is a prerequisite for realizing time-slotted MAC protocols introduced in Sec. V, such as SA and ADRA random access protocols or RR scheduling. To that end, we follow a similar approach as in [33] and employ periodic transmission of beacon packets at the beginning of each 20 slots long *frame* structure. A beacon packet is composed of three main fields:

- **MAC header**: Contains information such as packet type, MAC sequence number, source and destination addresses.
- **Payload**: Contains information specific to the employed MAC protocol, such as frame length, time slot duration, i.e., $T_s = 10$ ms, time slot index, and the transmission schedule, if applicable, e.g., for the MEF scheduler.
- **CRC**: Contains the 16-bits long cyclic redundancy check (CRC) field used to detect errors in the data reception, mainly caused by packet collisions in our setup.

Upon the detection of a beacon packet, each $\mathcal{S}_i$ marks the current time as the beginning of the next frame and sets the current slot to the time slot index contained within the *Payload* field.[10] This is based on the assumption that the difference in processing delays at each $\mathcal{S}_i$ is negligible. GNU Radio's *high_res_timer* library has been used for time stamping purposes with high resolution.

VII. EVALUATION

In order to simplify the implementation and the analysis of the results, we have selected scalar control loops of three different classes. The least challenging category of systems are $\mathcal{I}_{easy} = \{1, 4, 7, 10, 13\}$ with the system matrix $\boldsymbol{A}_1 = \boldsymbol{A}_4 = \cdots = \boldsymbol{A}_{13} = 1.0$. The second and third classes of systems, i.e., $\mathcal{I}_{mid} = \{2, 5, 8, 11, 14\}$ and $\mathcal{I}_{hard} = \{3, 6, 9, 12, 15\}$ have the system matrices $\boldsymbol{A}_2 = \cdots = \boldsymbol{A}_{14} = 1.1$ and $\boldsymbol{A}_3 = \cdots = \boldsymbol{A}_{15} = 1.2$, respectively. The relationship between the system matrix and difficulty of control can be deducted from (1) intuitively, which shows the proportionality between the current state $\boldsymbol{x}_i[t]$ and the next state $\boldsymbol{x}_i[t+1]$. Additionally, the input and covariance matrices are chosen as $\boldsymbol{B}_i = 1.0, \forall i$ and $\boldsymbol{\Sigma}_i = 1.0, \forall i$.

The design of the LQR controller has been done with $\boldsymbol{Q}_i = 100.0$ and $\boldsymbol{R}_i = 1.0$ for all control loops. In simple words, this means that the state error is penalized a hundred times more than the control effort while calculating the optimal control input in the infinite horizon LQR problem. The optimal $\boldsymbol{L}^*{}_i$ is obtained from the solution of the discrete algebraic Riccati equation as given in (6).

Our results are obtained by performing 20 repetitions of 30 seconds long measurement runs. As mentioned in VI-A, we did not consider each run's first and last five seconds to avoid transitional effects. Therefore, the evaluation of each metric starts after the 500-th discrete time step and ends with the 2500-th time step. As a result, the network-wide mean AoI is obtained as:

$$\bar{\Delta} = \frac{1}{2000 \cdot N} \sum_{t=501}^{2500} \sum_{i=1}^{N} \Delta_i[t] \tag{23}$$

where each instantaneous AoI $\Delta_i[t]$ is measured at the end of each sampling period t. In order to capture the control quality, we have selected the mean squared estimation error $\overline{MSE}$ and the LQG cost $\overline{F}$ with:

$$\overline{F} \triangleq \frac{1}{2000 \cdot N} \sum_{i=1}^{N} \sum_{t=501}^{2500} (\boldsymbol{x}_i[t])^T \boldsymbol{Q}_i \boldsymbol{x}_i[t] + (\boldsymbol{u}_i[t])^T \boldsymbol{R}_i \boldsymbol{u}_i[t] \tag{24}$$

with $\overline{F}_i$ as in (3). The calculation of $\overline{MSE}$ is analogue to $\overline{\Delta}$ and can be obtained by replacing $\Delta_i[t]$ in (23) with $MSE_i[t]$, i.e.:

$$\overline{MSE} = \frac{1}{2000 \cdot N} \sum_{t=501}^{2500} \sum_{i=1}^{N} MSE_i[t] \tag{25}$$

It is essential to state that due to the selection of $\boldsymbol{\Sigma}_i = 1.0, \forall i$, the denominator in the RHS of (22) becomes 1. Therefore, the raw MSE and the normalized nMSE are equivalent for sections VII-A and VII-B, i.e., $\overline{MSE} = \overline{\|MSE\|}$. Note that this does not apply to Sec. $VII - C$, in which we introduce a new control loop class into the network.

A. CONTENTION-BASED PROTOCOLS' PERFORMANCE

In Sec. V-A, we have introduced three contention-based MAC protocols, namely ALOHA, SA, and ADRA. Fig. 7 presents the measured mean AoI and its theoretical expectation, i.e., $\bar{\Delta}_{SA}$ and $\bar{\Delta}_{ADRA}$. We do not include ALOHA in the figure because of presentation purposes. That is, the ALOHA protocol performs significantly worse than the other two already for a very low number of users in the network. For instance,

[10] Information on slot duration and frame length are contained in the beacon packet as well, although they are assumed constant in this work. The reason is to increase the flexibility of our implementation and facilitate the study on the effect of varying slot on network and control performance.

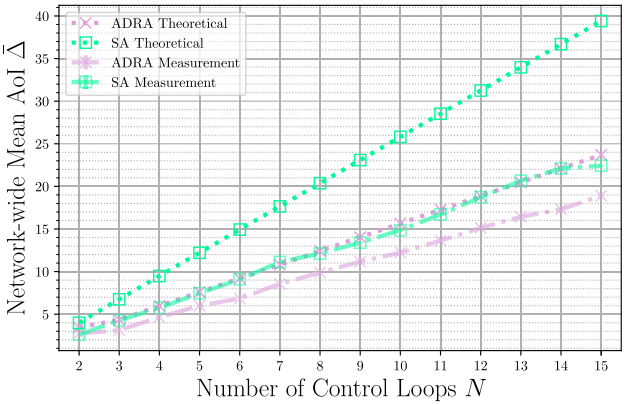

FIGURE 7. Mean AoI of contention-based access protocols, i.e., slotted ALOHA (SA) and age dependent random access (ADRA). Vertical bars illustrate 99% confidence intervals.

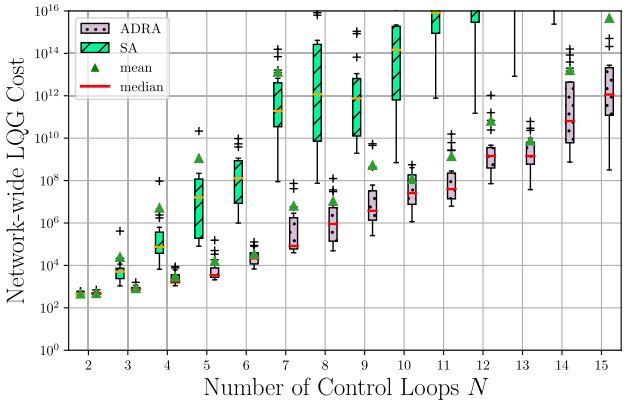

FIGURE 8. Control performance of contention-based protocols. It is captured by the LQG Cost $\bar{F}$ as defined in (24), where a lower LQG cost represents a higher performance. y-axis has been limited for presentation purposes.

the instantaneous AoI up to 1900 was observed in one of the measurements for $N = 3$. Therefore, we omit ALOHA in the remaining evaluation since it is unsuitable for time-sensitive wireless networks with multiple users.

From the figure, we observe that the ADRA protocol outperforms the age-independent protocol SA as expected. However, we observe a deviation between the measurement results and the theoretical results from [33]. In fact, our framework is able to achieve better results than analytical mean values. In our opinion, this has two main reasons: 1) simultaneous transmissions are being decoded despite their overlapping. This issue has already been raised in [33]. 2) The transmission of a packet does not occupy a full slot. In our framework, a slot is 10 ms long, whereas our measurements indicate an approximate transmission duration of 3 ms for each data packet. As we do not force any synchronization in the application layer, this allows some of the packets to miss each other in time, although they are transmitted in the same slot. This phenomenon increases the packet delivery rate per slot far beyond one, which causes an improvement over the theoretical expectation.

If we look at Fig. 8, we observe that the LQG cost representing the control performance shows divergent behavior for both contention-based protocols. Especially, already for $N = 8$, SA reaches an LQG cost up to 10^{16}, indicating instability of the system state. The same applies to ADRA for $N = 15$, showing the inadequacy of these protocols for multi-user scenarios with time-sensitive control applications.

B. CONTENTION-FREE PROTOCOLS' PERFORMANCE

From the fundamentals of communications theory, we know that the main strength of the contention-free protocols over random access is their significantly lower packet loss rate. This comes at a price of increased complexity and communication overhead, as in the case of polling-based protocols. First, let us analyze the performance of contention-free protocols w.r.t. information freshness.

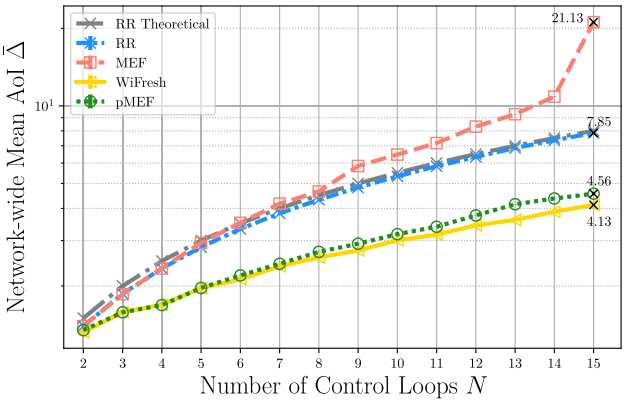

FIGURE 9. Mean AoI of contention-free protocols, i.e., round robin (RR), maximum error first (MEF), WiFresh and polling-based MEF (pMEF). Vertical bars illustrate 99% confidence intervals. y-axis is drawn on logarithmic scale.

Fig. 9 presents the network-wide AoI for $2 \leq N \leq 15$. We observe that the polling-based WiFresh protocol outperforms all others by at least 10% as in the case of the pMEF scheduler. This is an expected result due to the heterogeneous prioritization of sub-systems by the pMEF algorithm caused by its control awareness. In other words, while WiFresh considers the AoI and therefore polls sub-systems in a RR fashion under constant channel conditions, the pMEF allocates a bigger portion of the network resources to the class of more challenging sub-systems $\mathcal{I}_{hard}$. This leads to an unbalanced distribution of AoI in the network and increases $\bar{\Delta}$. Nevertheless, as we are going to show later in this section, pMEF is able to achieve better performance for the given control task via its ability to identify the most relevant information.

In our setup, the average polling time, which is the time between a poll request and the reception of the corresponding data packet, is shorter than a time slot. Thus, the beacon-based protocols, i.e., RR and MEF, achieve lower throughput, fewer transmissions, and hence higher AoI than WiFresh and pMEF.

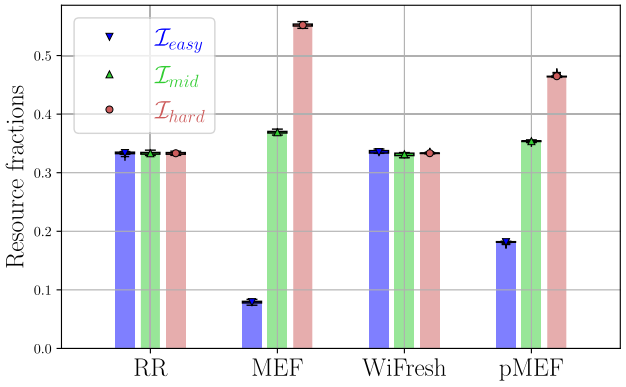

FIGURE 10. The fractions of network resources allocated to each control class. Control-unaware protocols. i.e., RR and WiFresh treat all system classes equally. On the other hand, the control-aware protocols, i.e., MEF and pMEF lead to an unbalanced distribution of resources.

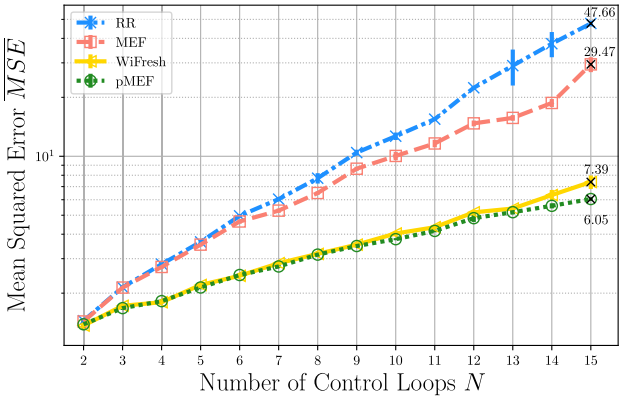

FIGURE 11. Estimation performance of contention-free protocols, i.e., RR, MEF, WiFresh and pMEF. It is captured by the MSE as defined in (25). Note that a lower $\overline{MSE}$ represents a higher performance. Vertical bars illustrate 99% confidence intervals. y-axis is drawn on logarithmic scale.

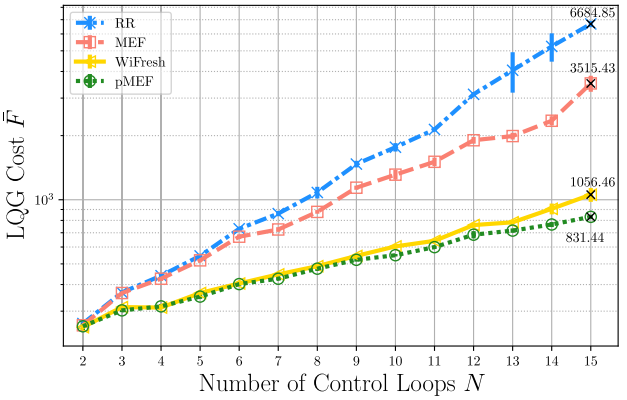

FIGURE 12. Control performance of contention-free protocols, i.e., RR, MEF, WiFresh and pMEF. It is captured by the LQG Cost $\bar{F}$ as defined in (24). Note that a smaller $\bar{F}$ represents a higher performance. Vertical bars illustrate 99% confidence intervals. y-axis is drawn on logarithmic scale.

This leads to resource scarcity and longer idle periods for less critical sub-systems in the case of MEF. As a result, the gap in AoI between control-aware and control-unaware protocols, i.e., MEF and RR, is increased. Particularly, MEF achieves $\bar{\Delta}$ beyond 20 for $N = 15$, while RR achieves less than 10 for the same number of sub-systems. Note that the experimental AoI for RR matches the theoretical mean AoI derived in (17).

As a next step, we present Fig. 10, which shows the fraction of network resources allocated to each control class $\mathcal{I}_{easy}$, $\mathcal{I}_{mid}$, and $\mathcal{I}_{hard}$. In fact, it confirms that all classes are treated equally when RR and WiFresh are applied, whereas MEF and pMEF schedule more challenging sub-systems more frequently.

So far, we have only presented the results of contention-based protocols concerning information freshness. However, as we are dealing with NCSs that are communicating to achieve a specific control goal, we need to go beyond AoI and focus on control-related KPIs such as MSE and control cost. Firstly, we present Fig. 11, which shows the estimation performance in the network captured by $\overline{MSE}$. It is evident from the figure that the control-aware protocols, i.e., MEF and pMEF, outperform their direct competitors, i.e., RR and WiFresh, respectively. Especially, as the resource scarcity becomes more significant, e.g., $N = 15$, the importance of control awareness stands out. That is, MEF is able to achieve relatively lower $\overline{MSE}$ than RR, although it performs worse than RR in terms of $\overline{\Delta}$. One can also say that the information freshness is traded for increased estimation performance. Similar behavior is observed when pMEF and WiFresh are compared, i.e., pMEF outperforms WiFresh by up to 18% when there are 15 control sub-systems in the network.

The MSE captures the estimation accuracy at the monitor. On the other hand, the quality of control is not measured by the estimation accuracy but rather by the state error and the control effort that is spent in order to drive the state to the desired set point. However, the control performance is strongly intertwined with the estimation accuracy, as discussed in detail in section IV-B. Due to this indirect relationship between the estimation and control performances, we observe a similar trend for the LQG cost as for MSE. Fig. 12 presents the main results of this work, i.e., the network-wide control cost $\overline{F}$ for different protocols. From the figure, we can see that pMEF is able to outperform the WiFresh protocol by up to 21%. The beacon-based protocols' performance follows a similar trend, with MEF outperforming RR by 47%. Furthermore, please notice that the contention-free protocols clearly outperform the contention-based schemes in terms of LQG cost.

C. A REAL-LIFE APPLICATION CASE STUDY: INVERTED PENDULUM

In the previous sections, we have shown the performance improvement of our proposed scheduling algorithm over the existing protocols from the SotA. The selected scalar control systems were theoretical ones to illustrate this effect in a simple scenario. In this section, we introduce the emulation of

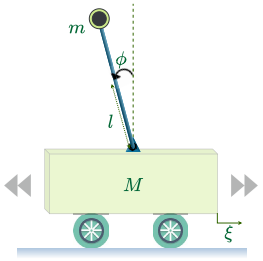

FIGURE 13. An inverted pendulum with motorized cart.

a real-life application to our network, the inverted pendulum (IP). IP is a well-studied control application that is widely used in control theory textbooks [36]. As depicted in Fig. 13, it consists of a pendulum mounted on a motorized cart where the controller's objective is to hold the pendulum in an upright position by moving the cart back and forward. For the sake of completeness, we provide the two continuous-time equations of motion around the unstable upward equilibrium:

$$(I + ml^2)\ddot{\phi} - mgl\phi = ml\ddot{\xi}, \tag{26}$$

$$(M + m)\ddot{\xi} + b\dot{\xi} - ml\ddot{\phi} = u, \tag{27}$$

for the state vector $\boldsymbol{x} = [\xi, \dot{\xi}, \phi, \dot{\phi}]^T$. Here, ξ is the position of the cart, ϕ is the deviation of the pendulum's position from equilibrium, u is the input force applied to the cart. M and m are the mass of the cart, mass of the pendulum, respectively. l denotes the length to pendulum's center of mass. In addition, b is the coefficient of friction for the cart and I is the moment of inertia of the pendulum. g is the standard acceleration due to gravity. The selected set of parameters are summarized in the following table:

M	0.5 kg
m	0.2 kg
b	0.1 N/m/s
l	0.3 m
I	0.006 kgm^2
g	9.81 m/s^2

As we are working with digital systems, we are interested in the discrete-time state-space representation of the form (1). Therefore, we select a sampling frequency of the system as 100 Hz that leads to the following state and input matrices:

$$\tilde{\boldsymbol{A}} = \begin{bmatrix} 1 & 0.01 & 0.0001 & 0 \\ 0 & 0.9983 & 0.0191 & 0.0001 \\ 0 & 0 & 1.0017 & 0.01 \\ 0 & -0.0049 & 0.3351 & 1.0017 \end{bmatrix}, \tilde{\boldsymbol{B}} = \begin{bmatrix} 0.0001 \\ 0.0182 \\ 0.0002 \\ 0.0454 \end{bmatrix}.$$

Moreover, the noise covariance matrix is selected as:

$$\tilde{\boldsymbol{\Sigma}} = \begin{bmatrix} 6.4 \cdot 10^{-7} & 0 & 0 & 0 \\ 0 & 4.9 \cdot 10^{-7} & 0 & 0 \\ 0 & 0 & 2.742 \cdot 10^{-5} & 0 \\ 0 & 0 & 0 & 4.874 \cdot 10^{-5} \end{bmatrix}.$$

The resulting linearized model approximates the inverted pendulum's behavior around the equilibrium point, i.e., $\phi = 0$, in discrete-time. Note that the linearization of non-linear dynamics is a well-established method to model system dynamics in control theory.

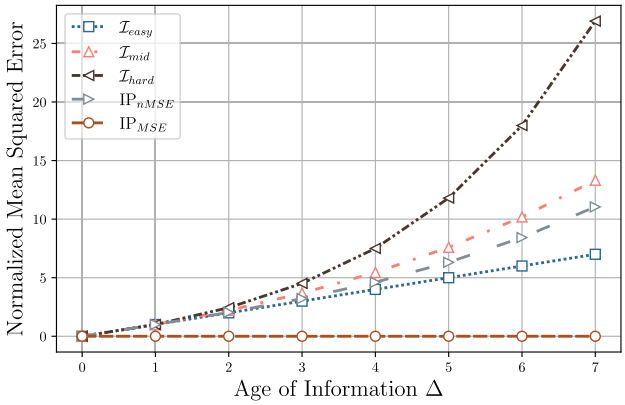

FIGURE 14. The normalized mean squared error plotted against AoI, Δ for different control sub-systems used in this work, i.e., inverted pendulum (IP), $\mathcal{I}_{easy}$, $\mathcal{I}_{mid}$, $\mathcal{I}_{hard}$ as before. In addition, we present the raw MSE for IP before the normalization step from (22) to illustrate its necessity.

The LQR method is used to determine the stabilizing feedback gain with weighting matrices $\boldsymbol{Q} = \text{diag}(5000, 0, 100, 0)$ and $\boldsymbol{R} = 1$ as in (3). In order to see the proposed nMSE metric in action, we repeat our measurements with 15 control sub-systems, where we substitute all the sub-systems of class $\mathcal{I}_{mid}$ with IPs, i.e., $\boldsymbol{A}_2 = \boldsymbol{A}_5 = \cdots = \boldsymbol{A}_{14} = \tilde{\boldsymbol{A}}$. We modify the input and noise covariance matrix analogously, as described above.

Fig. 14 shows the evolution of nMSE with increasing AoI together with the MSE for IP without the normalization step from (22). It illustrates the different growth speed of the nMSE for our considered application classes. Additionally, it reveals that the IP lies between the $\mathcal{I}_{hard}$ and $\mathcal{I}_{easy}$ classes concerning the nMSE. Note that due to the significant difference in magnitude of order between IP$_{MSE}$ and other curves, the usage of the raw MSE would lead to resource starvation for sub-systems of class IP and destabilization of the corresponding control loops. Therefore, the following discussion considers only the usage of nMSE both for MEF and pMEF strategies.

Similar to the previous subsections, we measure the control KPIs in order to validate the applicability of our proposed protocol for real-life applications. To that end, we have recorded pendulum angle ϕ and cart position ξ trajectories throughout 20 measurements. In order to narrow down the focus on IP, the following discussion is limited to the IP relevant metrics such as ϕ and ξ and does not contain the detailed state trajectories of other sub-systems of class $\mathcal{I}_{easy}$, i.e., $i \in \{1, 4, 7, 10, 13\}$ and of class $\mathcal{I}_{hard}$, i.e., $i \in \{3, 6, 9, 12, 15\}$.

In Fig. 15, we present an example trajectory of $\phi_i[t]$ in degrees for $t \in [500, 2500]$ and a randomly selected loop i. It has been recorded during one of the measurements when pMEF scheduler operating with nMSE was in use.[11] From the

[11] The selection of the specific measurement run and loop have been made randomly, and they do not represent an outlier w.r.t. control performance.

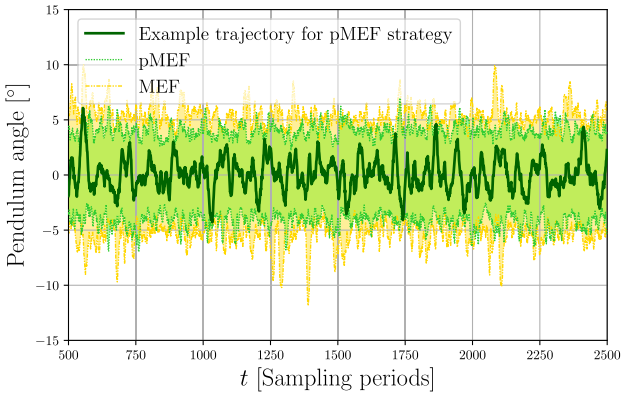

FIGURE 15. An example trajectory of the pendulum angle ϕ when maximum error first (MEF) and polling MEF schedulers are applied. ϕ is plotted in degrees.

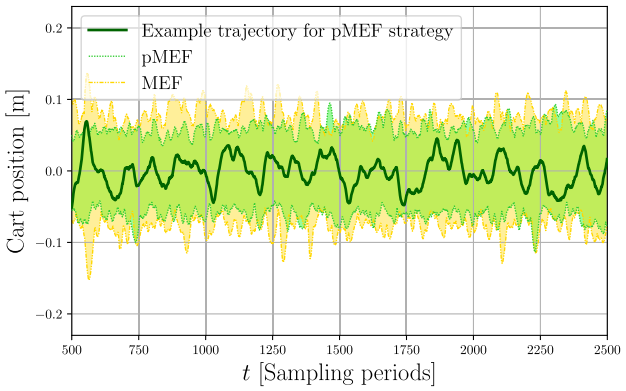

FIGURE 16. An example trajectory of the cart position ξ when maximum error first (MEF) and polling MEF schedulers are applied. ξ is plotted in meters.

figure, we are able to observe that the pendulum angle is kept within $\pm$ 5 degrees. In addition, Fig. 15 shows the maximum and minimum values that are reached by all IPs in the network when MEF and pMEF is employed. Due to the higher sensor-to-controller delivery rate of pMEF compared to MEF, the pMEF achieves a better control performance w.r.t. ϕ. The same conclusion can be drawn if we look at Fig. 16 where the minimum and maximum ξ values are presented. In particular, we are able to observe larger spikes of ξ achieved by MEF than pMEF throughout the measurements. To put it another way, the cart needed to move further away from its desired set point, i.e., $\xi = 0$ in order to keep the pendulum upright.

Last but not least, Fig. 17 depicts the average nMSE achieved when RR, MEF, WiFresh, and pMEF are used. Each boxplot represents a control class-scheduling strategy combination. In other words, it presents the contribution of each control class to the overall nMSE performance separately. The figure shows that control-unaware strategies, namely the RR and WiFresh strategies, lead to an increased nMSE for the $\mathcal{I}_{hard}$ class systems. This is an expected result of equal treatment of all sub-systems in the network, which lead to

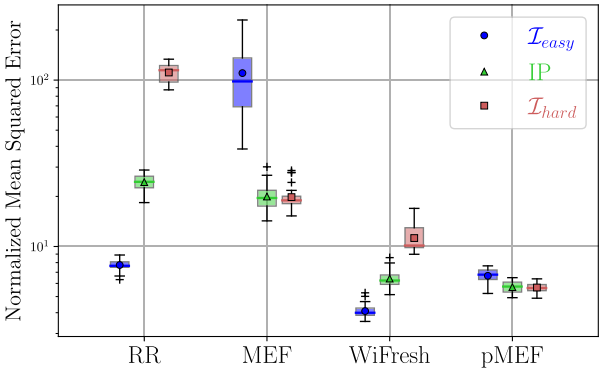

FIGURE 17. The normalized mean squared error (nMSE) achieved when round robin (RR), maximum error first (MEF), WiFresh and polling MEF schedulers are employed. y-axis is drawn on logarithmic scale.

higher error values for more critical applications. On the other hand, as we know from Fig. 10, MEF and pMEF allocate more resources to $\mathcal{I}_{hard}$ systems than IP and $\mathcal{I}_{easy}$. As a result, they are able to balance out the higher task criticality of those sub-systems through their awareness of nMSE displayed in Fig. 14.

VIII. CONCLUSION AND FINAL REMARKS

AoI has been used for remote monitoring and control scenarios to quantify information freshness. However, in a network of heterogeneous control applications, providing freshness may not guarantee optimal performance due to diversified system dynamics and task-criticalities. Hence, customization of the network through application-dependent metrics is beneficial for satisfying the heterogeneous demands of such systems.

In this work, we study the practical implementation of various customized MAC protocols that have been proposed for increased information freshness and control performance. In addition, we propose and implement a new task-oriented contention-free protocol that considers the quality of estimation at the monitor and employs the estimation error as a metric for resource scheduling. Through real-world measurements using SDRs, we show that our proposed solution outperforms the selected existing strategies w.r.t. control and estimation performance. Moreover, we propose a new metric called the normalized mean squared error, a modified version of the previously proposed age-dependent MSE for NCSs. We demonstrate its applicability as a scheduling metric when the control loops are of heterogeneous types and dimensions. Our results reveal the high potential in cross-layer protocol design for task-oriented communications and networked control.

We expect the era of semantic communications to bring various research domains together, implying a convergence of multiple layers of the communication stack. Particularly in task-oriented communications, lower layers are expected to be aware of the information content and track the semantics of data, such as AoI and value of information. However, when it

comes to practical deployment, it might be challenging to execute decision-making due to a lack of relevant information. For instance, the implementation of a distributed AoI-based MAC protocol implies remote tracking of AoI within the data link layer of the source, although AoI is an application layer metric defined at the receiver. Furthermore, to identify the value of information within lower layers, certain knowledge about its content, context, and communication purpose is crucial. This introduces various new challenges and requires system-wide flexibility and programmability of the communication stack.

The system research is currently lacking behind theory due to the limited availability of easily programmable platforms. Additionally, the increased complexity and design challenges hidden prior to deployment constitute a barrier to practical implementation. With this work, we aim to encourage researchers towards a tighter integration of practice into theory and vice versa. Moreover, we intend to provide initial design considerations and insights into customized protocol implementation for AoI and NCSs communities.

REFERENCES

[1] W. Saad, M. Bennis, and M. Chen, "A vision of 6G wireless systems: Applications, trends, technologies, and open research problems," *IEEE Netw.*, vol. 34, no. 3, pp. 134–142, May/Jun. 2020.

[2] A. Mostaani, T. X. Vu, S. Krishna Sharma, Q. Liao, and S. Chatzinotas, "Task-oriented communication system design in cyber-physical systems: A survey on theory and applications," 2021, *arXiv:2102.07166*.

[3] S. Kaul, R. Yates, and M. Gruteser, "Real-time status: How often should one update?" in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 2731–2735.

[4] E. Uysal, O. Kaya, A. Ephremides, J. Gross, M. Codreanu, P. Popovski, M. Assaad, G. Liva, A. Munari, T. Soleymani, B. Soret, and K. Henrik Johansson, "Semantic communications in networked systems: A data significance perspective," 2021, *arXiv:2103.05391*.

[5] E. Calvanese Strinati and S. Barbarossa, "6G networks: Beyond Shannon towards semantic and goal-oriented communications," *Comput. Netw.*, vol. 190, May 2021, Art. no. 107930.

[6] O. Ayan, M. Vilgelm, M. Klügel, S. Hirche, and W. Kellerer, "Age-of-information vs. value-of-information scheduling for cellular networked control systems," in *Proc. 10th ACM/IEEE Int. Conf. Cyber-Physical Syst.*, Apr. 2019, pp. 109–117.

[7] I. Kadota, A. Sinha, E. Uysal-Biyikoglu, R. Singh, and E. Modiano, "Scheduling policies for minimizing age of information in broadcast wireless networks," *IEEE/ACM Trans. Netw.*, vol. 26, no. 6, pp. 2637–2650, Dec. 2018.

[8] *GNU Radio*. Accessed: Oct. 28, 2020. [Online]. Available: https://www.gnuradio.org

[9] I. Kadota, E. Uysal-Biyikoglu, R. Singh, and E. Modiano, "Minimizing the age of information in broadcast wireless networks," in *Proc. 54th Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Sep. 2016, pp. 844–851.

[10] Y.-P. Hsu, E. Modiano, and L. Duan, "Scheduling algorithms for minimizing age of information in wireless broadcast networks with random arrivals: The no-buffer case," 2017, *arXiv:1712.07419*.

[11] Z. Fang, J. Wang, Y. Ren, Z. Han, H. V. Poor, and L. Hanzo, "Age of information in energy harvesting aided massive multiple access networks," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 5, pp. 1441–1456, May 2022.

[12] A. Maatouk, M. Assaad, and A. Ephremides, "The age of updates in a simple relay network," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Nov. 2018, pp. 1–5.

[13] O. Vikhrova, F. Chiariotti, B. Soret, G. Araniti, A. Molinaro, and P. Popovski, "Age of information in multi-hop networks with priorities," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2020, pp. 1–6.

[14] R. Talak, S. Karaman, and E. Modiano, "Minimizing age-of-information in multi-hop wireless networks," in *Proc. 55th Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Oct. 2017, pp. 486–493.

[15] V. Tripathi, R. Talak, and E. Modiano, "Information freshness in multihop wireless networks," *IEEE/ACM Trans. Netw.*, pp. 1–16, 2022.

[16] S. K. Kaul, R. D. Yates, and M. Gruteser, "Status updates through queues," in *Proc. 46th Annu. Conf. Inf. Sci. Syst. (CISS)*, Mar. 2012, pp. 1–6.

[17] M. Costa, M. Codreanu, and A. Ephremides, "Age of information with packet management," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2014, pp. 1583–1587.

[18] C. Kam, S. Kompella, G. D. Nguyen, J. E. Wieselthier, and A. Ephremides, "Controlling the age of information: Buffer size, deadline, and packet replacement," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Nov. 2016, pp. 301–306.

[19] A. Kosta, N. Pappas, A. Ephremides, and V. Angelakis, "Age and value of information: Non-linear age case," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2017, pp. 326–330.

[20] A. Maatouk, S. Kriouile, M. Assaad, and A. Ephremides, "The age of incorrect information: A new performance metric for status updates," *IEEE/ACM Trans. Netw.*, vol. 28, no. 5, pp. 2215–2228, Oct. 2020.

[21] A. Kosta, N. Pappas, A. Ephremides, and V. Angelakis, "The age of information in a discrete time queue: Stationary distribution and non-linear age mean analysis," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 5, pp. 1352–1364, May 2021.

[22] Z. Wang, M.-A. Badiu, and J. P. Coon, "Relationship between age and value of information for a noisy ornstein–uhlenbeck process," *Entropy*, vol. 23, no. 8, p. 940, 2021.

[23] P. Zou and S. Subramaniam, "On the value of information in status update systems," *Entropy*, vol. 24, no. 4, p. 449, Mar. 2022.

[24] M. Noroozi and M. Fidler, "Age- and deviation-of-information of time-triggered and event-triggered systems," 2022, *arXiv:2206.01428*.

[25] O. Ayan, A. Ephremides, and W. Kellerer, "Age of information: An indirect way to improve control system performance," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, May 2021, pp. 1–7.

[26] J. P. Champati, M. H. Mamduhi, K. H. Johansson, and J. Gross, "Performance characterization using AoI in a single-loop networked control system," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Apr. 2019, pp. 197–203.

[27] G. C. Walsh and H. Ye, "Scheduling of networked control systems," *IEEE Control Syst. Mag.*, vol. 21, no. 1, pp. 57–65, Feb. 2001.

[28] C. Sonmez, S. Baghaee, A. Ergisi, and E. Uysal-Biyikoglu, "Age-of-information in practice: Status age measured over TCP/IP connections through WiFi, Ethernet and LTE," in *Proc. IEEE Int. Black Sea Conf. Commun. Netw. (BlackSeaCom)*, Jun. 2018, pp. 1–5.

[29] B. Barakat, H. Yassine, S. Keates, I. Wassell, and K. Arshad, "How to measure the average and peak age of information in real networks?" in *Proc. 25th Eur. Wireless Conf.*, May 2019, pp. 1–5.

[30] H. B. Beytur, S. Baghaee, and E. Uysal, "Measuring age of information on real-life connections," in *Proc. 27th Signal Process. Commun. Appl. Conf. (SIU)*, Apr. 2019, pp. 1–4.

[31] T. Shreedhar, S. K. Kaul, and R. D. Yates, "An age control transport protocol for delivering fresh updates in the Internet-of-Things," in *Proc. IEEE 20th Int. Symp. 'World Wireless, Mobile Multimedia Netw.' (WoWMoM)*, Jun. 2019, pp. 1–7.

[32] I. Kadota, M. S. Rahman, and E. Modiano, "WiFresh: Age-of-information from theory to implementation," in *Proc. Int. Conf. Comput. Commun. Netw. (ICCCN)*, Jul. 2021, pp. 1–11.

[33] Z. Han, J. Liang, Y. Gu, and H. Chen, "Software-defined radio implementation of age-of-information-oriented random access," in *Proc. 46th Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Oct. 2020, pp. 4374–4379.

[34] O. Ayan, H. Y. Ozkan, and W. Kellerer, "An experimental framework for age of information and networked control via software-defined radios," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2021, pp. 1–6.

[35] T. K. Oğuz, E. T. Ceran, E. Uysal, and T. Girici, "Implementation and evaluation of age-aware downlink scheduling policies in push-based and pull-based communication," *Entropy*, vol. 24, no. 5, p. 673, May 2022.

[36] K. J. Åström and R. M. Murray, *Feedback Systems: An Introduction for Scientists Engineers*. Princeton, NJ, USA: Princeton Univ. Press, 2008.

[37] D. Maity, M. H. Mamduhi, S. Hirche, and K. H. Johansson, "Optimal LQG control of networked systems under traffic-correlated delay and dropout," *IEEE Control Syst. Lett.*, vol. 6, pp. 1280–1285, 2022.

[38] N. Abramson, "The ALOHA system: Another alternative for computer communications," in *Proc. Fall Joint Comput. Conf.*, vol. 1970, Nov. 1970, pp. 281–285.

[39] L. G. Roberts, "ALOHA packet system with and without slots and capture," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 5, no. 2, pp. 28–42, 1975.

[40] H. Chen, Y. Gu, and S.-C. Liew, "Age-of-information dependent random access for massive IoT networks," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Jul. 2020, pp. 930–935.
[41] X. Zheng, S. Zhou, Z. Jiang, and Z. Niu, "Closed-form analysis of non-linear age of information in status updates with an energy harvesting transmitter," *IEEE Trans. Wireless Commun.*, vol. 18, no. 8, pp. 4129–4142, Aug. 2019.
[42] Y. Sun, E. Uysal-Biyikoglu, R. D. Yates, C. E. Koksal, and N. B. Shroff, "Update or wait: How to keep your data fresh," *IEEE Trans. Inf. Theory*, vol. 63, no. 11, pp. 7492–7508, Nov. 2017.

ONUR AYAN (Graduate Student Member, IEEE) received the B.Sc. degree in electrical engineering and the M.Sc. degree from the Technical University of Munich (TUM), Munich, Germany, in 2013 and 2017, respectively. In January 2017, he joined the Chair of Communication Networks, where he is currently working as a Research and Teaching Associate. His research interests include design and evaluation of wireless networked control systems, cyber-physical systems, and semantics of information.

POLINA KUTSEVOL (Graduate Student Member, IEEE) received the B.Sc. degree in applied mathematics and physics from the Moscow Institute of Physics and Technology, Moscow, Russia, in 2019, and the M.Sc. degree in communication engineering from the Technical University of Munich, Munich, Germany, in 2021, where she is currently pursuing the Ph.D. degree with the Chair of Communication Networks. Her current research interests include resource management for wireless and mobile communication networks, cyber-physical systems, and networked control systems.

HASAN YAĞIZ ÖZKAN received the bachelor's and master's degrees in electrical engineering and information technology from the Technical University of Munich, Germany, where he is currently pursuing the Ph.D. degree with the Chair of Communication Networks. His research interests include dependability analysis of softwarized networks, and classification and characterization of threats in different software modules.

WOLFGANG KELLERER (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from the Technical University of Munich (TUM), Germany, in 2002. He is currently a Full Professor with TUM, heading the Chair of Communication Networks at the School of Computation, Information and Technology. He was a Visiting Researcher at the Information Systems Laboratory, Stanford University, CA, USA, in 2001. Prior to joining TUM, he pursued an industrial career being for over ten years with NTT DOCOMO's European Research Laboratories. He was the Director of the Infrastructure Research Department, where he led various projects for wireless communication and mobile networking contributing to research and standardization of LTE-A and 5G technologies. In 2015, he was awarded with an ERC Consolidator Grant from the European Commission for his research on flexibility in communication networks. He currently serves as an Associate Editor for IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT and the Area Editor for Network Virtualization for IEEE COMMUNICATIONS SURVEYS AND TUTORIALS.