This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2909198, IEEE Internet of Things Journal



# Control Aware Radio Resource Allocation in Low Latency Wireless Control Systems

Mark Eisen* Mohammad M. Rashid† Konstantinos Gatsis*

Dave Cavalcanti† Nageen Himayat† Alejandro Ribeiro*

***Abstract*—We consider the problem of allocating radio resources over wireless communication links to control a series of independent wireless control systems. Low-latency transmissions are necessary in enabling time-sensitive control systems with high sampling rates to operate over wireless links. Enabling low-latency through fast data rates comes at the cost of reliability in the form of higher packet error rates due to channel noise. However, the impact of such communication link errors on the control system performance depends dynamically on the control system state. We propose a novel control-aware communication design to the low-latency resource allocation problem. In our proposed method, we incorporate both control and channel state information in scheduling transmissions across time slots, frequency bands, and data rates using the next-generation Wi-Fi scheduling architecture. Control systems that are closer to instability or further from a desired range in a given control cycle are given higher packet delivery rate targets to meet. Rather than a simple priority ranking, we derive precise adaptive packet error rate targets for each system needed to satisfy control-specific performance requirements. We use these adaptive rate targets to make scheduling decisions that reduce total transmission time. The resulting Control-Aware Low Latency Scheduling (CALLS) method is tested in numerous simulation experiments that demonstrate its effectiveness in meeting control-based goals under tight latency constraints relative to control-agnostic scheduling.**

***Index Terms*—** wireless control, low-latency, codesign, IEEE 802.11ax

## I. Introduction

The Internet-of-Things (IoT) promises enhanced modes of interaction with the physical world through the deployment of large numbers of sensing and control devices. Relative to conventional communication systems, the deployment of a control system over a communication network increases the sensitivity to packet loss and latency. This is not a major consideration if we rely on wired networks that can simultaneously achieve very low latency and ultra high reliability. However, the cost of installing and maintaining a wired network poses significant challenges [1], [2] that motivate the use of wireless communications. Consequently, there has been great effort in the design of wireless control systems that can achieve high performance in terms of reliability and latency [3], [4]. While this effort is widespread in its range of applications it is of note that wireless control systems hold promise in streamlining industrial control [1], [3], [5]–[7]—e.g. factory automation [8], [9].

The primary challenge in designing this ultra reliable low latency communications (URLLC) systems is the tradeoff between reliability and latency. To achieve ultra-high reliability we need significant protection against packet losses. This can be achieved by increasing packet length, thereby increasing latency, or by increasing bandwidth consumption. Such a tradeoff between latency and reliability is present in any communication medium but it is exacerbated in wireless communications because of fading and shadowing effects. The physical properties of a wireless channel impose inherent limitations in achieving both ultra high reliability and low latency. Such a mismatch has lead to the proposal of several radio resource allocation schemes have been proposed to improve the management of the latency reliability tradeoff in wireless systems [10]. This include seminal works on the design of delay-aware schedulers for communication systems in general [11], [12] and wireless control systems in particular [13]. The exploitation of spatial and frequency diversity deserves particular mention as a technique that is increasingly recognized as a necessary component of URLLC [6], [14], [15].

In this paper we propose an alternative approach in which we adapt the communication reliability to the dynamics and state of the plant. We expect this to provide significant advantages because high communication reliability is not a strict requirement of control reliability. In fact, control loops can, in general, drop packets with small impact if the plant is not close to unsafe states. The successful transmission of a packet becomes crucial only when in these unsafe regions. In adapting reliability to the state of the plant we expect that the resources that are saved with plants in safe states are available to achieve high reliability with plants close to unsafe states. Our specific contribution is to develop a control-aware scheduling protocol designed to enable larger scale low latency systems. This is done through the mathematical formulation of the control system design goal in the form of a Lyapunov function that ensures stability of the control system. This formulation naturally induces a bound on the packet delivery rate each control system needs to achieve to meet the control-based goal. Such packet delivery rates depend upon current control and channel states and thus dynamically change over the course of the system life time. In particular, we use IEEE 802.11ax

Supported by the Intel Science and Technology Center for Wireless Autonomous Systems. The authors are with the (*)Department of Electrical and Systems Engineering, University of Pennsylvania and (†)Wireless Communications Research, Intel Corporation. Email: maeisen@seas.upenn.edu, mamun.rashid@intel.com, kgatsis@seas.upenn.edu, dave.cavalcanti@intel.com, nageen.himayat@intel.com, aribeiro@seas.upenn.edu. 

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2909198, IEEE Internet of Things Journal



WiFi [16] to allocate bandwidth and data rates to reduce total transmission times. As these control-based reliability targets may be significantly lower than traditional, high reliability communication demands, the proposed method is better suited to find scheduling configurations that can meet strict latency requirements imposed by the physical system.

We remark that in the context of wireless control systems, there have been a range of works that incorporates control system information in the networking and communication policies. For example, control system stability under fixed periodic protocols, e.g. round-robin, can be analyzed—see, e.g., [17]–[20]. Periodic sequences leading to stability [21], controllability and observability [22], or optimizing control objectives [23]–[25] have been proposed. More sophisticated schedulers do not rely on a predefined sequence but try to dynamically access the communication medium at each step. Initial approaches abstract control performance requirements in the time/frequency domain, e.g., how often a task needs resource access, employing algorithms from real-time scheduling theory [26], [27]. More recent scheduling approaches often depend on the current control system states, i.e., informally the subsystem with the largest state discrepancy is scheduled to communicate—see, e.g., [19], [28]–[31]. Alternatively scheduling can take into account current wireless channel conditions opportunistically to meet target control system reliability requirements [32]. None of these approaches, however, are explicitly designed for low-latency communication systems, which is the subject of our work and a key contribution with respect to previous approaches.

The paper is organized as follows. We formulate the wireless control system in which state information is communicated to the control over a wireless channel. Due to the potential for random packet drops, this is modeled as a switched dynamical system (Section II). A Lyapunov function is used to evaluate the stability of the control state, and the uncertainty in this measurement grows the more consecutive packets are lost for a particular system. We then discuss the scheduling parameters of the IEEE 802.11ax communication model (Section II-A). From there, we derive a mathematical formulation of the optimal scheduling problem (Section III). This can be formulation by minimizing a control cost with an explicitly latency constraint (Section III-A) or minimizing transmission time with an explicit control performance constraint (Section III-B).

Using this formulation, we develop the control-aware low latency scheduling (CALLS) method (Section IV). The CALLS method uses current control states and channel conditions to derive dynamic packet success rates for each user (Section IV-A). In this way, control systems that are closest to instability will be given priority in the scheduling so that they may close their control loops. The scheduling procedure consists of a random user selection procedure to reduce the number of required PPDUs that incur significant overhead (Section IV-B), followed by an assignment-method based scheduling of selected users to minimize total transmission time (Section IV-C). The performance of the CALLS method is analyzed in a series of simulation experiments in which its performance is compared against a control-agnostic procedure (Section V). We demonstrate in numerous control systems that the control-aware, adaptive reliability approach may support more users than the alternative and achieve more robust overall performance.

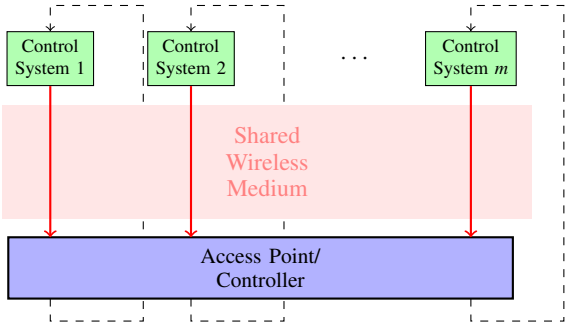


Figure 1: Wireless control system with $m$ independent systems. Each system contains a sensor that measure state information, which is transmitted to the controller over a wireless channel. The state information is used by the controller to determine control policies for each of the systems.The communication is assumed to be wireless in the uplink and ideal in the downlink.

## II. Wireless Control Sysyem

Consider a system of $m$ independent linear control systems, or devices, where each system $i = 1, \dots, m$ maintains a state variable $\mathbf{x}_i \in \mathbb{R}^p$. The dynamics are discretized so that the state evolves over time index $k$. Applying an input $\mathbf{u}_{i,k} \in \mathbb{R}^q$ causes the state and output to evolve based on the discrete-time state space equations,

$$\mathbf{x}_{i,k+1} = \mathbf{A}_i \mathbf{x}_{i,k} + \mathbf{B}_i \mathbf{u}_{i,k} + \mathbf{w}_k \tag{1}$$

where $\mathbf{A}_i \in \mathbb{R}^{p \times p}$ and $\mathbf{B}_i \in \mathbb{R}^{p \times q}$ are matrices that define the system dynamics, and $\mathbf{w}_k \in \mathbb{R}^p$ is Gaussian noise with covariance $\mathbf{W}_i$ that captures the errors in the linear model (due to, e.g., unknown dynamics or from linearizion of non-linear dynamics). We further assume the state transition matrix $\mathbf{A}_i$ is on its own unstable, i.e. has at least one eigenvalue greater than 1. This is to say that, without an input, the dynamics will drive the state $\mathbf{x}_{i,k} \to \infty$ as $k \to \infty$.

In the wireless control system model presented in Figure 1. Each system is closed over a wireless medium, over which the sensor located at the control system sends state information to the controller located at a wireless access point (AP) shared among all systems. Using the state information $\mathbf{x}_{i,k}$ received from device $i$ at time $k$, the controller determines the input $\mathbf{u}_{i,k}$ to be applied. We stress in Figure 1 we restrict our attention to the wireless communications at the sensing, or "uplink", while the control actuation, or "downlink, is assumed to occur over an ideal channel. We point out that while a more complete model may include packet drops in the downlink, in practice the more significant latency overhead occurs in the uplink. We therefore keep this simpler model for mathematical coherence. In low-latency applications, a high state sampling rate is required be able to adapt to the fast-moving dynamics This subsequently places a tight restriction on the latency in the wireless transmission, so as to avoid losing sampled state information. This specific latency requirement between the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2909198, IEEE Internet of Things Journal



sensor and AP we denote by $\tau_{\max}$, and is often considered to be in the order of milliseconds.

Because the control loop in Figure 1 is closed over a wireless channel, there exists a possibility at each cycle $k$ that the transmission fails and state information is not received by the controller. We refer to this as the "open-loop' configuration; when state information is received, the system operates in "closed-loop." As such, it is necessary to define the system dynamics in both configurations. Consider a generic linear control, in which the input being determined as $\mathbf{u}_{i,k} = \mathbf{K}_i \mathbf{x}_{i,k}$ for some matrix $\mathbf{K}_i \in \mathbb{R}^{q \times p}$. Many common control policies indeed can be formulated in such a manner, such as LQR control. In general, this matrix $\mathbf{K}$ is chosen such as that the closed loop dynamic matrix $\mathbf{A} + \mathbf{BK}$ is stable, i.e. has all eigenvalues less that 1. Thus, application of this control over time will drive the state $\mathbf{x}_{i,k} \to 0$ as $k \to \infty$. We assume that this choice of $\mathbf{K}$ is given—in other words, the controller is pre-designed with respect to ideal closed loop behavior. As the controller does not always have access to state information, we alternatively consider the estimate of state information of device $i$ known to the controller at time $k$ as

$$\hat{\mathbf{x}}_{i,k}^{(l_i)} := (\mathbf{A}_i + \mathbf{B}_i \mathbf{K}_i)^{l_i} \mathbf{x}_{i,k-l_i}, \tag{2}$$

where $k - l_i \geq k - 1$ is the last time instance in which control system $i$ was closed. There are two important things to note in (2). First, this is the estimated state *before* a transmission has been attempted at time $k$; hence, $l_i = 1$ when state information was received at the previous time. Second, observe that in (2) we assume that the AP/controller has knowledge of the dynamics $\mathbf{A}_i$ and $\mathbf{B}_i$, as well as the linear control matrix $\mathbf{K}_i$. Any gap in this knowledge of dynamics is captured in the noise $\mathbf{w}_k$ in the actual dynamics in (35). Note that the estimated state (2) is used in place of the true state in both the determination of the control *and* the radio resource allocation decisions as discussed later in this paper.

At time $k$, if the state information is received, the controller can apply the input $\mathbf{u}_{i,k} = \mathbf{K}_i \mathbf{x}_{i,k}$ exactly, otherwise it applies an input using the estimated state, i.e. $\mathbf{u}_{i,k} = \mathbf{K}_i \hat{\mathbf{x}}_{i,k}$. Thus, in place of (35), we obtain the following switched system dynamics for $\mathbf{x}_{i,k}$ as

$$\mathbf{x}_{i,k+1} = \begin{cases} (\mathbf{A}_i + \mathbf{B}_i \mathbf{K}_i)\mathbf{x}_{i,k} + \mathbf{w}_k, & \text{in closed-loop,} \\ \mathbf{A}_i \mathbf{x}_{i,k} + \mathbf{B}_i \mathbf{K}_i \hat{\mathbf{x}}_{i,k}^{(l_i)} + \mathbf{w}_k, & \text{in open-loop.} \end{cases} \tag{3}$$

The transmission counter $l_i$ is updated at time $k$ as

$$l_i \leftarrow \begin{cases} 1, & \text{in closed-loop,} \\ l_1 + 1, & \text{in open-loop.} \end{cases} \tag{4}$$

Observe in (3) that, when the system operates in open loop, the control is not applied relative to the current state $\mathbf{x}_{i,t}$ but on the estimated state $\hat{\mathbf{x}}_{i,k}^{(l_i)}$, which indeed may not be close to the true state. In this case, the state may not be driven to zero as in the closed-loop configuration. To see the effect of operating in open loop for many successive iterations, we can

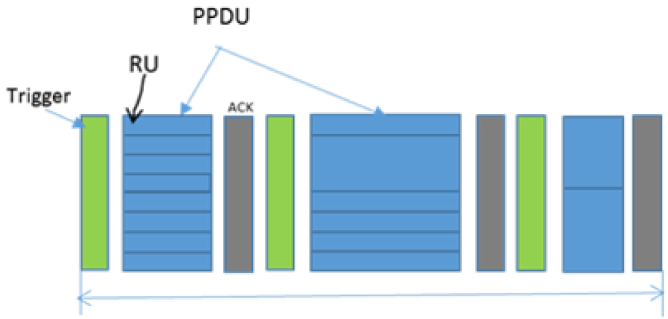


Figure 2: Multiplexing of frequencies (RU) and time (PPDU) in IEEE 802.11ax transmission window (formally referred as Transmission Opportunity or TXOP in the standard. The total transmission time is the time of all PPDUs, including the overhead of trigger frames (TF) and acknowledgments.

write the error between the true and estimated state as

$$\mathbf{e}_{i,k} := \mathbf{x}_{i,k} - \hat{\mathbf{x}}_{i,k}^{(l_i)} = \sum_{j=0}^{l_i - 1} \mathbf{A}_i^j \mathbf{w}_{i,k-j-1}. \tag{5}$$

In (5), it can be seen that as $l_i$ grows, the error $\mathbf{e}_{i,k}$ grows with the accumulation of the noise present in the actual state but not considered in the estimated state. Thus, if $l_i$ is large and $\mathbf{w}_{i,k}$ is large (i.e., high variance), this error will become large as well.

To conclude the development of the wireless control formulation, we define a quadratic Lyapunov function $L(\mathbf{x}) := \mathbf{x}^T \mathbf{P} \mathbf{x}$ for some positive definite $\mathbf{P} \in \mathbb{R}^{p \times p}$ that measures the performance of the system as a function of the state. Because the scheduler only has access to estimated state info, we consider the expected value of $L(\mathbf{x})$ given the state estimate, which can be found via (5) as

$$\mathbb{E}[L(\mathbf{x}_{i,k}) \mid \hat{\mathbf{x}}_{i,k}^{(l_i)}] \tag{6}$$
$$= (\hat{\mathbf{x}}_{i,k}^{(l_i)})^T \mathbf{P} (\hat{\mathbf{x}}_{i,k}^{(l_i)}) + \sum_{j=0}^{l_i - 1} \mathrm{Tr}[(\mathbf{A}_i^T \mathbf{P}^{\frac{1}{j}} \mathbf{A}_i)^j \mathbf{W}_i].$$

Thus, the control-specific goal is to keep $\mathbb{E}[L(\mathbf{x}_{i,k}) \mid \hat{\mathbf{x}}_{i,k}^{(l_i)}]$ within acceptable bounds for each system $i$. We now proceed to discuss the wireless communication model that determines the resource allocations necessary to close the loop.

### A. IEEE 802.11ax communication model

We consider the communication model provided in the next-generation Wi-Fi standard IEEE 802.11ax. While 3GPP wireless systems such as LTE [33] or the next generation 5G [34] can also be considered as alternate communication models, most factory floors are already equipped with Wi-Fi connectivity and, moreover, Wi-Fi can operate in the unlicensed band. It is generally considered to be cost-effective to operate and maintain.

Traditional Wi-Fi systems rely only on contention-based channel access and may introduce high or variable latency in congested or dense deployment scenarios even in a fully managed Wi-Fi network, which is typically available in

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2909198, IEEE Internet of Things Journal



industrial control and automation scenarios. To address the problems with dense deployment, the draft 802.11ax amendment has defined scheduling capability for Wi-Fi access points (APs). Wi-Fi devices can now be scheduled for accessing the channel in addition to the traditional contention-based channel access. Such scheduled access enables more controlled and deterministic behavior in the Wi-Fi networks. Within each transmission window (formally referred as transmission opportunity or TXOP in the standard), the AP may schedule devices through both frequency and time division multiplexing using the multi-user (MU) OFDMA technique. This to say that devices can be slotted in various frequency bands—formally called resource units (RUs)—and in different timed transmission slots—formally called PPDUs. An example of the multiplexing of devices across time and frequency is demonstrated in Figure 2. The AP additionally sends a trigger frame (TF) indicating which devices should transmit data in the current TXOP and the time/frequency resources these triggered devices should use in their transmissions.

To state this model formally, the scheduling parameters assigned by the AP to each device consist of a frequency-slotted RU, time-slotted PPDU, and an associated modulation and coding scheme (MCS) to determine the transmission format. The transmission power is assumed to be fixed and equally divided amongst all devices. We define the following notations to formulate these parameters. To specify an RU, we first notate by $f^1, f^2, \ldots, f^b$, where $n$ is the number of discrete frequency bands of fixed bandwidth (typically 2 MHz) in which a device can transmit; in a 20MHz channel, for example, there are $n = 10$ such bands. For each device, we then define a set of binary variables $\varsigma_i^j \in \{0, 1\}$ if device $i$ transmits in band $f^j$ and collect all such variables for device $i$ in $\boldsymbol{\varsigma}_i = [\varsigma_i^1; \ldots; \varsigma_i^b] \in \{0, 1\}^b$ and for all devices in $\boldsymbol{\Sigma} := [\boldsymbol{\varsigma}_i, \ldots, \boldsymbol{\varsigma}_m] \in \{0, 1\}^{b \times m}$. A device may transmit in bands in certain multiples of 2MHz as well, which would be notated as, e.g. $\boldsymbol{\varsigma}_i = [1; 1; 0; \ldots; 0]$ for transmission in an RU of size 4MHz. Note, however, that allowable RU's contain only sizes of certain multiples of 2MHz—namely, 2MHz, 4MHz, 8MHz, and 20MHz in the 802.11ax standard. Furthermore, it is only permissible to transmit in *adjacent* bands, e.g. $f^j$ and $f^{j+1}$. We therefore define the set $\mathcal{S} \subset \{0, 1\}^b$ as the set of binary vectors that define permissible RUs and consider only $\boldsymbol{\varsigma}_i \in \mathcal{S}$ for all devices $i$. Finally, note that the RU assignment $\mathbf{0} \in \mathcal{S}$ signifies a device does not transmit in this particular transmission window.

To specify the PPDU of all scheduled devices, we define for device $i$ a positive integer value $\alpha_i \in \mathbb{Z}_{++}$ that denotes the PPDU slot in which it transmits and collect such variables for all devices in $\boldsymbol{\alpha} = [\alpha_1; \ldots; \alpha_m] \in \mathbb{Z}_{++}^m$. Likewise, device $i$ is given an MCS $\mu_i$ from the discrete space $\mathcal{M} = \{0, 1, 2, \ldots, 10\}$. The MCS in particular defines a pair of modulation scheme and coding rate that subsequently determine both the data rate and packet error rate of the transmission. The allowable MCS settings provided in 802.11ax are provided in Table I. Finally, we notate by $\mathbf{h}_i := [h_i^1; h_i^2; \ldots; h_i^b] \in \mathbb{R}_+^b$ a set of channel states experienced by device $i$, where $h_i^j$ is the gain of a wireless fading channel in frequency band $f^j$. We assume that channel conditions are constant within a single TXOP, i.e. do not vary across PPDUs.

| $\mu$ | Modulation type | Coding rate | Data rate (Mb/s) |
|---|---|---|---|
| 0 | BPSK | 1/2 | 4 |
| 1 | QPSK | 1/2 | 16 |
| 2 | QPSK | 3/4 | 24 |
| 3 | 16-QAM | 1/2 | 33 |
| 4 | 16-QAM | 3/4 | 49 |
| 5 | 64-QAM | 2/3 | 65 |
| 6 | 64-QAM | 3/4 | 73 |
| 7 | 64-QAM | 5/6 | 81 |
| 8 | 256-QAM | 3/4 | 98 |
| 9 | 256-QAM | 5/6 | 108 |
| 10 | 1024-QAM | 3/4 | 122 |

Table I: Data rates for MCS configurations in IEEE 802.11ax for 20MHz channel. The modulation type and coding rate in the first 2 columns together specify a PDR function $q(\boldsymbol{\mu}, \boldsymbol{\varsigma})$ for RU $\boldsymbol{\varsigma}$. The data rate in the third column specifies the associated transmission time $\tau(\mu, \boldsymbol{\varsigma})$.

We now proceed to define two functions that describe the wireless communications over the channel. Firstly, we define a function $q : \mathbb{R}_+^b \times \mathcal{M} \times \mathcal{S} \to [0, 1]$ which, given a set of channel conditions $\mathbf{h}$, MCS $\mu$, and RU $\boldsymbol{\varsigma}$, returns the probability of successful transmission, otherwise called packet delivery rate (PDR). Furthermore, define by $\tau : \mathcal{M} \times \mathcal{S} \to \mathbb{R}_+$ a function that, given an MCS $\mu$ and RU $\boldsymbol{\varsigma}$, returns the maximum time taken for a single transmission attempt. Assuming a fixed packet size, such a function can be determined from the data rates associated with each MCS in Table I. Observe that all functions just defined are determined independent of the PPDU slot the transmission takes place in, while transmission time is also independent of the channel state. Because a PPDU cannot finish until all transmissions within the PPDU have been completed, the total transmission time of a single PPDU $s$ is the maximum transmission time taken by all devices within that time slot. We define the transmission time of PPDU slot $s$ as

$$\hat{\tau}(\boldsymbol{\Sigma}, \boldsymbol{\mu}, \boldsymbol{\alpha}, s) := \max_{i:\alpha_i = s} \tau(\mu_i, \boldsymbol{\varsigma}_i) + \tau_0(\boldsymbol{\alpha}, s), \tag{7}$$

where $\tau_0 : \mathbb{Z}_{++}^m \times \mathbb{Z}_{++} \to \mathbb{R}_+$ is a function that specifies the communication overhead of PPDU $s$. This overhead may consist of, e.g., the time required to send TFs to scheduled users, as seen in Figure 2.

## III. Optimal Control Aware Scheduling

Using the communication model of 802.11ax just outlined and the control-based Lyapunov metric of (6), we can formulate an optimization problem that characterizes the exact optimal scheduling of transmissions with a transmission window to maximize control performance. The optimal scheduling and allocation selects the set of RUs $\boldsymbol{\Sigma}$. MCS $\boldsymbol{\mu}$, and PPDUs $\boldsymbol{\alpha}$ for all devices–which in effect fully determine the schedule—such to minimize a cost subject to scheduling design and feasibility constraints. In particular, we discuss two related, alternative formulations of the low-latency scheduling problem.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2909198, IEEE Internet of Things Journal



### A. Latency-constrained scheduling

In the latency-constrained formulation, we are interested in minimizing a common control cost subject to strict latency requirements. In particular, in the low-latency setting we set a bound $\tau_{\max}$ on the total transmission time across all PPDUs in a TXOP. This constraint is relevant in design of MAC-layer protocols that set strict limits on transmission times. In addition, the RU and PPDU allocation across devices must be feasible, i.e., two devices cannot be transmitting in the same frequency band in the same PPDU.

Recall the PDR function $q(\mathbf{h}, \mu, \boldsymbol{\varsigma})$ and consider that this can alternatively be interpreted as the probability of closing the control loop under certain channel conditions and scheduling parameters. From there, we can now write the expected Lyapunov value for at time $k+1$ given its current state $\mathbf{x}_{i,k}$, channel state $\mathbf{h}_{i,k}$, MCS $\mu_i$, and RU $\boldsymbol{\varsigma}_i$ using the expected cost in (6). By defining $\mathbf{x}^c_{i,k+1}$ and $\mathbf{x}^o_{i,k+1}$ as the closed loop and open loop states, respectively, as determined by the switched system in (3), this is written as

$$\begin{aligned} J_i(\hat{\mathbf{x}}^{(l_i)}_{i,k}, \mathbf{h}_{i,k}, \mu_i, \boldsymbol{\varsigma}_i) &:= \mathbb{E}(L(\mathbf{x}_{i,k+1}) \mid \hat{\mathbf{x}}^{(l_i)}_{i,k}, \mathbf{h}_{i,k}, \mu_i, \boldsymbol{\varsigma}_i) \\ &= (1 - q(\mathbf{h}_{i,k}, \mu_i, \boldsymbol{\varsigma}_i))\mathbb{E}L(\mathbf{x}^o_{i,k+1} \mid \hat{\mathbf{x}}^{(l_i)}_{i,k}) \\ &\quad + q(\mathbf{h}_{i,k}, \mu_i, \boldsymbol{\varsigma}_i)\mathbb{E}L(\mathbf{x}^c_{i,k+1} \mid \hat{\mathbf{x}}^{(l_i)}_{i,k}). \quad (8) \end{aligned}$$

For notational convenience, we collect all current estimated control states at time $k$ as $\hat{\mathbf{X}}_k := [\hat{\mathbf{x}}^{(l_1)}_{1,k}, \ldots, \hat{\mathbf{x}}^{(l_m)}_{m,k}]$ and channel states $\mathbf{H}_k = [\mathbf{h}_{1,k}, \ldots, \mathbf{h}_{m,k}]$. Now, define the total control cost, given the current states and scheduling parameters as some aggregation of the combined expected future Lyapunov costs across all devices, i.e.,

$$\begin{aligned} &\tilde{J}(\hat{\mathbf{X}}_k, \mathbf{H}_k, \boldsymbol{\mu}, \boldsymbol{\Sigma}) := \quad (9) \\ &\quad g(J_1(\hat{\mathbf{x}}^{(l_1)}_{1,k}, \mathbf{h}_{1,k}, \mu_1, \boldsymbol{\varsigma}_1), \ldots, J_m(\hat{\mathbf{x}}^{(l_m)}_{m,k}, \mathbf{h}_{m,k}, \mu_m, \boldsymbol{\varsigma}_m)). \end{aligned}$$

Natural choices of the aggregation function $g(\cdot)$ are, for example, either the sum or maximum of its arguments.

The optimal scheduling at transmission time $k$ is formulated as the one which minimizes this cost $\tilde{J}$ while satisfying low-latency and feasibility requirements of the schedule, expressed formally with the following optimization problem.

$$[\boldsymbol{\Sigma}^*_k, \boldsymbol{\mu}^*_k, \boldsymbol{\alpha}^*_k] := \underset{\boldsymbol{\Sigma}, \boldsymbol{\mu}, \boldsymbol{\alpha}, S}{\operatorname{argmin}} \tilde{J}(\hat{\mathbf{X}}_k, \mathbf{H}_k, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \quad (10)$$

$$\text{s.t.} \sum_{i:\alpha_i = s} \varsigma^j_i \leq 1, \quad \forall j, s, \quad (11)$$

$$\sum_{s=1}^{S} \hat{\tau}(\boldsymbol{\Sigma}, \boldsymbol{\mu}, \boldsymbol{\alpha}, s) \leq \tau_{\max}, \quad (12)$$

$$1 \leq \alpha_i \leq S, \quad \forall i, \quad (13)$$

$$\boldsymbol{\varsigma}_i \in \mathcal{S}, \ \forall i, \quad \boldsymbol{\mu} \in \mathcal{M}^m, \quad \boldsymbol{\alpha} \in \mathbb{Z}^m_+, \quad S \in \mathbb{Z}_+. \quad (14)$$

The optimization problem in (10) provides a precise and instantaneous selection of frequency allocations between devices given their current control states $\hat{\mathbf{X}}_k$ and communication states $\mathbf{H}_k$. The constraints in (11)-(14) encode the following scheduling conditions. The constraint (11) ensures that for every PPDU $s$, there is only one device transmitting on a frequency slot $j$. In (12), we set the low-latency transmission time constraint in terms of the sum of all transmission times for each PPDU $s$. The constraint in (13) bounds each transmission slot by the total number of PPDU's $S$ while (14) constrains each variable to its respective feasible set. Note that $S$ is itself treated as an optimization variable in the above problem, so that the number of PPDUs may vary as needed.

Observe in the objective in (10) that, by minimizing an aggregate of local control costs, the devices with the highest cost $J_i$ as described by (8) will be given the most bandwidth or most favorable frequency bands to increase probability of successful transmission $q(\mathbf{h}_{i,k}, \mu_i, \boldsymbol{\varsigma}_i)$. This in effect increases the chances those devices will close their control loops and be driven towards a more favorable state. Likewise, a device who is experiencing very adverse channel conditions may not be allocated prime transmission slots to reserve such resources who have more favorable channel conditions. In this way, we say this is *control-aware* scheduling, as it considers both the control and channel states of the devices to determine optimal scheduling. However, we stress that the optimization problem described in (10)-(14) is by no means easy to solve. In fact, the optimization over multiple discrete variables makes this problem combinatorial in nature. In the following section, we discuss a practical reformulation of the problem above and develop heuristic methods to approximate the solutions in realistic low-latency wireless applications.

### B. Control-constrained scheduling

We reformulate the problem in (10)-(14) to an alternative formulation that more directly informs the control-aware, low-latency scheduling method to be developed. To do so, we introduce a *control-constrained* formulation, in which the Lyapunov decrease goals are presented as explicit requirement, i.e. constraints in the optimization problem. We are interested, then, in constraint of the form

$$\tilde{J}(\hat{\mathbf{X}}_k, \mathbf{H}_k, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \leq J_{\max}, \quad (15)$$

where $J_{\max}$ is a limiting term design to enforce desired system performance. Determining this constant is largely dependent on the particular application of interest, needs of the control systems, and also may be related to the choice aggregation function $g(\cdot)$ in (9). For example, $J_{\max}$ may represent a point at which control systems become volatile, unsafe, or unstable.

For the scheduling procedure developed in this paper, we focus on a particular formulation of the control constraint in (15) that constrains the expected future Lyapunov value of each system by a rate decrease of its current value. In particular the following rate-decrease condition for each device $i$,

$$J_i(\hat{\mathbf{x}}^{(l_i)}_{i,k}, \mathbf{h}_{i,k}, \mu_i, \boldsymbol{\varsigma}_i) \leq \rho_i \mathbb{E}[L(\mathbf{x}_{i,k}) \mid \hat{\mathbf{x}}^{(l_i)}_{i,k}] + c_i, \quad (16)$$

where $\rho_i \in (0, 1]$ is a decrease rate and $c_i \geq 0$ is a constant. Recall the definition of $J_i(\hat{\mathbf{x}}^{(l_i)}_{i,k}, \mathbf{h}_{i,k}, \mu_i, \boldsymbol{\varsigma}_i)$ in (8) as the expected Lyapunov value of time $k+1$ given its current estimate and scheduling $\mu_i, \boldsymbol{\varsigma}_i$. The constraint in (16) ensures the future Lyapunov cost will exhibit a decrease of at least a rate of $\rho_i$ for device $i$ in expectation. The constant $c_i$ is included to ensure

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2909198, IEEE Internet of Things Journal



this condition is satisfied by default if the state $\hat{\mathbf{x}}_{i,k}^{(l_i)}$ is already sufficiently small.

We formulate the control-constrained scheduling problem by substituting the latency constraint with the control constraint in (16), i.e.,

$$[\boldsymbol{\Sigma}_k^*, \boldsymbol{\mu}_k^*, \boldsymbol{\alpha}_k^*] := \underset{\boldsymbol{\Sigma},\boldsymbol{\mu},\boldsymbol{\alpha},S}{\operatorname{argmin}} \sum_{s=1}^{S} \hat{\tau}(\boldsymbol{\Sigma}, \boldsymbol{\mu}, \boldsymbol{\alpha}, s) \tag{17}$$

$$\text{s.t.} \sum_{i:\alpha_i=s} \varsigma_i^j \leq 1, \quad \forall j, s, \tag{18}$$

$$J_i(\hat{\mathbf{x}}_{i,k}^{(l_i)}, \mathbf{h}_{i,k}, \mu_i, \boldsymbol{\varsigma}_i) \leq \rho \mathbb{E}[L(\mathbf{x}_{i,k}) \mid \hat{\mathbf{x}}_{i,k}^{(l_i)}] + c_i \ \forall i, \tag{19}$$

$$1 \leq \alpha_i \leq S, \quad \forall i, \tag{20}$$

$$\boldsymbol{\varsigma}_i \in \mathcal{S}, \ \forall i, \quad \boldsymbol{\mu} \in \mathcal{M}^m, \quad \boldsymbol{\alpha} \in \mathbb{Z}_+^m, \quad S \in \mathbb{Z}_+. \tag{21}$$

Observe that the objective in (17) is now to minimize the total transmission time, rather than being forced as an explicit constraint. In this way, the optimization problem defined in (17)-(21) can be viewed as an alternative to the latency constrained problem in (10)-(14). Because the scheduling algorithm we develop in this paper requires the ability to quickly identify feasible solutions, we focus our attention on the control-constrained formulation in (17)-(21). Before presenting the details of the scheduling algorithm, we present a brief remark regarding the addition of "safety", or worst-case, constraints to either problem formulation.

**Remark 1** The control constraint in (19) is formulated to guarantee an average decrease of expected Lyapunov value by a rate of $\rho$. This is of interest to ensure the system states are driven to zero over time. However, in practical systems we may also be interested in protecting against worst-case behavior, e.g. entering an unsafe or unstable region. Consider a vector $\mathbf{b}_i \in \mathbb{R}^p$ as the boundary of safe operation of system $i$. A constraint that protects against exceeding this boundary can be written as

$$\mathbb{P}[|\mathbf{x}_{i,k+1}| \geq \mathbf{b}_i \mid \hat{\mathbf{x}}_{i,k}^{(l_i)}, \mathbf{h}_{i,k}, \mu_i, \boldsymbol{\varsigma}_i] \leq \delta, \tag{22}$$

where $\delta \in (0, 1)$ is small. The expression in (22) can be included as an additional constraint to either the latency-constrained or control-constrained scheduling problems previously discussed.

## IV. Control-Aware Low-Latency Scheduling (CALLS)

We develop a control-aware low-latency scheduling (CALLS) algorithm to approximately solve the control-constrained scheduling formulation in (17)-(21). Because this problem is combinatorial in nature, it is infeasible to solve exactly. Instead, we focus on a practical and efficient means of solving approximately. In particular, we identify sets of feasible points and use a heuristic approach towards minimizing the transmission time objective among the set of feasible points. Additionally, within the development of the CALLS method we identify and characterize new PDR requirements that are defined relative to the control system requirements; these are generally significantly less strict than the PDR requirements often considered in general high reliability communication systems without codesign. Overall, the CALLS method consists of (i) the derivation of adaptive control-aware PDR targets, (ii) a principled random selection of devices to schedule to reduce latency, and (iii) the use of assignment based methods to find a low-latency schedule. We discuss these three components in detail in the proceeding subsections.

### A. Control adaptive PDR

Due to the complexity of the scheduling problem in (17)-(21), we first focus our attention on identifying scheduling parameters $\{\boldsymbol{\Sigma}_k, \boldsymbol{\mu}_k, \boldsymbol{\alpha}_k\}$ that are feasible, i.e. satisfy the constraints in (18)-(21). In particular, the Lyapunov control constraint in (19) is of significant interest. Recall that the control cost function $J_i(\hat{\mathbf{x}}_{i,k}^{(l_i)}, \mathbf{h}_{i,k}, \mu_i, \boldsymbol{\varsigma}_i)$ is itself determined by the PDR $q(\mathbf{h}_{i,k}, \mu_i, \boldsymbol{\varsigma}_i)$, as per (8). Thus, the constraint in (19) can be seen as indirectly placing a constraint on the required PDR necessary to achieve a $\rho_i$-rate decrease in expectation. The equivalent condition on PDR $q(\mathbf{h}_{i,k}, \mu_i, \boldsymbol{\varsigma}_i)$ is presented in the following proposition.

**Proposition 1** *Consider the Lyapunov control constraint in* (19) *and the definition of* $J_i(\hat{\mathbf{x}}_{i,k}^{(l_i)}, \mathbf{h}_{i,k}, \mu_i, \boldsymbol{\varsigma}_i)$ *given in* (8)*. Define the closed-loop state transition matrix* $\mathbf{A}_i^c := \mathbf{A}_i + \mathbf{B}_i\mathbf{K}_i$ *and* $j$*-accumulated noise* $\omega_i^j := Tr[(\mathbf{A}_i^T\mathbf{P}^{1/j}\mathbf{A}_i)^j\mathbf{W}_i]$*. The control constraint in* (19) *is satisfied for device* $i$ *if and only if the following condition on PDR* $q(\mathbf{h}_{i,k}, \mu_i, \boldsymbol{\varsigma}_i)$ *holds,*

$$q(\mathbf{h}_{i,k}, \mu_i, \boldsymbol{\varsigma}_i) \geq \tilde{q}_i(\hat{\mathbf{x}}_{i,k}^{(l_i)}) := \tag{23}$$
$$\frac{1}{\Delta_i}\left[\left\|(\mathbf{A}_i^c - \rho_i\mathbf{I})\hat{\mathbf{x}}_{i,k}^{(l_i)}\right\|_{\mathbf{P}^{\frac{1}{2}}}^2 + (1-\rho_i)\sum_{j=0}^{l_i-1}\omega_i^j + \omega_i^{l_i} - c_i\right],$$

*where we have further defined the constant*

$$\Delta_i := \sum_{j=0}^{l_i-1}[\omega_i^{j+1} - Tr(\mathbf{A}_i^{cT}(\mathbf{A}_i^T\mathbf{P}^{1/j}\mathbf{A}_i)^j\mathbf{A}_i^c\mathbf{W}_i)]. \tag{24}$$

**Proof:** Consider the Lyapunov decrease constraint as written in (19). As the same logic holds for all $i$ and $k$, for ease of presentation we remove all subscripts when presenting the details of this proof. We further introduce the simpler notation $q := q(\mathbf{h}, \mu, \boldsymbol{\varsigma})$. Now, we may expand the left hand side of (19) be rewriting the definition in (8) as

$$J(\hat{\mathbf{x}}^{(l)}, \mathbf{h}, \mu, \boldsymbol{\varsigma}) = q\mathbb{E}_{\mathbf{w}}[L(\mathbf{A}_c\mathbf{x} + \mathbf{w})] + (1-q)\mathbb{E}_{\mathbf{w}}[L(\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{K}\hat{\mathbf{x}} + \mathbf{w})]. \tag{25}$$

Recall the definition of the quadratic Lyapunov function $L(\mathbf{x}) := \mathbf{x}^T\mathbf{P}\mathbf{x}$ for some positive definite $\mathbf{P}$. Further recall the relation $\mathbf{x} = \hat{\mathbf{x}} + \mathbf{e}$ as described by (5). Combining these, we expand the right hand size of (25) as

$$\begin{aligned} J(\hat{\mathbf{x}}^{(l)}, \mathbf{h}, \mu, \boldsymbol{\varsigma}) = \\ q\mathbb{E}_{\mathbf{w}}\left[\mathbf{A}_c\left(\hat{\mathbf{x}} + \mathbf{e}\right) + \mathbf{w}\right]^T\mathbf{P}\left[\mathbf{A}_c\left(\hat{\mathbf{x}} + \mathbf{e}\right) + \mathbf{w}\right] \\ + (1-q)\mathbb{E}_{\mathbf{w}}\left[\mathbf{A}_c\hat{\mathbf{x}} + \mathbf{A}\mathbf{e} + \mathbf{w}\right]^T\mathbf{P}\left[\mathbf{A}_c\hat{\mathbf{x}} + \mathbf{A}\mathbf{e} + \mathbf{w}\right]. \end{aligned} \tag{26}$$

To evaluate the expectations in (26), recall the random noise $\mathbf{w}$ follows a Gaussian distribution with zero mean and covariance

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2909198, IEEE Internet of Things Journal



$\mathbf{W}$. Thus, the expectation can be evaluated over $\mathbf{w}$ and expanded as

$$J(\hat{\mathbf{x}}^{(l)}, \mathbf{h}, \mu, \boldsymbol{\varsigma}) = \tag{27}$$
$$q\left[\|\mathbf{A}_c\hat{\mathbf{x}}\|^2_{\mathbf{P}^{\frac{1}{2}}} + \mathrm{Tr}(\mathbf{PW}) + \sum_{j=0}^{l-1}\mathrm{Tr}(\mathbf{A}_c(\mathbf{A}^T\mathbf{P}^{\frac{1}{j}}\mathbf{A})^j\mathbf{A}_c\mathbf{W})\right] +$$
$$(1-q)\left[\|\mathbf{A}_c\hat{\mathbf{x}}\|^2_{\mathbf{P}^{\frac{1}{2}}} + \mathrm{Tr}(\mathbf{PW}) + \sum_{j=1}^{l}\mathrm{Tr}((\mathbf{A}^T\mathbf{P}^{\frac{1}{j}}\mathbf{A})^j\mathbf{W})\right].$$

From here, we rearrange terms and substitute the notation $\omega^j := \mathrm{Tr}[(\mathbf{A}^T\mathbf{P}^{1/j}\mathbf{A})^j\mathbf{W}]$ to obtain that the control cost can be written as

$$J(\hat{\mathbf{x}}^{(l)}, \mathbf{h}, \mu, \boldsymbol{\varsigma}) = \left[\|\mathbf{A}_c\hat{\mathbf{x}}\|^2_{\mathbf{P}^{\frac{1}{2}}} + \mathrm{Tr}(\mathbf{PW}) + \sum_{j=1}^{l}\omega^j\right] \tag{28}$$
$$+ q\sum_{j=0}^{l-1}[\mathrm{Tr}(\mathbf{A}_c(\mathbf{A}^T\mathbf{P}^{\frac{1}{j}}\mathbf{A})^j\mathbf{A}_c\mathbf{W}) - \omega^{j+1}].$$

With (28), we have expanded the control cost in terms of the PDR $q$. Now, we return to the constraint in (19). Recall the expansion for $\mathbb{E}[L(\mathbf{x}) \mid \hat{\mathbf{x}}^{(l)}]$ via (6). By combining this with the expansion in (28), the terms in(19) can be rearranged to obtain the inequality in (23). ■

In Proposition 1 we establish a lower bound $\tilde{q}_i(\hat{\mathbf{x}}_{i,k}^{(l_i)})$ on the PDR of device $i$ that is dependent upon the current estimated state $\hat{\mathbf{x}}_{i,k}^{(l_i)}$ and system dynamics determined by $\mathbf{A}_i^c, \mathbf{A}_i$, and $\mathbf{W}^i$. We may note the following intuitions about the constraint in (23). The PDR condition naturally grows stricter as the bound $\tilde{q}_i(\hat{\mathbf{x}}_{i,k}^{(l_i)})$ defined on the right hand side of (23) gets larger. The first term on the right hand side reflects the current estimated channel state, and will become larger as the state gets larger. Similarly, the latter two terms on the right hand side together reflect the size of the noise that has accumulated by operating in open loop. When the noise variance $\mathbf{W}_i$ is high and when the last-update counter $l_i$ is large, these latter two noise terms will both be large. Thus, both the current magnitude of the control state and the growing uncertainty from infrequent transmissions together determine how large is the PDR requirement in (23).

We stress the value of the PDR condition in (23) is both in its adaptability to the control system state and dynamics, as well as its identification of precise target delivery rates that are necessary to keep the control systems moving towards stability on average. Depending on the particular system dynamics as described in (35), such PDR's may be, and often are considerably more lenient than the default target transmission success rates used in practical wireless systems (e.g. $q = 0.999$). Thus, through (23) we make a claim that, with knowledge of the control system dynamics and targeted *control performance*, we can effectively soften the targeted *communication performance*—or "reliability"— accordingly to something more easily obtained in low-latency constrained systems.

**Remark 2** It is worthwhile to note that by placing a stricter Lyapunov decrease constraint with smaller rate $\rho_i$ in (19), then the first term on the right hand side of (23) also grows larger and increases the necessary PDR. Generally, selecting a smaller $\rho$ will result in a faster convergence to stability but will require stricter communication requirements. In fact, we may use the inherent bound on the probability $q(\mathbf{h}_{i,k}, \mu_i, \boldsymbol{\varsigma}_i) \leq 1$ to find a lower bound on the Lyapunov decrease rate $\rho_i$ that can be feasibly obtained based upon current control state and system dynamics. This bound, however, may not be obtainable in practice due to the scheduling constraints. In practice, we select $\rho_i$ to be in the interval $[0.90, 0.1)$.

### *B. Selective scheduling*

We now proceed to describe the procedure with which we can find a set of feasible scheduling decisions $\{\boldsymbol{\Sigma}_k, \boldsymbol{\mu}_k, \boldsymbol{\alpha}_k\}$. To begin, we first consider a stochastically *selective scheduling* protocol, whereby we do not attempt to schedule every device at each transmission cycle, but instead select a subset to schedule a principled random manner. Define by $\nu_{i,k} \in [0, 1]$ the probability that device $i$ is included in the transmission schedule at time $k$ and further recall by $q(\mathbf{h}_{i,k}, \mu_i, \boldsymbol{\varsigma}_i)$ to be the packet delivery rate with which it transmits. Then, we may consider the *effective* packet delivery rate $\hat{q}$ as

$$\hat{q}(\mathbf{h}_{i,k}, \mu_i, \boldsymbol{\varsigma}_i) = \nu_{i,k}q(\mathbf{h}_{i,k}, \mu_i, \boldsymbol{\varsigma}_i) \tag{29}$$

Selective scheduling is motivated by the ultimate goal of minimizing total transmit time as described in the objective in (17). As we consider a large number of total devices $m$, scheduling all such devices will require a larger number of PPDU slots—a maximum of 9 devices can transmit within a single PPDU. Recall in (7) that each additional PPDU requires unavoidable overhead in $\tau_0$, which in aggregation over multiple PPDUs may become a significant bottleneck in minimizing $\hat{\tau}$ or meeting a strict latency requirement $\tau_{max}$. Thus, by decreasing the amount of scheduled devices, we may decrease the number of total PPDUs and the overhead that is added to the total transmission time.

Observe that by introducing the term $\nu_i$ to the evaluation of effective PDR $\tilde{q}_i$ in (29), we would thus need to transmit with higher PDR $q(\mathbf{h}_{i,k}, \mu_i, \boldsymbol{\varsigma}_i) \geq \tilde{q}_i(\hat{\mathbf{x}}_{i,k}^{(l_i)})/\nu_{i,k}$ to meet the condition in (23). While imposing a tighter PDR requirement will indeed require longer transmission times, this added time cost is generally less than the transmission overhead of additional PPDUs. In this work, we use the determine scheduling probability of device $i$ through its PDR requirement $\tilde{q}_i(\hat{\mathbf{x}}_{i,k}^{(l_i)})$ as

$$\nu_{i,k} := e^{\tilde{q}_i(\hat{\mathbf{x}}_{i,k}^{(l_i)})-1}. \tag{30}$$

With (30), the probability of scheduling device $i$ increases as the required PDR increases. Notice that, when a transmission is required, i.e. $\tilde{q}_i(\hat{\mathbf{x}}_{i,k}^{(l_i)}) = 1$, then device $i$ is included in the scheduling with probability 1. In general, devices with very high PDR requirements, e.g. $> 0.99$, will be scheduled with very high probability. Thus, the transmission time gains that are provided through selective scheduling using (30) would be minimal, if non-existent, in high-reliability settings in which PDR requirements remain high at all times. However, with the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2909198, IEEE Internet of Things Journal



lower PDR requirement obtained through the control-aware scheduling in (23), selective scheduling as the potential to create significant time savings, as will be later shown in Section V of this paper.

### C. Assignment-based scheduling

We now proceed to discuss how the PDR requirements previously derived are used to schedule the devices during a TXOP. Rather than employing a greedy method as is commonly done in wireless scheduling problems, in the proposed method we use assignment-type methods. In such assignment-type methods, we assign all scheduled devices to a PPDU and RU at the beginning of the TXOP rather than make scheduling decisions after each PPDU. To begin, we must determine a set of schedules that satisfy the constraints in (18)-(21). Recall each device $i$ is selected to be scheduled at cycle $k$ with probability $\nu_{i,k}$ and define the set of $m_k$ devices to selected be scheduled as $\mathcal{I}_k \subseteq \{1, 2, \ldots, m\}$ where $|\mathcal{I}_k| = m_k$. To specify the sets of RUs that we consider in our scheduling, we first define some notation necessary in the description. We define $\hat{\mathcal{S}}_{(n)} \subset \mathcal{S}$ to be an arbitrary set of RUs that do not intersect over any frequency bands (i.e. satisfy the constraint in (18)) with exactly $n$ elements. To accommodate the $m_k$ devices to be scheduled, we consider a set of $S_k$ such sets $\hat{\mathcal{S}}_{(n_s)}$ with size $n_s$, whose combined elements total $\sum_{s=1}^{S_k} n_s = m_k$. In other words, we identify a set $S_k$ PPDUs in which the $s$th PPDU contains $n_s$ non-intersecting PPDUs. We define this full set of assignable RUs at cycle $k$ as

$$\mathcal{S}'_k := \hat{\mathcal{S}}^1_{(n_1)} \cup \hat{\mathcal{S}}^1_{(n_2)} \cup \ldots \cup \hat{\mathcal{S}}^{S_k}_{(n_{S_k})}. \tag{31}$$

Note that in (31) we further superindex each set by a PPDU index $s$, in order to stress that elements are distinct between sets. That is, an RU $\boldsymbol{\varsigma}$ present in sets $\hat{\mathcal{S}}^x_{(n_x)}$ and $\hat{\mathcal{S}}^y_{(n_y)}$ is considered as two distinct elements in $\mathcal{S}'_k$, denoted $\boldsymbol{\varsigma}^x$ and $\boldsymbol{\varsigma}^y$, respectively. In this way (31) defines a complete set of combinations of frequency-allocated RU and *time*-allocated PPDUs to assign devices during this cycle. We point out that there are numerous ways in which to define such sets of RUs in each PPDU that total $m_k$ assignments. There are various heuristic methods that may be employed to quickly identify a permissible assignment pool $\mathcal{S}'_k$, and various simple heuristics may be developed to make this selection in a manner that reduces the overall latency of the transmission window. An example of the set $\mathcal{S}'_k$ for scheduling $m_k = 14$ devices is shown in Table II.

For all $i \in \mathcal{I}_k$ and RU $\boldsymbol{\varsigma} \in \mathcal{S}'_k$, define the largest affordable MCS given the *modified* PDR requirement $\tilde{q}_i(\hat{\mathbf{x}}^{(l_i)}_{i,k})/\nu_{i,k}$ by

$$\mu_{i,k}(\boldsymbol{\varsigma}) := \begin{cases} \max\{\mu \mid q(\mathbf{h}_{i,k}, \mu, \boldsymbol{\varsigma}) \geq \tilde{q}_i(\hat{\mathbf{x}}^{(l_i)}_{i,k})/\nu_{i,k}\} \\ 1, \quad \text{if } q(\mathbf{h}_{i,k}, \mu, \boldsymbol{\varsigma}) < \tilde{q}_i(\hat{\mathbf{x}}^{(l_i)}_{i,k})/\nu_{i,k} \ \forall \mu \end{cases} \tag{32}$$

Observe in (32) that, when no MCS achieves the desired PDR in a particular RU, this value is set to $\mu = 1$ by default. The above adaptive MCS selection can be achieved based on channel conditions using the techniques outlined in [35]. This MCS selection subsequently then yields a corresponding time cost $\tau(\mu_{i,k}(\boldsymbol{\varsigma}), \boldsymbol{\varsigma})$ for assigning device $i$ to RU $\boldsymbol{\varsigma}$. Further

| PPDU 1 | PPDU 2 | PPDU 3 |
|---|---|---|
| RU 1 | RU 10 | RU 13 |
| RU 2 | | |
| RU 3 | RU 11 | |
| RU 4 | | |
| RU 5 | RU 12 | RU 14 |
| RU 6 | | |
| RU 7 | | |
| RU 8 | | |
| RU 9 | | |

Table II: Example of RU selection with $m_k = 14$ devices. There are a total of $S_k = 3$ PPDUs, given $n_1 = 9$, $n_2 = 3$, $n_3 = 2$ RUs, respectively.

**Algorithm 1** Control-Aware Low Latency Scheduling (CALLS) at cycle $k$

1: **Parameters:** Lyapunov decrease rate $\rho$
2: **Input:** Channel conditions $\mathbf{H}_k$ and estimated states $\hat{\mathbf{X}}_k$
3: Compute target PDR $\tilde{q}_i(\hat{\mathbf{x}}^{(l_i)}_{i,k})$ for each device $i$ [cf. (23)].
4: Determine selection probabilities $\nu_{i,k}$ for each device [cf. (30)].
5: Select devices $\mathcal{I}_k$ with probs. $\{\nu_{1,k}, \ldots, \nu_{m,k}\}$
6: Determine set of RUs/PPDUs $\mathcal{S}'_k$ [cf. (31)].
7: Determine maximum MCS for each device/RU assignment [cf. (32)].
8: Schedule selected devices via assignment method.
9: **Return:** Scheduling variables $\{\boldsymbol{\Sigma}_k, \boldsymbol{\mu}_k, \boldsymbol{\alpha}_k\}$

define an 3-D assignment tensor $V$—where $v^s_{ij} = 1$ when device $i$ is assigned to RU $\boldsymbol{\varsigma}^s_j$ and 0 otherwise—and $\mathcal{V}$ as the set of all possible assignments. Recalling the form of the total transmission time given PPDU arrangements in (7), the assignment that minimizes total transmission time is given by

$$V^* = \underset{V \in \mathcal{V}}{\operatorname{argmin}} \sum_{s=1}^{S} \max_j \left[ v^s_{ij} \tau(\mu_{i,k}(\boldsymbol{\varsigma}^s_j), \boldsymbol{\varsigma}^s_j) \right]. \tag{33}$$

The expression in (33) can be identified as a particular form of the *assignment problem*, a common combinatorial optimization problem in which the selection of mutually exclusive assignment of agents to tasks incurs some cost. Here, the cost is the total transmission time across all PPDUs necessary for scheduled devices to meet the target PDRs. Assignment problems are generally very challenging to solve—there are $m_k!$ combinations—although polynomial-time algorithms exist for simple cases. The Hungarian method [36], for example, is a standard method for solving linear-cost assignment problems. While the cost we consider in (33) is nonlinear, the Hungarian method may be used as an approximation. Alternatively, other heuristic assignment approaches may be designed to approximate the solution to (33). We note that, for the simulations performed later in this paper, we apply such a heuristic method, the details of which are left out for proprietary reasons.

By combining these methods with the control-based PDR

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2909198, IEEE Internet of Things Journal



targets and selective scheduling procedure, we obtain the complete control-aware low-latency scheduling (CALLS) algorithm. The steps as performed by the centralized AP/controller are outlined in Algorithm 1. At each cycle $k$, the AP determines the scheduling parameters based on the current channel states $\mathbf{H}_k$ (obtained via pilot signals) and the current estimated control states $\hat{\mathbf{X}}_k$ (obtained via (2) for each device $i$). With the current state estimates, the AP computes target PDRs $\tilde{q}_i(\hat{\mathbf{x}}_{i,k}^{(l_i)})$ for each device via (23) in Step 3. In Step 4, the target PDRs are used to establish selection probabilities $\nu_{i,k}$ for each agent with (30). After randomly selecting devices $\mathcal{I}_k$ with their associated probabilities in Step 5, the set of RUs and PPDUs $\mathcal{S}'_k$ are determined in Step 6 as in (31) , based upon the number of devices selected to be scheduled $|\mathcal{I}_k|$. In Step 7, the associated MCS values are determined each possible assignment of device to RU via (32). Finally, in Step 8 the assignment is performed using either the Hungarian method [36] or other user-designed heuristic assignment method. The resulting assignment determines the scheduling parameters $\boldsymbol{\Sigma}_k, \boldsymbol{\mu}_k, \boldsymbol{\alpha}_k$ for the current cycle.

**Remark 3** Observe that the CALLS method as outlined in Algorithm 1 seeks to minimize the total latency of the transmission but does not explicitly prevent latency from exceeding some specific threshold $\tau_{\max}$. In practical systems, this limit may need to be enforced. In such a setting, the CALLS method can be modified so that all devices scheduled in PPDUs whose transmission end after $\tau_{\max}$ seconds do not transmit.

**Remark 4** In practical systems, the channel state information $\mathbf{H}_k$ is often obtained with some estimation errors, which may impact the channel aware scheduling approach taken here. Observe, however, that the computation of adaptive PDR targets in (23) does not depend upon channel state information. Thus, the primary component of the CALLS method—namely Steps 3-6 in Algorithm 1— is unaffected by inaccurate channel estimation. Likewise, the assignment method-based scheduling in Step 8 does also not directly depend upon channel information—see the formulation of the assignment problem in (33). The only component that may be negatively impacted by estimation errors is Step 7, i.e. the maximum MCS selection in (32). Here, estimation errors may result in the selection of an MCS that cannot meet the target PDR targets. Such an effect can be mitigated by selecting a smaller, more conservative MCS to account for channel estimation errors. Such a conservative scheme would tradeoff latency to the benefit of reliability or robustness.

## V. Simulation Results

In this section, we simulate the implementation of both the control-aware CALLS method and a standard "control-agnostic" scheduling methods for various low-latency control systems over a simulated wireless channel. We point out the low-latency based scheduling/assignment approaches of both methods being compared are identical, with the distinguishing features being the dynamic control-aware packet delivery rates incorporated in the CALLS method. In doing so, we may analyze the performance of the control-aware design outlined in the previous section relative to a standard latency-aware approach in terms of, e.g., number of users supported with fixed latency threshold or best latency achieved with fixed number of users. As we are interested primarily in low latency settings that tightly restrict the communication resources, we consider two standard control systems whose rapidly changing state requires high sampling rates, and consequently a communication latency on the order of milliseconds. The parameters for the simulation setup are provided in Table III. The wireless fading channel modeled using IEEE Indoor Channel Model E. In our performance analysis, we use link layer abstractions commonly used for wireless system level simulations (SLS) to model the wireless physical layer. In this approach, the AWGN SINR-BLER curves are used to evaluate the packet delivery rate function $q(\mathbf{h}, \mu, \boldsymbol{\varsigma})$; note that the curves are evaluated at the effective SNR (ESINR) values that take into account the instantaneous fading channel conditions and selected MCS. The transmission time $\tau(\mu, \boldsymbol{\varsigma})$ is computed in the simulations using the associated data rates of an MCS in Table I for a 100 byte packet and overhead (e.g. TFs) of the 802.11ax specifications. The latency overhead for this setting amounts to approximately $\tau_0 \approx 100\mu$s.

| | |
|---|---|
| Channel model | IEEE Model E (indoor) [37] |
| Sensor to AP distances | Random (1 to 50 meters) |
| Transmit power | 23 dbm |
| Channel bandwidth | 20 MHz |
| RU sizes | 2, 4, 8, 20 MHz |
| # of antennas at AP | 2 |
| # of antennas at sensors | 1 |
| MCS options | See Table I |
| State sampling period | 10 ms |

Table III: Simulation setting parameters.

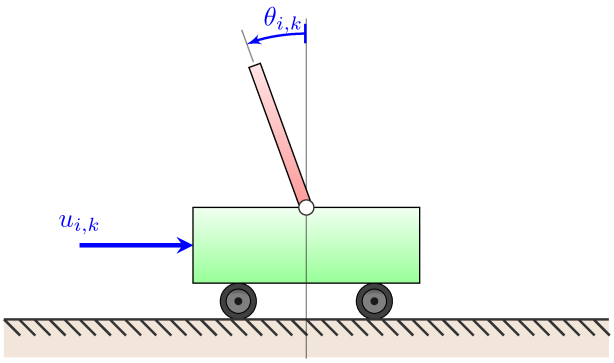


Figure 3: Inverted pendulum-cart system $i$. The state $\mathbf{x}_{i,k} = [x_{i,k}, \dot{x}_{i,k}, \theta_{i,k}, \dot{\theta}_{i,k}]$ contains specifies angle $\theta_{i,k}$ of the pendulum to the vertical, while the input $u_{i,k}$ reflects a horizontal force on the cart.

### A. Inverted pendulum system

We perform an initial set of simulations on the well-studied problem of controlling a series of inverted pendulums on a

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2909198, IEEE Internet of Things Journal



horizontal cart. While conceptually simple, the highly unstable dynamics of the inverted pendulum make it a representative example of control system that requires fast control cycles, and subsequently low-latency communications when being controlled over a wireless medium. Consider a series of $m$ identical inverted pendulums, as pictured in Figure 3. Each pendulum of length $L$ is attached at one end to a cart that can move along a single, horizontal axis. The position of the pendulum changes by the effects of gravity and the force applied to the linear cart. For our experiments, we use the modeling of the inverted pendulum as provided by Quanser [38]. The state is $p = 4$ dimensional vector that maintains the position and velocity of the cart along the horizontal axis, and the angular position and velocity of the pendulum, i.e. $\mathbf{x}_{i,k} := [x_{i,k}, \dot{x}_{i,k}, \theta_{i,k}, \dot{\theta}_{i,k}]$. The system input $u_{i,k}$ reflects a horizontal force placed upon $i$th pendulum. By applying a zeroth order hold on the continuous dynamics with a state sampling rate of $0.01$ seconds and linearizing, we obtained the following discrete linear dynamic matrices of the pendulum system

$$\mathbf{A}_i = \begin{bmatrix} 1 & 0.037 & 3.477 & 0.042 \\ 0 & 2.055 & -0.722 & 4.828 \\ 0 & 0.023 & 0.91 & 0.037 \\ 0 & 0.677 & -0.453 & 2.055 \end{bmatrix}, \mathbf{B}_i = \begin{bmatrix} 0.034 \\ 0.168 \\ 0.019 \\ 0.105 \end{bmatrix}. \quad (34)$$

Because the state $\mathbf{x}_{i,k}$ measures the angle of the $i$th pendulum at time $k$, the goal is to keep this close to zero, signifying that the pendulum remains upright. The input matrix $\mathbf{K}$ is computed to be a standard LQR-controller.

We perform a set of simulations scheduling the transmissions to control a series of inverted pendulums, varying both the latency threshold $\tau_{\max}$ and number of devices $m$. We perform the scheduling using the proposed CALLS method for control-aware low latency scheduling an, as a point of comparison, consider scheduling using a fixed "high-reliability" PDR of $0.99$ for all devices. Each simulation is run for a total of $1000$ seconds and is deemed "successful" if all pendulums remain upright for the entire run. We perform $100$ such simulations for each combination of latency threshold and number of devices to determine how many devices we can support at each latency threshold using both the CALLS and fixed-PDR methods for scheduling.

In Figure 4 we show the results of a representative simulation of the control of $m = 25$ pendulum systems with a latency bound of $\tau_{\max} = 10^{-3}$ seconds. In both graphs we show the average distance from the center vertical of each pendulum over the course of $1000$ seconds. In the top figure, we see by using the control-aware CALLS method we are able to keep each of the 25 pendulums close to the vertical for the whole simulation. Meanwhile, using the standard fixed PDR, we are unable to meet the scheduling limitations imposed by the latency threshold, and many of the pendulums swing are unable to be kept upright, as signified by the large deviations from the origin. This is due to the fact that certain pendulums were not scheduled when most critical, and they subsequently became unstable.

We present in Figure 5 the final capacity results obtained over all the simulations. We say that a scheduling method was able

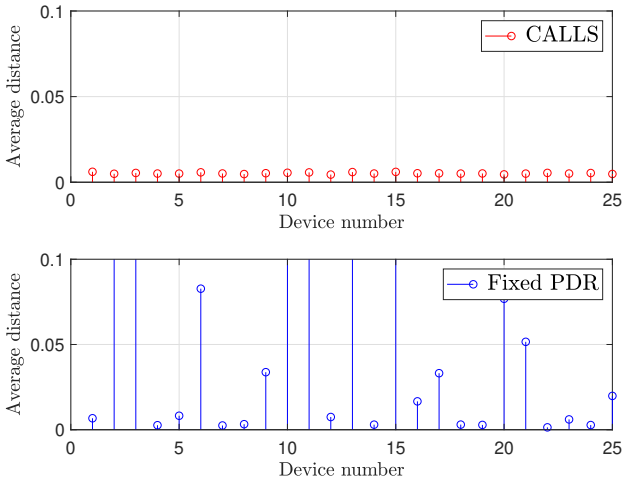


Figure 4: Average pendulum distance to center vertical for $m = 25$ devices using (top) CALLS and (bottom) fixed-PDR scheduling with $\tau_{\max} = 1$ ms latency threshold. The proposed control aware scheme keeps all pendulums close to the vertical, while fixed-PDR scheduling cannot.

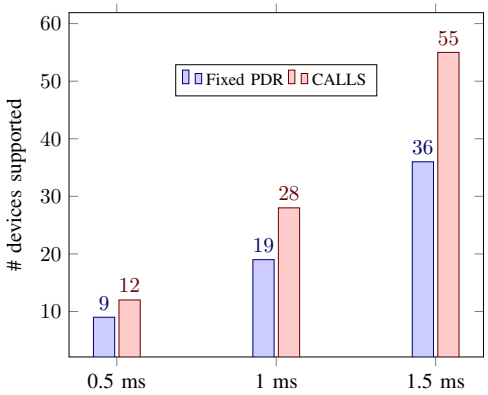


Figure 5: Total number of inverted pendulum devices that can be controlled using Fixed-PDR and CALLS scheduling for various latency thresholds.

to successfully serve $m'$ devices if it keeps all devices within a $|\theta_{i,k}| \leq 0.05$ error region for 100 independent simulations. Observe that the proposed approach is able to increase the number of devices supported in each case, with up to 1.5 factor increase over the standard fixed PDR approach. Indeed, the proposed CALLS method is able to allocate the available resource in a more principled manner, which allows for the support of more devices simultaneously being controlled.

### *B. Balancing board ball system*

We perform another series of experiments on the wireless control of a series of balancing board ball systems developed by Acrome [39]. In such a system, a ball is kept on a rectangular board with a single point of stability in the center of the board. Two servo motors underneath the board are used to push the board in the horizontal and vertical directions, with the objective to keep the ball close to the center of the board. The state here reflects the position and velocity in the horizontal

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2909198, IEEE Internet of Things Journal



and vertical axes, i.e. $\mathbf{x}_{i,k} := [x_{i,k}, \dot{x}_{i,k}, y_{i,k}, \dot{y}_{i,k}]$. The input $\mathbf{u}_{i,k} = [v_x, v_y]$ reflects the voltage applied to the horizontal and vertical motors. As before, we apply a zeroth order hold on the continuous dynamics with a state sampling rate of 0.01 seconds and linearize, thus obtaining the following dynamic system matrices,

$$\mathbf{A}_i = \begin{bmatrix} 1 & 0.01 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.01 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{B}_i = \begin{bmatrix} -0.0001 & 0 \\ -0.02 & 0 \\ 0 & -0.00008 \\ 0 & -0.01 \end{bmatrix}. \tag{35}$$

As before, we compute the control matrix $\mathbf{K}$ using standard LQR-control computation.

In the simulations performed with the balancing board system, in addition to making comparisons of the CALLS method to a fixed PDR low latency scheduling scheme, we perform additional comparisons to a standard control-aware scheduling approach—namely, the event-triggered scheduling approach [28], [29]. In event triggered scheduling, we schedule devices only when its estimated control state goes above some threshold value. When such an event occurs, this device is scheduled with a fixed high reliability PDR using a low-latency assignment based scheduling method. This, in effect, combines the selective scheduling approach of CALLS with fixed high reliability PDR targets commonly used in URLLC.

In Figure 6 we show the results of a representative simulation of the control of $m = 50$ balancing board ball systems with a latency bound of $\tau_{\max} = 10^{-3}$ seconds. Observe that, in this system, even with a large number of users, both the event-triggered scheduling and the CALLS method can keep all systems very close to the center of the board, while the fixed PDR scheduler loses a few of the balls due to the agnosticism of the scheduler.

To dive deeper into the benefits provided by control aware scheduling, we present in Figure 7 a histogram of the actual packet delivery rates each of the devices achieved over the representative simulation. It is interesting to observe that, for the CALLS method, the achieved PDRs are closely concentrated, ranging from 0.3 to 0.44. On the other hand, using either event-triggered or a fixed PDR scheduling scheme, the non-variable rates are too strict for the low-latency system to support, and without control-aware scheduling the achieved PDRs range wildly from close to 0 to close to 1. In this case, some devices are able to transmit almost every cycle while others are almost never able to successfully transmit their packets. This suggests that, by using control aware scheduling, we indirectly achieve a sense of fairness across users over the long term. Further note that the PDRs required to keep the balancing board ball stable, e.g. 0.4, are relatively small. This is due to the fact that the balancing board ball features relatively slow moving dynamics, making it easier to control with less frequent transmissions. This is comparison to the inverted pendulum system, in which the pendulums were kept stable with PDRs in the range 0.6-0.75.

We present in Figure 8 the final capacity results obtained over all the simulations for the balancing board ball system. Observe

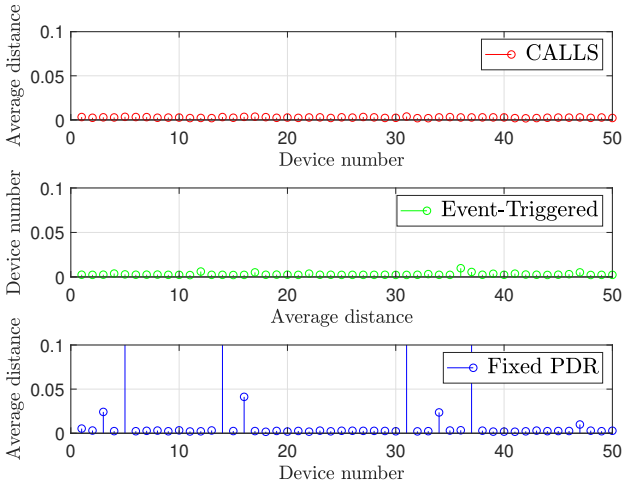


Figure 6: Average ball distance to center for $m = 50$ devices using (top) CALLS, (middle) event-triggered, and (bottom) fixed-PDR scheduling with $\tau_{\max} = 1$ ms latency threshold. The control aware schemes keeps all balancing balls close to center, while fixed-PDR scheduling cannot.

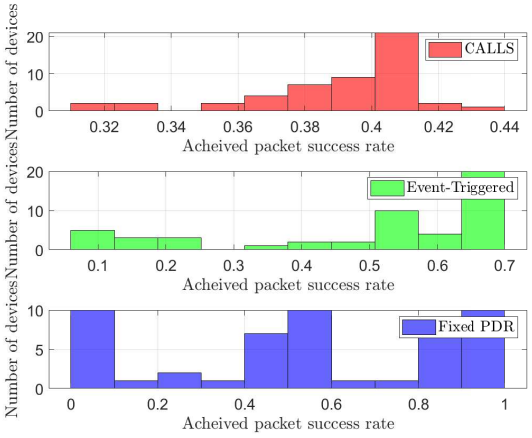


Figure 7: Histogram of achieved PDRs in $m = 50$ balancing board systems (top) CALLS, (middle) event-triggered, and (bottom) fixed-PDR scheduling with $\tau_{\max} = 1$ ms latency threshold. The proposed CALLS method achieves similar PDRs for all devices, while the fixed-PDR and event-triggered scheduling results in large variation in packet delivery rates.

that proposed approach increases the number of supported devices by factor of 2 relative to the standard fixed PDR approach. The even greater improvement here relative to the inverted pendulum simulations can be attributed to the slower dynamics of the balancing board ball, which allows for even more gains using control-aware PDRs due to the lower PDR requirements of the system. Likewise, the Lyapunov-based adaptive PDR requirements allow for even greater scalability than the more standard event-triggered approach, which can service only 17 and 50 users with 0.5 and 1 ms latency thresholds, respectively.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2909198, IEEE Internet of Things Journal



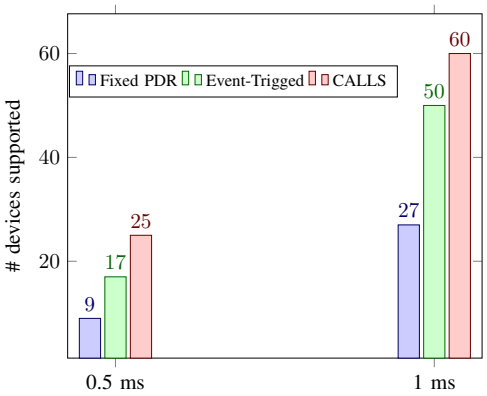


Figure 8: Total number of balancing ball board devices that can be controlled using Fixed-PDR, Event-Triggered, and CALLS scheduling for various latency thresholds.

## VI. Discussion and Conclusions

In this paper we proposed a novel control-communication co-design approach to solving the radio resource allocation problem for time-sensitive wireless control systems. Given a channel state and control state, we mathematically derive a minimum packet delivery rate a device must meet to maintain a control-orientated target, as defined by a stability-inducing Lyapunov function. By dynamically assigning variable packet delivery rate targets to each device based on its current conditions, we are able to more easily meet feasibility requirements of a latency-constrained wireless control problem and maintain stability and strong performance. We perform simulations on numerous well-studied low-latency control problems to demonstrate the benefits of using the control-aware approach, which can include a 2x gain on number of devices that can be supported. In future research, we aim to investigate how more sophisticated and realistic modeling, such as non-linear control or actuation over wireless links, may be used in this control-aware framework.

The results presented in this paper suggest an interesting potential for control-aware resource allocation and scheduling, particularly in low-latency industrial systems. By considering the control-specific targets such as maintaining stability or an error margin, we observe that the standard high reliability targets considered in URLLC (e.g. packet delivery rates $\geq 0.999$) can in some cases be substantially stricter than necessary for adequate performance. Wireless control systems with sufficiently slow dynamics can be kept stable with much lower packet delivery rates, which in turn make low-latency communications more achievable. Furthermore, in realistic industrial systems there will be many heterogeneous devices being controlled, whose variation in communication needs is well-served by control-aware adaptivity proposed in this paper. This suggests the potential for wireless communications to be adopted using a smart control-communication co-design approach even while ultra-reliable wireless system technology remains under development.

## References

[1] P. Zand, S. Chatterjea, K. Das, and P. Havinga, "Wireless industrial monitoring and control networks: The journey so far and the road ahead," *Journal of sensor and actuator networks*, vol. 1, no. 2, pp. 123–152, 2012.

[2] M. Wollschlaeger, T. Sauter, and J. Jasperneite, "The future of industrial communication: Automation networks in the era of the internet of things and industry 4.0," *IEEE Industrial Electronics Magazine*, vol. 11, no. 1, pp. 17–27, 2017.

[3] A. Varghese and D. Tandur, "Wireless requirements and challenges in industry 4.0," in *Contemporary Computing and Informatics (IC3I), 2014 International Conference on*. IEEE, 2014, pp. 634–638.

[4] X. Li, D. Li, J. Wan, A. V. Vasilakos, C.-F. Lai, and S. Wang, "A review of industrial wireless networks in the context of industry 4.0," *Wireless networks*, vol. 23, no. 1, pp. 23–41, 2017.

[5] M. Weiner, M. Jorgovanovic, A. Sahai, and B. Nikolié, "Design of a low-latency, high-reliability wireless communication system for control applications," in *Communications (ICC), 2014 IEEE International Conference on*. IEEE, 2014, pp. 3829–3835.

[6] P. Popovski, J. J. Nielsen, C. Stefanovic, E. de Carvalho, E. Strom, K. F. Trillingsgaard, A.-S. Bana, D. M. Kim, R. Kotaba, J. Park *et al.*, "Wireless access for ultra-reliable low-latency communication: Principles and building blocks," *IEEE Network*, vol. 32, no. 2, pp. 16–23, 2018.

[7] M. Bennis, M. Debbah, and H. V. Poor, "Ultra-reliable and low-latency wireless communication: Tail, risk and scale," *arXiv preprint arXiv:1801.01270*, 2018.

[8] N. Brahmi, O. N. Yilmaz, K. W. Helmersson, S. A. Ashraf, and J. Torsner, "Deployment strategies for ultra-reliable and low-latency communication in factory automation," in *Globecom Workshops (GC Wkshps), 2015 IEEE*. IEEE, 2015, pp. 1–6.

[9] O. N. Yilmaz, Y.-P. E. Wang, N. A. Johansson, N. Brahmi, S. A. Ashraf, and J. Sachs, "Analysis of ultra-reliable and low-latency 5g communication for a factory automation use case," in *Communication Workshop (ICCW), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1190–1195.

[10] E. Yaacoub and Z. Dawy, "A survey on uplink resource allocation in ofdma wireless networks," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 2, pp. 322–337, 2012.

[11] S. Lu, V. Bharghavan, and R. Srikant, "Fair scheduling in wireless packet networks," *IEEE/ACM Transactions on networking*, vol. 7, no. 4, pp. 473–489, 1999.

[12] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, P. Whiting, and R. Vijayakumar, "Providing quality of service over a shared wireless link," *IEEE Communications magazine*, vol. 39, no. 2, pp. 150–154, 2001.

[13] C. Wu, M. Sha, D. Gunatilaka, A. Saifullah, C. Lu, and Y. Chen, "Analysis of edf scheduling for wireless sensor-actuator networks," in *Quality of Service (IWQoS), 2014 IEEE 22nd International Symposium of*. IEEE, 2014, pp. 31–40.

[14] V. N. Swamy, S. Suri, P. Rigge, M. Weiner, G. Ranade, A. Sahai, and B. Nikolić, "Cooperative communication for high-reliability low-latency wireless control," in *Communications (ICC), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4380–4386.

[15] J. J. Nielsen, R. Liu, and P. Popovski, "Ultra-reliable low latency communication using interface diversity," *IEEE Transactions on Communications*, vol. 66, no. 3, pp. 1322–1334, 2018.

[16] B. Bellalta, "Ieee 802.11 ax: High-efficiency wlans," *IEEE Wireless Communications*, vol. 23, no. 1, pp. 38–46, 2016.

[17] J. Hespanha, P. Naghshtabrizi, and Y. Xu, "A survey of recent results in networked control systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 138–162, 2007.

[18] L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S. Sastry, "Foundations of control and estimation over lossy networks," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 163–187, 2007.

[19] M. Donkers, W. Heemels, N. Van De Wouw, and L. Hetel, "Stability analysis of networked control systems using a switched linear systems approach," *IEEE Transactions on Automatic Control*, vol. 56, no. 9, pp. 2101–2115, 2011.

[20] W. Zhang, M. Branicky, and S. Phillips, "Stability of networked control systems," *IEEE Control Systems Mag.*, vol. 21, no. 1, pp. 84–99, 2001.

[21] D. Hristu-Varsakelis, "Feedback control systems as users of a shared network: Communication sequences that guarantee stability," in *Proc. of the 40th IEEE Conf. on Dec. and Control (CDC)*, vol. 4, 2001, pp. 3631–3636.

[22] L. Zhang and D. Hristu-Varsakelis, "Communication and control co-design for networked control systems," *Automatica*, vol. 42, no. 6, pp. 953–958, 2006.

[23] J. Le Ny, E. Feron, and G. J. Pappas, "Resource constrained lqr control under fast sampling," in *Proc. of the 14th International Conference on Hybrid Systems: Computation and Control*. ACM, 2011, pp. 271–280.

[24] L. Meier, J. Peschon, and R. M. Dressler, "Optimal control of measurement subsystems," *IEEE Transactions on Automatic Control*, vol. 12, no. 5, pp. 528–536, 1967.
[25] H. Rehbinder and M. Sanfridson, "Scheduling of a limited communication channel for optimal control," *Automatica*, vol. 40, no. 3, pp. 491–500, 2004.
[26] M. S. Branicky, S. M. Phillips, and W. Zhang, "Scheduling and feedback co-design for networked control systems," in *Proc.of the 41st IEEE Conf. on Dec. and Control (CDC)*, vol. 2, 2002, pp. 1211–1217.
[27] J. W. Liu, *Real-Time Systems*. Prentice-Hall, Inc, 2000.
[28] A. Cervin and T. Henningsson, "Scheduling of event-triggered controllers on a shared network," in *Proc. of the 47th IEEE Conf. on Dec. and Control (CDC)*, 2008, pp. 3601–3606.
[29] M. H. Mamduhi, D. Tolić, A. Molin, and S. Hirche, "Event-triggered scheduling for stochastic multi-loop networked control systems with packet dropouts," in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*. IEEE, 2014, pp. 2776–2782.
[30] L. Shi, P. Cheng, and J. Chen, "Optimal periodic sensor scheduling with limited resources," *IEEE Transactions on Automatic Control*, vol. 56, no. 9, pp. 2190–2195, 2011.
[31] D. Han, J. Wu, H. Zhang, and L. Shi, "Optimal sensor scheduling for multiple linear dynamical systems," *Automatica*, vol. 75, pp. 260–270, 2017.
[32] K. Gatsis, M. Pajic, A. Ribeiro, and G. J. Pappas, "Opportunistic control over shared wireless channels," *IEEE Transactions on Automatic Control*, vol. 60, no. 12, pp. 3140–3155, December 2015.
[33] S. Sesia, M. Baker, and I. Toufik, *LTE-the UMTS long term evolution: from theory to practice*. John Wiley & Sons, 2011.
[34] M. Agiwal, A. Roy, and N. Saxena, "Next generation 5g wireless networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 1617–1655, 2016.
[35] R. P. F. Hoefel and O. Bejarano, "On application of phy layer abstraction techniques for system level simulation and adaptive modulation in IEEE 802.11 ac/ax systems," *Journal of Communication and Information Systems*, vol. 31, no. 1, 2016.
[36] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.
[37] J. Liu, R. Porat, N. Jindal *et al.*, "Ieee 802.11 ax channel model document," *Wireless LANs, Rep. IEEE 802.11–14/0882r3*, 2014.
[38] Quanser, "Linear inverted pendulum user manual." [Online]. Available: https://www.quanser.com/products/linear-servo-base-unit-inverted-pendulum/
[39] ACROME. [Online]. Available: https://www.acrome.net/ball-balancing-table