The Wireless Control Bus: Enabling Efficient Multi-hop Event-Triggered Control with Concurrent Transmissions

MATTEO TROBINGER, University of Trento, Italy
GABRIEL DE ALBUQUERQUE GLEIZER, Delft University of Technology, The Netherlands
TIMOFEI ISTOMIN, University of Trento, Italy
MANUEL MAZO JR., Delft University of Technology, The Netherlands
AMY L. MURPHY, Bruno Kessler Foundation, Italy
GIAN PIETRO PICCO, University of Trento, Italy

Event-triggered control (ETC) holds the potential to significantly improve the efficiency of wireless networked control systems. Unfortunately, its real-world impact has hitherto been hampered by the lack of a network stack able to transfer its benefits from theory to practice specifically by supporting the latency and reliability requirements of the aperiodic communication ETC induces. This is precisely the contribution of this paper.

Our *Wireless Control Bus (WCB)* exploits carefully orchestrated network-wide floods of concurrent transmissions to minimize overhead during quiescent, steady-state periods, and ensures timely and reliable collection of sensor readings and dissemination of actuation commands when an ETC triggering condition is violated. Using a cyber-physical testbed emulating a water distribution system controlled over a real-world multi-hop wireless network, we show that ETC over WCB achieves the same quality of periodic control at a fraction of the energy costs, therefore unleashing and concretely demonstrating its full potential for the first time.

CCS Concepts: • **Networks** → **Network protocol design**; • **Computer systems organization** → *Sensors and actuators*; • **Information systems** → *Process control systems*.

Additional Key Words and Phrases: event-based control, wireless sensor and actuator networks, concurrent transmissions.

1 INTRODUCTION

As a result of the joint effort of academia and industry, low-power wireless sensor networks (WSNs) are today a well-established technology, proven to be very dependable and energy-efficient. In the last twenty years, they have become the leading solution in a wide domain of applications, including environmental monitoring [11], wildlife tracking [42], smart cities [12], and the Internet of Things (IoT) at large [44]. This is due to the high scalability and (re)placement flexibility, yielding

Authors' addresses: Matteo Trobinger, Timofei Istomin, Gian Pietro Picco, {matteo.trobinger, timofei.istomin, gian-pietro.picco}@unitn.it, University of Trento, v. Sommarive 9, Trento, Italy, 38122; Gabriel de Albuquerque Gleizer, Manuel Mazo Jr., {G.deAlbuquerqueGleizer, m.mazo}@tudelft.nl, Delft University of Technology, Mekelweg 2, Delft, The Netherlands, 2628 CD; Amy L. Murphy, murphy@fbk.eu, Bruno Kessler Foundation, v. Sommarive 14, Trento, Italy, 38122.

lower installation and maintenance costs, and to ever-improving computing and communication features available on their untethered, autonomously-powered, small hardware footprint.

Low-power wireless networking for control: Challenges. The benefits are so significant that low-power wireless networking is now appealing also in traditionally wired domains like industrial control [28, 37]. Nonetheless, although WSNs are widely adopted for *monitoring*, their use for control and automation of plants and processes is still very limited [4]. Key concerns hampering wider adoption are the reliability of communication and the stability and magnitude of its latency. Modern controllers depend on the reliable and timely communication of relatively small data packets containing measurements and commands, generated frequently at the sensors and controller. Guaranteeing these properties is challenging in the large-scale, multi-hop scenarios that are often the main reason for a wireless approach. Moreover, staple applications for wireless control rely on battery-powered sensors, which places energy efficiency in the limelight, as replacing batteries is often costly or impractical. In this respect, it is well-known that radio activity, both listening and transmitting, is the main source of energy consumption. Therefore, the design of low-power wireless protocol stacks capable of minimizing communication without hampering control performance is of utmost importance for the widespread adoption of wireless control systems.

Event-triggered control: A missed opportunity? To facilitate the design of communications and simplify the control performance analysis, most networked control systems (NCS), whether wireless or wired, employ the classical periodic sampling of sensor data and update of actuator commands. The choice of sampling period involves a conservative, worst-case analysis of the closed-loop system dynamics. However, this conservative design enters in direct conflict with the objective of reducing energy consumption, enabled by the low-power WSN operation and key to *wireless* NCS (WNCS). To address this limitation, aperiodic methods adapting to the dynamic needs of the system have been investigated for a couple of decades (see, e.g., [3]).

A strong surge of interest began in 2007 with the systematic way of designing aperiodic sampling proposed in [48], currently known as *event-triggered control (ETC)*, revolving around the design of a *triggering condition* that only depends on sensor data. While this condition remains unsatisfied, a reference Lyapunov function decays at a certain speed[1]; otherwise, as soon as it is satisfied, sensor data is transmitted and control commands are updated. This procedure guarantees a prescribed decay of the Lyapunov function, serving as a certificate of performance for the control system, while significantly reducing the need for communication and, at least in principle, energy consumption.

Since then, many researchers embraced ETC and contributed to its theoretical foundations [20, 21, 39, 48, 52]. However, its application is still problematic. Although ETC naturally fosters resilience to communication delays [48], this tolerance has its own limitations, and the latency of communication imposes a limit on the achievable performance in terms of convergence rate to an equilibrium. Therefore, minimizing delays remains a critical goal for network stacks supporting ETC. Similar comments hold for reliability, whose crucial role is exacerbated as the entire network must timely and reliably react to the violation of triggering conditions for ETC to operate properly.

Guaranteeing these and other properties with a proper network stack is the most significant hampering factor to a wider adoption of ETC. Although wireless implementations of ETC exist [5, 17, 27, 45, 50], these are limited to small-scale, single-hop networks and exhibit poor reliability, high energy consumption, large and unpredictable delays, or a combination thereof, ultimately

---

[1]Lyapunov functions are widely employed in stability and performance analysis and design of control systems. Informally, it can be seen as a mathematical generalization of the energy of a system: it is always positive, it grows with the magnitude of the states, and it is zero only at the desired equilibrium point. A decaying Lyapunov function implies that the system is approaching the equilibrium point. For an exposition, see, e.g., [29].

preventing the overall system to seize the energy savings potentially enabled by ETC. This state of affairs is eloquently summarized in a recent survey on wireless control ([8], p. 22):

> "While in the control community, many so-called event-triggered estimation and control approaches have been developed in the last two decades, it remains largely unclear whether and how these can be *integrated* with the communication system and indeed result in demonstrable resource reallocation, savings, or other advantages for wireless systems in practice."

A wireless control bus for ETC. In this paper, we answer this question by providing a full-fledged network stack operating in conjunction with ETC, therefore unlocking its remarkable potential for energy savings hitherto hampered by the lack of appropriate communication support.

Our approach exploits *concurrent transmissions* (CTX) on the same radio channel, a technique popularized by Glossy [19] that has proven a very effective building block for protocol design. Several protocols embraced this technique[2], pushing the envelope of what can be achieved by IEEE 802.15.4 and, recently, other low-power wireless radios including BLE, UWB, and LoRa [58].

CTX-based protocols achieve at once very low latency, high reliability, low energy consumption, and accurate time synchronization. Based on efficient network-wide floods, they require neither a MAC nor a routing layer, and their performance is largely unaffected by changes in the topology induced, e.g., by node and link failures. This is a significant departure from conventional techniques (e.g., WirelessHART [1], ISA100.11.a [2], 6TiSCH [51]) that mitigate the packet losses and missed deadlines induced by network vagaries with continuous, high-overhead topology maintenance.

Instead, CTX-based protocols allow for the communication medium to be abstracted into a globally shared bus [18]; application data is broadcast to the entire network and can therefore be read by each node. In our context, this makes centralized control more appealing and even efficient than decentralized and distributed alternatives. Centralized controllers are generally easier to design and provide better performance than controllers accounting for network topology constraints; further, in the specific case of ETC they usually lead to fewer events being triggered. Unfortunately, the use of CTX in control is hitherto largely unexplored, apart from few recent exceptions [7, 9, 37] that however focus on periodic and self-triggered sampling rather than ETC.

Methodology and contributions. To achieve the remarkable potential benefits of CTX-based communications in ETC, co-design is fundamental. The control algorithm must work hand-in-hand with the underlying network stack to seize opportunities to reduce the radio active time while ensuring the timeliness and reliability key to control performance. In ETC, control update times are not defined a priori; sensors decide on-the-fly whether to send updated readings based on their triggering condition. This *in theory* reduces communication w.r.t. classic control approaches; *in practice*, it must be supported by a network stack capable of *i)* minimizing network overhead during the control idle times, and *ii)* promptly react to triggered events by ensuring timely and reliable collection of sensor readings at the controller and dissemination of updated actuation commands.

We address these challenges with the *Wireless Control Bus (WCB)*, a novel protocol that, to the best of our knowledge, is the first supporting multi-hop communication for ETC, and does so efficiently and reliably. We first summarize the technical foundations of ETC and, motivated by the co-design of control and communication in WCB, put forth a side contribution further reducing communication via rejection of step-disturbances (§2). We then illustrate how the design of WCB (§3) exploits CTX to meet the above requirements of ETC w.r.t. latency, reliability, and energy efficiency. Moreover, we present a WCB variant that can easily accommodate conventional periodic

---

[2]Some authors use the label *synchronous transmissions* instead, with equivalent meaning in this context.

strategies, endowing them with similarly unprecedented performance and ultimately fostering a holistic approach to control design enabled by a *single* network stack.

We demonstrate the effectiveness of our solutions via a water-irrigation system (WIS) test case, for which we define an ETC-based control strategy (§4). A WIS typically extends for kilometers, likely requiring multi-hop communication, in turn demanding complex decentralized or distributed control strategies, as in [10]. In contrast, our combination of WCB and ETC enables a simpler centralized control, as we show experimentally. In this respect, a realistic evaluation is a challenge per se, as we are not aware of large-scale WIS testbeds. Small-scale ones, e.g., the double-tank system [38], are widely adopted but rely on a single-hop, star topology, unsuited to evaluate the multi-hop systems envisioned for industrial wireless control and targeted by this work.

We overcome these limitations with a secondary contribution: the design of a cyber-physical testbed (§5) that adopts a real-time, network-in-the-loop approach integrating *i)* a Simulink *model* emulating the physical system, and *ii) real* embedded devices acting as sensors, actuators, for-warders, and controller, executing our control and protocol stack and interacting only wirelessly. We experiment with two distinct networks, where we analyze the sensitivity of WCB to its parameters (§6), identify the configuration we use in our extensive experimental campaign, and assess the impact of different scales and topologies on the performance of our ETC system.

The experimental results (§7) demonstrate the effectiveness of our approach. The quality of the control achieved by ETC over WCB is virtually the same as periodic sampling. However, it comes at a fraction of communication costs; sample count is reduced by >89%, yielding a >62% reduction in radio-on time w.r.t. periodic control—far more than previously observed in the ETC literature [6] in significantly more constrained setups. This confirms that WCB not only provides a network stack, hitherto missing, enabling ETC in multi-hop networks, but also effectively translates the reduction of control traffic enabled by ETC into corresponding savings in energy consumption.

The paper ends with a summary of related work (§8) and brief concluding remarks outlining opportunities for future work (§9) on WCB, which we intend to release publicly as open source.

2 EVENT-TRIGGERED CONTROL

Event-triggered control (ETC) is a sampling strategy in which the update of sensor data to feedback controllers and of control commands to actuators is determined *on-the-fly* by a *triggering condition*. This is a drastic departure from time-triggered control, which includes the classic periodic control.

In a nutshell, when something relevant happens on the state of a dynamic system, the sensors communicate their most recent values to the controller; otherwise, these values are held constant, and actuators typically also hold their positions. Intuitively, data is sampled *only when needed*, reducing the communication induced by control. In practice, determining when fresh data is needed is somewhat involved and requires control theory to ensure stability and good performance.

We formally describe ETC, including equations for a distributed implementation suited to CTX. In doing so, we also present two contributions: *i)* a generalization of the decentralized ETC strategy in [39] to a broader class of triggering conditions and sensor node arrangements (§2.3), and *ii)* an adaptation of unperturbed ETC strategies to the problem of step disturbance rejection (§2.4).

2.1 Sample-and-hold control

Hereafter, we consider a linear time-invariant (LTI) system with measurable states of the form

$$\dot{x}(t) = Ax(t) + Bu(t) + Ew(t), \tag{1}$$

where $x(t) \in \mathbb{R}^n$ is the vector of states, $u(t) \in \mathbb{R}^m$ is the vector of control inputs, $w(t) \in \mathbb{R}^p$ is the vector of exogenous unmeasured disturbances, and A, B, E are known system matrices of appropriate dimensions. In this work, we assume that all states are measured by sensors. For digital

implementation, we consider a state-feedback controller realized in a sample-and-hold fashion:

$$\boldsymbol{u}(t) = \boldsymbol{K}\hat{\boldsymbol{x}}(t), \tag{2}$$

where $\boldsymbol{K}$ is a control gain matrix to be designed, and $\hat{\boldsymbol{x}}(t)$ is the sampled state, which satisfies, for a sequence of sampling times $\{t_i\}_{i\in\mathbb{N}}$,

$$\hat{\boldsymbol{x}}(t) = \boldsymbol{x}(t_i), \forall t \in [t_i, t_{i+1}). \tag{3}$$

We say that the obtained closed-loop system is *globally exponentially stable* if, for every initial condition $\boldsymbol{x}(0)$, all of its solutions satisfy $|\boldsymbol{x}(t)| \leq M|\boldsymbol{x}(0)|\mathrm{e}^{-\rho t}$ for some $0 \leq M < \infty$ and $\rho > 0$, where ρ is called the *decay rate* of the system.

When using *periodic sampling*, the sequence $\{t_i\}_{i\in\mathbb{N}}$ satisfies $t_i = ih$, for some designed sampling time h. In ETC, the sequence of sampling times is *not* known a priori; instead, it is generated based on some designed *triggering condition* dependent on the states. Although there is a vast literature on ETC, this section focuses on mechanisms enabling two important practical aspects for WNCS:

1. *Triggering conditions can be checked periodically.* This is known as periodic ETC, or PETC [21], and it can achieve a control performance arbitrarily close to that of classical ETC [43]. Periodic checking of triggering conditions allows for an efficient scheduling of sleep times, which we exploit in the design of WCB (§3). In contrast, classical ETC requires continuous monitoring of triggering conditions, forcing sensors to be always active and preventing energy savings.
2. *Triggering conditions can be checked locally on the sensor nodes.* The alternative of checking them on the controller side would require sensors to send data to it periodically, which would eliminate any communication-related energy savings.

We detail these two aspects next. Moreover, we note that our focus on LTI systems is mainly due to the test case we consider, which can be tackled as an LTI control problem (§4.1). PETC, decentralized ETC, and robust ETC for general non-linear systems have been addressed in, e.g., [43], [39], and [34], respectively. Stability analysis in the non-linear case differs from the one presented in this paper, but the structure of the ETC mechanism is the same. Hence, the WCB wireless protocol proposed here can be applied to non-linear or robust control problems without any changes.

2.2 Periodic event-triggered control

Using the framework of [21], we define a periodic event-triggered state-feedback system as the one captured by (1)–(3) with the triggering times satisfying

$$t_{i+1} = \inf \left\{ t = kh > t_i, k \in \mathbb{N} \, \left| \, \begin{bmatrix} \boldsymbol{x}(t) \\ \hat{\boldsymbol{x}}(t) \end{bmatrix}^\top \boldsymbol{T} \begin{bmatrix} \boldsymbol{x}(t) \\ \hat{\boldsymbol{x}}(t) \end{bmatrix} > \epsilon^2 \right. \right\}, \tag{4}$$

where $\boldsymbol{T}$ is a triggering matrix to be designed and ϵ is a design parameter whose value controls the size of the terminal set to which the system converges. When $\epsilon = 0$, the system converges and stabilizes at the desired equilibrium. A small $\epsilon > 0$ increases the inter-sample times at the expense of stabilizing a set around the equilibrium, of size proportional to ϵ. When persistent external disturbances $\boldsymbol{w}$ are present, one cannot stabilize the origin; setting $\epsilon > 0$ is necessary to prevent excessive sampling precisely when the system is essentially under control, i.e., close to equilibrium.

Several tools are available to verify the stability of the closed-loop system using a given triggering matrix $\boldsymbol{T}$. We recall now one of the results from [21]:

THEOREM 2.1 ([21], THEOREM III.4). *With $\epsilon = 0$ and $w(t) \equiv 0$, the PETC system (1)–(4) is globally exponentially stable (GES) with decay rate ρ if there exist symmetric matrices $\boldsymbol{P}_1, \boldsymbol{P}_2$, and scalars*

$\alpha_{ij} \geq 0, \beta_{ij} \geq 0,$ and $\kappa_i \geq 0, i, j \in \{1, 2\},$ satisfying[3]

$$\mathrm{e}^{-2\rho h} P_i - A_i^\mathsf{T} P_j A_i + (-1)^i \alpha_{ij} T + (-1)^j \beta_{ij} A_i^\mathsf{T} T A_i \geq 0, \quad \forall i, j \in \{1, 2\}, \quad and$$

$$P_i + (-1)^i \kappa_i T > 0, \quad \forall i \in \{1, 2\},$$

$$\text{where } A_1 := \begin{bmatrix} A + BK & 0 \\ I & 0 \end{bmatrix}, \quad A_2 := \begin{bmatrix} A & BK \\ 0 & I \end{bmatrix}.$$

We use this result in our test case (§4) to design appropriate triggering conditions, i.e., a matrix T that guarantees appropriate control performance for a given sampling time h.

2.3 Distributed event-triggered conditions

The triggering condition in (4) is, in its most general form, a centralized one, i.e., all states are needed to determine when to sample. However, when sensors are remotely located w.r.t. each other, this approach becomes impractical. Fortunately, decentralized triggering conditions exist that address this issue. Here we focus on the strategy proposed in [39], consisting of three key steps posing corresponding requirements on the network stack supporting control:

1. Each sensor has its own triggering condition, which can trigger a controller update *independently* of readings from other sensors.
2. Upon one sensor triggering, *all* others must transmit their up-to-date readings to the controller.
3. Finally, the controller updates its control command and sends it to the actuators.

The following type of triggering condition is used as a starting point in [39]:

$$|\boldsymbol{x}(t) - \hat{\boldsymbol{x}}(t)| > \sigma |\boldsymbol{x}(t)|, \tag{6}$$

where σ is a triggering parameter and $|\cdot|$ is the Euclidean norm. This condition, introduced by the seminal work in [48], essentially compares the *sampling error* $\boldsymbol{x}(t) - \hat{\boldsymbol{x}}(t)$ against the state values themselves; if the error is large enough, it is time to update the measurements at the controller.

The main observation in [39] is that by rewriting (6) one obtains the implication:

$$\sum_{i=1}^{n} (x_i(t) - \hat{x}_i(t))^2 - \sigma^2 x_i^2(t) > 0 \Rightarrow \bigvee_{i=1}^{n} \left((x_i(t) - \hat{x}_i(t))^2 - \sigma^2 x_i^2(t) > \theta_i \right) \tag{7}$$

as long as $\sum_{i=1}^{n} \theta_i = 0$ for n state variables. This enables using each of the i-th conditions in the RHS of (7) independently at each sensor. The triggering parameters θ_i can be designed offline or adapted online. Hereafter, we focus on the former; details of their computation are found in [39].

Observe that (6) can be cast in the form of (4) with $T = \begin{bmatrix} (1-\sigma^2)I & -I \\ -I & I \end{bmatrix}$ and $\epsilon = 0$. Thus, a simple generalization of the approach described above is possible, to include a larger class of triggering conditions of the form (4), where more parameters (i.e., all elements of T) than simply σ are to be designed. This introduces additional design flexibility for the triggering conditions, which can be used to further reduce the amount of communication triggered by the system.

First, denote the sampling error $\boldsymbol{e} := \hat{\boldsymbol{x}} - \boldsymbol{x}$. Assume $q \leq n$ sensor nodes, each measuring one or more state variables, and denote by $\mathcal{I}_j \subseteq \{1, 2, ..., n\}$ those measured by node j with $\bigcap_{j=1}^{q} \mathcal{I}_j = \emptyset$, i.e., each state variable is measured by only one node. Then, a triggering condition of the form

$$\boldsymbol{e}(t)^\mathsf{T} \boldsymbol{M} \boldsymbol{e}(t) - \boldsymbol{x}(t)^\mathsf{T} \boldsymbol{N} \boldsymbol{x}(t) > \epsilon^2, \tag{8}$$

is decentralizable if the triggering matrices $\boldsymbol{M} = \boldsymbol{M}^\mathsf{T}$ and $\boldsymbol{N} = \boldsymbol{N}^\mathsf{T}$ have the following structure: an element $M_{ii'}$ ($N_{ii'}$) is nonzero if and only if i and i' belong to the same set $\mathcal{I}_j$ for some sensor

---

[3]For a symmetric matrix $A = A^\mathsf{T}$, we say that $A > 0$ ($A \geq 0$) if it is positive-(semi)definite.

node j. Then, denoting by x_j, e_j, M_j and N_j the subvectors and submatrices containing the rows and columns $\mathcal{I}_j$ of x, e, M and N, we obtain that (8) implies:

$$\bigvee_{j=1}^{q} (e_j(t)^{\mathsf{T}} M_j e_j(t) - x_j(t)^{\mathsf{T}} N_j x_j(t) > \theta_j), \qquad \text{with } \sum_{j=1}^{q} \theta_j = \epsilon^2. \tag{9}$$

To make triggering as infrequent as possible, during design one may want to maximize some norm of N and minimize M, so that the negative term in (8) dominates the inequality. Note that the triggering condition (8) admits the form in (4) with $T = \begin{bmatrix} M-N & -M \\ -M & M \end{bmatrix}$, therefore Theorem 2.1 can be used to verify global exponential stability. This theorem can also be used to co-design, and optimize for sparse sampling, the matrices P_i and the triggering matrices M and N; by fixing the values of $\kappa_i, \alpha_{ij},$ and β_{ij}, the problem becomes a linear matrix inequality (LMI) that can be easily solved with existing optimization software. To prevent the triggering condition from being repeatedly violated after the previous sample, when $e(t_i) = 0$, N must be positive semidefinite.

2.4 The problem of disturbance rejection

The ETC mechanisms presented in this section are associated with the problem of stabilizing the origin, disregarding the effects of disturbances. Still, the presented triggering strategies also give disturbance attenuation properties in the case of linear systems. For example, sufficient conditions to verify a finite $\mathcal{L}_\infty$ gain are also present in [21].

In disturbance rejection problems, like the one we address in the WIS example on which we evaluate our solution, there is an important specificity: with the appropriate control design, one can ensure that a set of states (the control outputs $y(t) \in \mathbb{R}^p, y = Cx$) still converge to zero; the remaining states also converge, but to some unknown signal dependent on the disturbances (constant values in the case of step disturbances).

If the objective is to stabilize the system to a given reference x^*, the general approach to event design is to perform a change of coordinates $\tilde{x} := x - x^*$, which renders the problem again stabilizing $\tilde{x}$ to the origin. With this change of coordinates, note that the sampling error component does not change, i.e. $\hat{e} = \tilde{\hat{x}} - \tilde{x} = \hat{x} - x = e$. Condition (8) becomes

$$e(t)^{\mathsf{T}} M e(t) - (x(t) - x^*)^{\mathsf{T}} N (x(t) - x^*) > \epsilon^2. \tag{10}$$

In the case of step disturbance rejection, some of the components of x^* are unknown and vary depending on the disturbance. This makes it impossible to implement (10) in its most general form. However, if one constrains the elements of N associated with the unknown entries of x^* to be zero, these terms do not appear in the equation, and the triggering condition is implementable regardless of the disturbance levels. Mathematically, the matrix on the second term of (10) takes the form $(C^{\mathsf{T}}C)N(C^{\mathsf{T}}C)$, and the triggering condition can be implemented as

$$e(t)^{\mathsf{T}} M e(t) - y(t)^{\mathsf{T}} C^{\mathsf{T}} N C y(t) > \epsilon^2, \tag{11}$$

which can be decentralized to take the form in (9). To verify stability, one can use Theorem 2.1 with

$$T = \begin{bmatrix} M - (C^{\mathsf{T}}C)N(C^{\mathsf{T}}C) & -M \\ -M & M \end{bmatrix}.$$

3 DESIGNING THE WIRELESS CONTROL BUS

The main focus of ETC is to avoid communication during steady-state, while preserving correct and timely control outside of it. From a network standpoint this means that *i)* when control traffic is absent, network overhead should be minimized; otherwise, *ii)* the collection of sensor readings at the controller and consequent dissemination of actuation commands should occur timely and

reliably. These requirements, already challenging when taken individually, are even harder to fulfill when combined; a quiescent network, ideal to minimize consumption, is intrinsically at odds with a reactive and reliable one. It is therefore not surprising that a wireless network stack efficiently supporting ETC is still missing, hampering the practical adoption of this control approach.

WCB tackles this challenge by relying on concurrent transmissions (§3.1), whose peculiar properties are exploited to cater for the specific needs of ETC (§3.2) and, within the same protocol framework, also of traditional periodic control (§3.3).

3.1 Concurrent Transmissions in a Nutshell

Conventional network protocols stagger transmissions to minimize packet collisions. In contrast, protocols based on concurrent transmissions (CTX) exploit nodes transmitting *at the same time*.

In IEEE 802.15.4, these protocols rely on two PHY-level phenomena [13, 58]. The so-called constructive (or, more correctly, non-destructive) interference occurs when *identical* packets from multiple senders arrive at the receiver with a time displacement <0.5 μs, the duration of a bit (chip) in the transmitted chip sequence obtained by the direct-sequence spread spectrum (DSSS) encoding of the original packet. In this case, the signals are likely to mix non-destructively, and the packet is successfully decoded. The capture effect, instead, occurs even for *different* packets, as long as they arrive with a relative shift <160 μs, the duration of the synchronization header; one of the packets is likely received, depending on the density of neighbors and their relative signal strength.

The effectiveness of CTX has been demonstrated by the Glossy system [19] that, originally designed for multi-hop time synchronization, exploits the two phenomena above to achieve fast, energy-efficient, and reliable network floods. The *initiator* begins a flood by broadcasting a packet. As the rest of the network is assumed to be already listening on the channel, the packet is received and immediately rebroadcast by neighbors, yielding CTX. For redundancy, each node retransmits the packet up to N times. The value of N is key to determine the balance among reliability, latency, and energy consumption. Similarly, the slot duration must be short, to minimize the energy consumption due to listening, yet be long enough to accommodate all required packet transmissions. Thanks to the massive concurrency, in practice the flood duration does *not* depend on the number of nodes in the network but only on the network radius, ensuring a latency—few milliseconds for few hops—very close to the theoretical minimum when using half-duplex radios.

Since [19], the popularity of CTX increased dramatically, leading to several low-power wireless systems significantly pushing the performance boundary along several protocol dimensions, even in PHY radio layers other than IEEE 802.15.4 [13, 58] and in presence of harsh RF interference [46]. These protocols typically exploit Glossy floods as primitive building blocks, composing and scheduling them differently in a distributed fashion, and exploiting either or both PHY-level properties of CTX depending on the protocol goals at hand. WCB adopts a similar approach, as described next.

3.2 A Network Stack for Event-Triggered Control

Core concepts. Communication in WCB is structured around non-overlapping time *slots*, each containing a separate Glossy flood, potentially initiated by different nodes. The same sequence of time slots repeats at all nodes with a fixed interval called *epoch*, characterized by a very short initial active portion where communication occurs, and a much longer one where nodes turn off their radio and remain in sleep mode.

This structure, common to many CTX-based systems, relies on the accurate, network-wide time synchronization enabled by Glossy as part of its operation, and effectively *abstracts the multi-hop wireless network into a shared control bus with time-slotted access*. This simplifies significantly the development of the overall control system by removing all the complexity typically associated with

(a) Supporting event-triggered control: WCB-E.

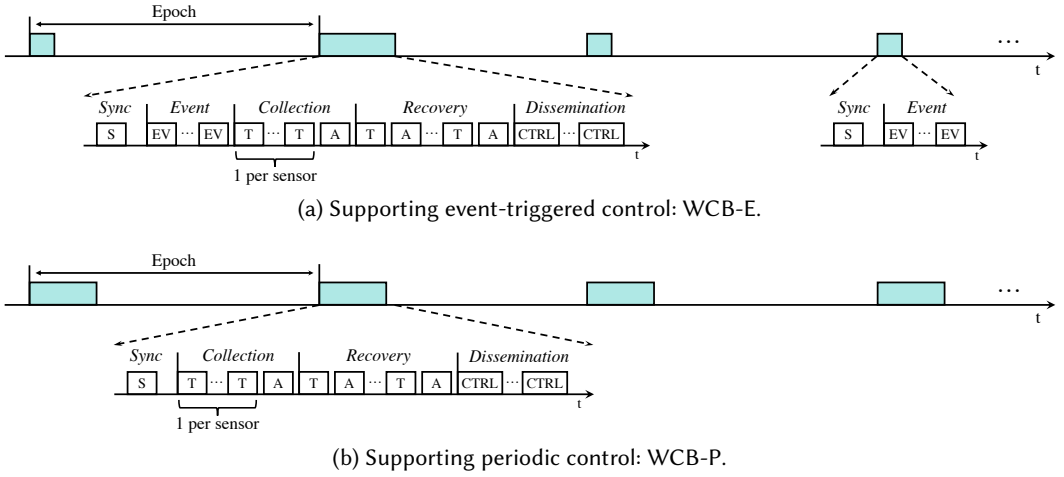(b) Supporting periodic control: WCB-P.

Fig. 1. The Wireless Control Bus, WCB.

multi-hop networks (e.g., at the MAC and routing layers) and, at the same time, ensuring high determinism in terms of latency and reliability—key for control design and performance.

Time slots can be *i)* dedicated to a single flood by one sender, *ii)* used by multiple senders concurrently flooding the same packet, or *iii)* by multiple senders flooding different packets competing in the same slot. Although in all cases one packet is received with high probability, experience with CTX-based systems shows that they offer decreasing degrees of reliability (§3.1). WCB balances the pros and cons of each slot type depending on the target functionality, described next.

Protocol phases. The active portion of a WCB epoch is structured in the following groups of functionally-related slots, or *phases* (Figure 1):

1. *Synchronization.* CTX require tight time synchronization, which is also useful to establish a common time reference for control. However, prolonged sleeping periods—the main asset in reducing energy consumption—significantly increase clock drift. Therefore, as common in CTX-based protocols (§8), each WCB epoch begins with a S slot whose flood, initiated by the controller, contains the timestamp of packet transmission. This is used as a time reference by all other nodes that, by combining this information with the number of hops the packet has traveled, realign their local time reference that of the controller.

2. *Event.* This phase is key to efficient ETC support. After synchronization, each sensor node acquires its measurements and evaluates the triggering condition in (11) (§2). If this holds, a special and very short event notification packet—the same at all nodes—is flooded in one or more EV slots. Multiple events may be generated simultaneously at different nodes. However, due to the properties of CTX (§3.1), this packet is received with very high reliability at all nodes, informing them *at once* of the need to participate in the subsequent network-wide data collection (left schedule, Figure 1a). Otherwise, if no event is generated, the nodes can safely enter sleep for the remaining portion of the epoch (right schedule, Figure 1a).

3. *Collection.* Sensors report their readings as a sequence of T slots, each reserved to a sensor node performing an isolated flood. At the end, the A slot is reserved for an acknowledgment flood by the controller, containing a bitmap denoting which sensor packets have been successfully

received. Thanks to the reliability of CTX, most of the times all reports are gathered, and all nodes can enter sleep until the dissemination phase (step 5).

4. *Recovery.* In the rare cases where a sensor node does not receive an acknowledgment or realizes that its packet is not confirmed in the bitmap, the node attempts retransmission in the subsequent T slot. Unlike collection, where each node transmits in a designated slot, during recovery unacknowledged sensors *compete* in the same T slot with concurrent floods for their missed packets. Again due to the properties of CTX (§3.1), one of these packets reaches with high probability the controller, which updates the acknowledgment bitmap and floods it back in the A slot, effectively eliminating one of the competing nodes from the next TA slot pair. This alternating sequence repeats until the controller acknowledges all packets, allowing nodes to safely enter sleep until dissemination, or a pre-defined number R of TA pairs is executed.

5. *Dissemination.* After collecting sensor readings, the controller generates the actuation commands. In the unlikely case where some readings are still missing after recovery, their values from the previous collection are employed by the controller. This is the choice best aligning with the properties of ETC (§2), although alternative ones can be easily integrated, if required. Actuation commands are packed in a single packet and disseminated in one or more CTRL slots by a controller-initiated flood; actuators apply the received commands upon their arrival. We *always* include commands for *all* actuators, even when their state is unchanged w.r.t. the previous dissemination, as this provides actuators with multiple chances to receive occasionally missed commands. Dissemination is the last phase of the epoch active portion; upon completion of the last CTRL flood, the network automatically deactivates and all nodes enter sleep mode.

Ensuring reliability. Each phase exploits different mechanisms to guarantee packet delivery. Recovery exploits an acknowledgment slot A after a T slot, enabling competing nodes to determine whether their packet has been received. This technique has proven very effective [25] when the number of concurrent transmitters is a priori unknown. Nevertheless, in the collection phase it would double the number of slots required and therefore latency and energy consumption. Instead, we exploit a priori knowledge that *all* sensors nodes must transmit, and send a single, cumulative acknowledgment in the A slot at the end of collection, itself triggering recovery only when needed.

Moreover, we note that a similar strategy could also lead to an alternative design of the recovery phase. The acknowledgment bitmap ending the collection phase provides nodes with enough global information to schedule their retransmissions back-to-back in dedicated slots, replied to by a single, final, collective acknowledgment. This scheme reduces the number of slots in the recovery phase and the contention in the T slots, and may be useful when several packets must be recovered at once. Nevertheless, it hinges on the correct reception of the acknowledgment bitmap, whose loss may be more common precisely in scenarios with several packet losses. Ultimately, these tradeoffs depend on the target environment; as our test environments (§6, §7) reveal very few lost packets, we use the simpler mechanism with competing retransmissions and individual acknowledgments.

The mechanisms above are effective when packets must be delivered to a *single* node—the controller—that can signal their failed receipt. However, they are impractical when packets must reliably reach *multiple* nodes, as in the event and dissemination phases. In these cases, we exploit *redundancy* as a simple yet effective technique to increase reliability, and repeat the EV or CTRL multiple times. The number of repetitions is crucial, as it governs the tradeoffs between reliability and energy consumption; we analyze this parameter experimentally in §6.

Finally, we exploit *channel hopping* to further increase resilience to interference, common in industrial scenarios but also in indoor settings (e.g., due to WiFi) like those in our experiments (§5). As WCB nodes execute the same schedule in lockstep, even during the dynamic recovery portion, the frequency channel to be used in each slot can change following a globally-known hopping

sequence. This technique is considered state-of-the-art in the context of CTX, as we further discuss in §7.2. Its effectiveness towards interference resilience, without hampering latency and energy consumption, has been thoroughly demonstrated in [26], which directly inspires our design.

3.3 One Wireless Bus to Rule Them All: Periodic Control over WCB

Our stated goal for the design of WCB is to efficiently support ETC. Nevertheless, our protocol can be easily tailored to periodic control by regarding it as a special case of ETC in which the triggering condition is violated during *all* epochs. This renders the dynamic and distributed coordination offered by the event phase superfluous, leading to the schedule in Figure 1b. Hereafter, we refer to this specific variant targeting periodic control as WCB-P whenever necessary to distinguish it from the original protocol targeting ETC (Figure 1a), itself referred to as WCB-E.

Although the modifications leading to WCB-P are simple, their impact should not be under-estimated. On one hand, the dedicated support offered by WCB-E to ETC remains crucial. The active periods in WCB-P are generally longer than in WCB-E, resulting in significantly less energy-efficient communication, as hinted at by the larger active portions of the former in Figure 1 and quantitatively shown in our experimental evaluation (§7). On the other hand, due to the specific application and control requirements, periodic control may be preferable to ETC. In these cases, the efficiency and performance offered by WCB-P over multi-hop networks is unprecedented. Further, the ability to use the *same* protocol stack for both flavors of control, ETC and periodic, is a tremendous asset. Not only it greatly reduces the complexity of control design and implementation, but also fosters a holistic approach where the selection of the best control strategy is driven solely by application requirements rather than the lack of a suitable network stack.

4 TEST CASE: A WATER IRRIGATION SYSTEM

To validate experimentally WCB in a realistic scenario, we use a water irrigation system (WIS) as our test case. A WIS is constituted by a set of pools, often a few kilometers long, connected to one other with controllable gates whose movement regulates the levels of each pool, providing customers with a relatively constant supply. Without communication between neighboring gates, each gate regulates the level of the pool immediately downstream or upstream without knowledge of what happens on the neighboring pools, in what is known as decentralized control. In [10] and [32], it is noted that decentralized control has several limitations that can waste water due to spillovers. These references suggest the use of more interconnected types of control such as centralized and distributed control architectures, in which information from neighboring pools (or all pools in the centralized case) is shared to improve control. With distances on the order of kilometers to be covered and the typical lack of existing infrastructure in these areas, WIS are one of the prototypical applications of control over multi-hop wireless networks.

Here we describe our test case, which builds on a real scenario [32]. We then present the periodic event-triggered control (PETC) design that is the basis of our experiments. It is not our intention in this paper to provide a complete solution to WIS; instead, our goal is to use this example as a proof-of-concept for the combination of ETC and WCB presented here. Therefore, we design a simple centralized state-feedback controller that captures the essence of the centralized control problem and allows us to showcase a centralized ETC solution over wireless. Control solutions considering more practical design criteria for WIS are available in, e.g., [10, 31].

4.1 System description and modeling

In our test case, we consider a WIS composed of multiple pools connected in series; a lateral view is depicted in Figure 2. The control problem is to regulate the levels of each pool to their setpoints by adjusting the position of the gates. Opening the gates increases the flow from pool $i - 1$ to pool i,
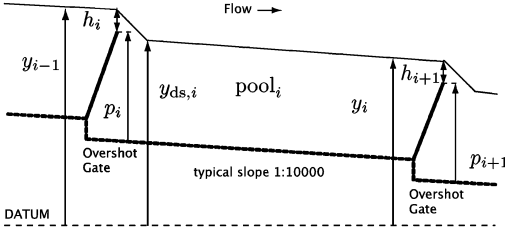
Fig. 2. A section of an open-water channel with overshot gates (from [10]).

Table 1. Parameters of the WIS models (13) and (14): delay (τ_i), surface area (α_i), and dominant wave frequency (φ_i).

Pool	1	2	3	4	5
τ_i (min)	4	2	4	4	6
α_i (m^2)	6492	2478	6084	5658	7650
φ_i (rad/min)	0.48	1.05	0.48	0.48	0.42

contributing to a reduction of level y_{i-1} and an increase of y_i. External off-take disturbances come mostly from end-users, and typically occur downstream in each pool. The control objectives w.r.t. level regulation are [10]: *i)* avoiding losses due to spillovers *ii)* keeping levels close to the setpoint to avoid oversupplying, and *iii)* preventing fluctuations occurring when dormant waves are excited.

Accurate models of open water dynamics are very complex. For control design, we can use a simpler one capturing the first modes of wave phenomena via the conservation of mass principle:

$$\pi_i\left(\frac{\mathrm{d}}{\mathrm{d}t}\right)y_i(t) = \gamma_i h_i^{3/2}(t-\tau_i) - \gamma_{i+1} h_{i+1}^{3/2}(t) - d_i(t), \tag{12}$$

where h_i is the relative height above gate i (Figure 2), d_i is the total flow of off-take disturbances, τ_i is the time for water to traverse the pool length, and γ_i is a parameter depending on the pool and gate geometry. The model dynamics are captured by a polynomial $\pi_i(\cdot)$: higher orders yield more accurate models. We assume that the flow $u_i(t) = \gamma_i h_i^{3/2}(t)$ over gate i can be directly manipulated[4], making (12) linear. For control design, a first-order polynomial π_i suffices [10, 31]

$$\alpha_i \dot{y}_i(t) = u_i(t-\tau_i) - u_{i+1}(t) - d_i(t), \tag{13}$$

where α_i is the pool surface area. However, this model is too simplistic for simulation, an integral part of the experimental setup (§5) supporting our combined evaluation of the control and network layers (§7). Therefore, as in [10], we use a third-order polynomial $\pi_i(\cdot)$ for the simulated plant:

$$\frac{\alpha_i}{\omega_{n,i}^2}\left(\dddot{y}_i(t) + 2\zeta_i \omega_{n,i} \ddot{y}(t) + \omega_{n,i}^2 \dot{y}(t)\right) = u_i(t-\tau_i) - u_{i+1}(t) - d_i(t), \tag{14}$$

where ζ_i and $\omega_{n,i}$ (satisfying $\varphi_i = \omega_{n,i}\sqrt{1-\zeta_i^2}$, for φ_i the dominant wave frequency), represent the first-mode wave damping ratio, and natural frequency of pool i respectively. In our test case, we consider a string of five pools representing a section of a water channel in New South Wales, Australia. The characteristics of this setup and related parameters (Table 1) are found in [32]. Moreover, we set the additional parameter $\zeta_i = 0.0151$ for all i, as in [54].

4.2 Event-triggered control design

For ETC design, we apply the principle of separation of concerns between control design and cyber-physical implementation. The controller is designed as a continuous-time controller, for which many methods are available. Then, a sampled-data implementation based on PETC is devised, which must consider the imperfections of the communication channel to retain some given performance specifications. This prevents changes (e.g., in network technology, topology, nodes, etc.) from

---

[4]An example of actuating device in this context is FlumeGate©, by the company Rubicon [53].
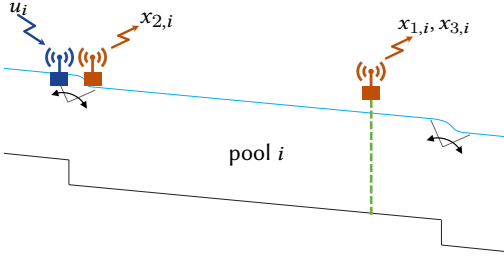
Fig. 3. Data communicated to/from nodes at pool i. The dashed green line denotes a height measurement sensor, while L-shaped gray elements denote gates with flow control and measurement capabilities.
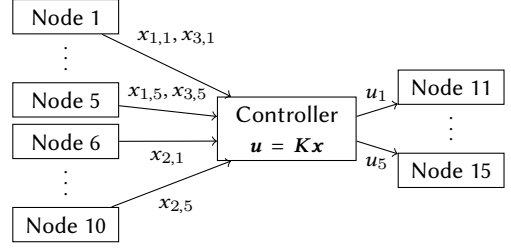

Fig. 4. Control data diagram for the 5-pool system. Each of nodes 6–10 is co-located with nodes 11–15, respectively; therefore, they can be hosted by the same physical device.

requiring a complete redesign of the controller. In our case, this is achieved with the following design procedure:

1. design a centralized state-feedback controller that rejects step disturbances;
2. select the sampling time h for monitoring and event-checking; and
3. design the distributed event-triggering parameters $\boldsymbol{M}_j, \boldsymbol{N}_j, \theta_j$ that achieve similar performance to the continuous-time controller (§2).

To design a centralized ETC for the WIS in §4.1, we need a state-space description of the system in (13). To this end, we replace the time-delay by its Padé approximation of order $(1, 1)$, as in [10], and extend the model with states $x_{3,i}$ integrating y_i, to enable rejection of persistent off-take disturbances by the controller. A state-space representation of the resulting model is given by:

$$\dot{x}_{1,i} = -\frac{1}{\tau_i} x_{2,i} - \frac{1}{\alpha_i} (u_i + u_{i+1} + d_i), \qquad \dot{x}_{2,i} = -\frac{2}{\tau_i} x_{2,i} + \frac{4}{\alpha_i} u_i, \qquad \dot{x}_{3,i} = x_{1,i}, \qquad (15)$$

where $x_{1,i} := y_i$, $x_{2,i}$ can be regarded[5] as a low-pass filter on the flow u_i, and $u_6(t) = 0$, $\forall t$, i.e., there is no controlled gate at the downstream side of the last pool. The variables $x_{2,i}$ and $x_{3,i}$ can be locally computed at the flow and height measurement nodes, respectively.

With this model, one can use standard state-space methods for control design. For our test case, we designed a linear-quadratic regulator (LQR) using diagonal weight matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$, with $\boldsymbol{R} = \boldsymbol{I}$ and $\boldsymbol{Q}$ with diagonal entries $(1250, 1250, 2500, 5000, 7500)$ for $x_{1,i}$, 0 for $x_{2,i}$, and $(1.25, 1.25, 2.5, 5, 7.5)$ for $x_{3,i}$. These values were tuned to achieve a uniform convergence across pools, a trade-off between speed of the state convergence and magnitude of control action, and robustness w.r.t. the natural frequency of oscillation of the pools. The GES decay rate (Theorem 2.1) of the continuous-time closed loop system is $\rho = 0.007$ min^{-1}.

Figure 3 illustrates how control data is communicated wirelessly. The height sensor node also performs the integration locally to compute $x_{3,i}$. The gate has one node to receive control inputs u_i and one to compute the filtered flow value $x_{2,i}$ and send it to the controller. For the 5-pool system we consider, a total of 10 sensor and 5 actuator nodes are used. The height setpoints are assumed to be locally available to the height device; hereafter, $x_{1,i} = y_i - y_i^*$, i.e., control regulates deviations of height w.r.t. its setpoint, assumed to be set constant throughout the experiment. Figure 4 shows a block diagram for the complete control system; note how the controller is a separate node.

We choose the fundamental sampling period $h = 1$ min as in [54], where this value is used for short pools up to 3200 m, as in our setup. As for ETC, we solve iteratively the LMIs in Theorem 2.1 to

---

[5]Alternatively, it can be viewed as the Padé approximant of the Smith predictor for the subsystem $\alpha_i \dot{x}_{2i} = u_i(t - \tau_i)$.

$$M_1 = \begin{bmatrix} 0.621 & 0.0030 \\ 0.003 & 0.0001 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 0.414 & 0.003 \\ 0.003 & 0.0002 \end{bmatrix}, \quad M_3 = \begin{bmatrix} 1.854 & -0.083 \\ -0.083 & 0.13 \end{bmatrix}, \quad M_4 = \begin{bmatrix} 2.48 & 0.012 \\ 0.012 & 0.001 \end{bmatrix}, \quad M_5 = \begin{bmatrix} 7.639 & 0.027 \\ 0.027 & 0.006 \end{bmatrix},$$

$$M_6 = 0.1147, \qquad M_7 = 0.0841, \qquad M_8 = 0.2337, \qquad M_9 = 0.5352, \qquad M_{10} = 1.4786,$$

$$N_1 = \begin{bmatrix} 2.5 \times 10^{-8} & 0 \\ 0 & 0 \end{bmatrix}, \quad N_2 = \begin{bmatrix} 0.0503 & 0 \\ 0 & 0 \end{bmatrix}, \quad N_3 = \begin{bmatrix} 1.2 \times 10^{-8} & 0 \\ 0 & 0 \end{bmatrix}, \quad N_4 = \begin{bmatrix} 10^{-6} & 0 \\ 0 & 0 \end{bmatrix}, \quad N_5 = \begin{bmatrix} 0.9497 & 0 \\ 0 & 0 \end{bmatrix},$$

$$N_6 = 0, \qquad N_7 = 0, \qquad N_8 = 0, \qquad N_9 = 0, \qquad N_{10} = 0,$$

$$\theta_1 = 0.415, \quad \theta_2 = 0.24, \quad \theta_3 = 0.987, \quad \theta_4 = 1.18, \quad \theta_5 = 2.15, \quad \theta_j = 9, \forall j \in \{6, ..., 10\},$$

Fig. 5. Triggering parameters applied in the test case.

find matrices M_j and N_j achieving a high sampling performance (§2.3). The triggering parameters θ_j are tuned to further improve the latter in a trade-off with steady-state error, for which a magnitude of 1 cm is deemed acceptable. Figure 5 shows the values of M_j, N_j, θ_j. Nodes 1–5 represent height sensors, with matrices partitioned according to $[x_{1,j} \ x_{3,j}]$, while nodes 6–10 represent filtered flow $(x_{2,j})$ sensors. The resulting decay rate, satisfying Theorem 2.1, is ρ=0.006 min^{-1}.

5 A CYBER-PHYSICAL EXPERIMENTAL TESTBED

A widely-adopted methodology for evaluating WNCS relies on small-scale laboratory setups mimicking industrial process control loops, e.g., the double-tank system [5, 6]. This approach tests the ability to control *real* physical processes, but often relies on single-hop networks, neglecting key networking aspects (e.g., packet delays and losses) which WCB instead explicitly addresses.

To overcome this limitation, we designed an experimental setup (Figure 6a) combining a simulated plant with a real large-scale wireless network. Its architecture is general and can be applied to systems exploring alternate control strategies and/or network stacks supporting them.

Real network, simulated plant. The plant model, implemented in MATLAB/Simulink, emulates the physical system; it receives actuator state changes as input and produces sensor readings as output. We replace the real drivers on the wireless devices with stubs interacting with the plant model, so that *i)* sensor nodes receive values from the model instead of real sensors, and *ii)* actuator nodes send the commands received from the controller to the model instead of the real actuators. Communication between the stubs and the computer running the plant model occurs out-of-band, via TCP/IP over Ethernet, to avoid interfering with the wireless network under study. The latter runs WCB unmodified, providing multi-hop communication

(a) Architecture.

(b) Block diagram of the simulation.

Fig. 6. Experimental Framework.

among sensor, actuator, and controller nodes distributed across large testbed areas. Each network node consists of a Zolertia Firefly [24], the actual embedded platform under test, connected via USB to a Raspberry Pi (RPi). The Firefly is equipped with a TI CC2538 SoC combining an ARM Cortex-M3 MCU and a 2.4GHz IEEE 802.15.4 radio. Our WCB prototype is built atop a Contiki OS port of Glossy for this SoC [23]. The RPi supports the above out-of-band channel between the Firefly board and the plant model, as well as enables the automation and remote execution of tests.
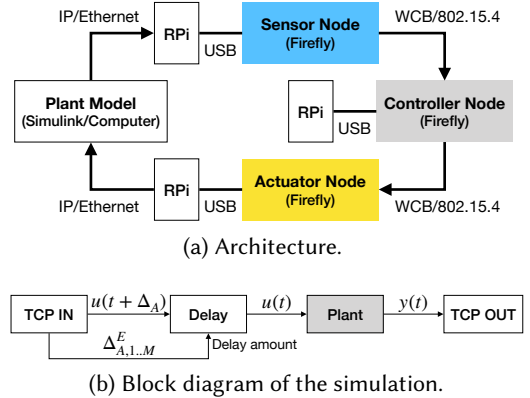
Dealing with time. For our setup to provide a realistic evaluation, it is crucial that the plant simulator, controller, and wireless network share the same notion of time. The main challenge is to realign the physical time the last two physical components rely on with the synthetic one in the plant simulator. Moreover, the out-of-band Ethernet bridging the real and simulated components is affected by random delays not present in a real system, which must be accounted for.

We address these issues as follows. First, we observe that, thanks to the synchronization inherent in WCB and other Glossy-based protocols, all wireless nodes, notably including the controller, share the same time reference with ms-level accuracy. Therefore, they can timestamp local events and perform their actions at specified instants in *global* time. Second, the joint operation of control and network is *periodic* and *structured*: *i)* (short) active periods where communication occurs are interleaved with (long) periods where the system is quiescent, and *ii)* during active periods, the interleaving of communication and control follows a well-defined pattern known a priori. Third, we leverage the presence of a simulated component to realign the physical and synthetic time references, precisely by exploiting the periodic and structured system nature. During the inactive portion of the schedule, the simulator runs at its own (faster) pace, generating the inputs to be fed to physical components at appropriate (global) times.

Figure 7 illustrates our strategy. Sensor acquisition during epoch E occurs at its start time, t_S^E. The WCB collection schedule unfolds and, after the recovery phase, the controller executes and generates the actuation commands. These are sent during the WCB dissemination phase, and received by each actuator $i \in \{1, \ldots, M\}$ at a potentially different time $t_{A,i}^E$. Once dissemination is complete, the WCB network enters sleep. During this inactive period, the actuator stubs send the received commands to the plant model over the out-of-band network, along with the reception times $t_{A,i}^E$ that, like t_S^E, are precisely timestamped, as per our first observation. These actuator states are collected at the computer running the plant model and input to Simulink, which executes the block diagram shown in Figure 6b with a simulation time synchronized with the epoch start, t_S^E. The timestamps $t_{A,i}^E$ are used to "replay" the arrival of the actuation commands u_i by taking into account the *real* delays $\Delta_{A,i}^E := t_{A,i}^E - t_S^E$. Based on this input vector $u_i(t + \Delta_{A,i}^E), i \in \{1, \ldots, M\}$, the simulator advances the model execution in the time interval $[t_S^E; t_S^{E+1}]$, generating the sensor readings for the acquisition at the beginning of the next epoch. These are sent to the stubs on the sensor nodes via the out-of-band network; when the (physical) time t_S^{E+1} arrives, the sensor nodes wake up and "acquire" these sensor readings. The process repeats in each epoch.

Nevertheless, the inactive period of the wireless network must accommodate the worst-case delays induced by model computation and Ethernet communication. Although we designed our testbed to stop upon detecting a violation of this requirement, this never happened in our experiments, where delays (<2 s) are significantly smaller than the control period (60 s). In cases where the control period is shorter than the delays, execution can be artificially slowed down by increasing the inactive period and removing the extra *empty* time in post processing. The opposite,
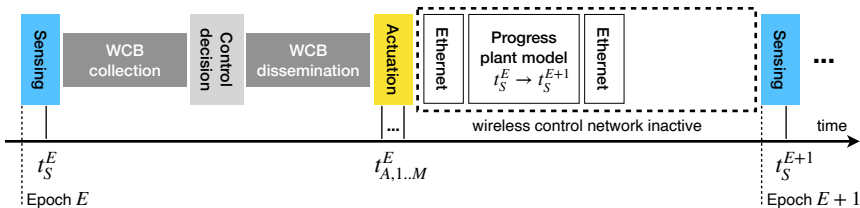

Fig. 7. Synchronous test execution with real wireless network and simulated plant.

i.e., shortening the inactive period and adding empty time in post processing, can also be done; we actually adopted this technique to speed up the execution of our experiments.

Wireless testbeds. We rely on two large-scale multi-hop wireless testbeds at our premises, called DEPT and HALL, constituted by 36 and 19 nodes, respectively. DEPT (Figure 8a) is deployed along office corridors, yielding a mostly linear topology spanning a 83×33 m^2 area; by disabling node 21–22 we enforce a 5-hop network. HALL (Figure 8b) is denser and spans a 56×30 m^2 L-shaped area; nodes in the same segment are within communication range, yielding a 2-hop network.

The role of each node (Figure 8) mimics our WIS test case (Figure 3): the actuator and flow sensor nodes of pool i are close to each another, while the height sensor is far from them, at the end of pool i and closer to the actuator and flow sensor of pool $i + 1$. Instead, the controller node position maximizes hop distance, creating a challenging topology for our evaluation.

Benefits and applicability. Our experimental setup is a contribution offering several advantages. It is *flexible*, enabling experimentation with control systems exhibiting diverse requirements and time scales by simply developing appropriate Simulink models. It is easily *replicable* and *scalable* as it does not require specific hardware components apart from mote-class and RPi-class devices; existing wireless testbeds [16, 33, 46] could easily support it. Finally, and most importantly, it fosters *repeatability*, as the control plant is simulated, hence not subject to the vagaries of a real system.

6 CONFIGURING (AND IMPROVING) THE WIRELESS CONTROL BUS

We empirically study how the parameters of WCB affect its performance, and determine the configuration used in the evaluation. This is also an opportunity to identify low-level optimizations further improving performance. Table 2 summarizes the key parameters, following the protocol description (§3). *Slot* parameters govern the behavior of a single Glossy flood, and can be tuned for each slot type. *Epoch* parameters govern the use of these slots inside the active period in each epoch. The table does not consider the number K of data collection slots T, one per sensor node, as this is an *application* parameter and therefore only known at deployment time.

Methodology. We determine the parameter values as inspired by [25]. We analyze the sensitivity of WCB to each parameter value via thousands of floods performed with the same topology, initiating nodes and packet size as in our evaluation (§7). An exception is the duration W_x of each slot type $x \in \{S, T, A, EV, CTRL\}$, determined analytically based on the corresponding number N_x of packet retransmissions and knowledge of network diameter, packet on-air duration, and Glossy delay between packet RX and TX, plus a small slack accounting for potential collisions.
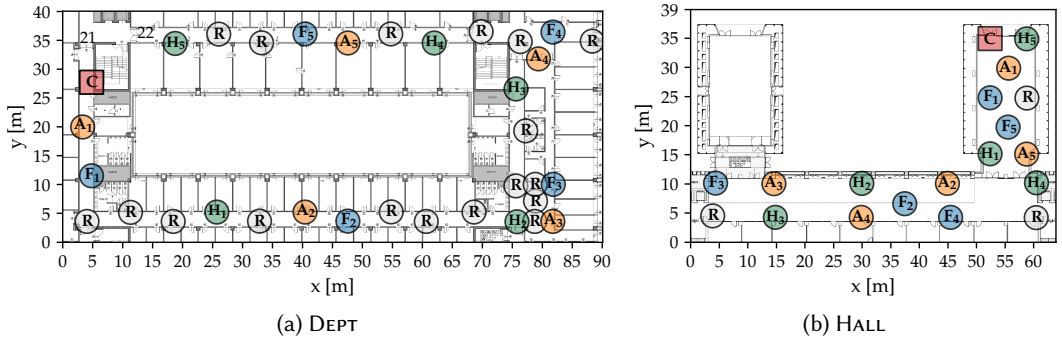

(a) DEPT   (b) HALL

Fig. 8. The wireless testbeds used in our experiments. The red square denotes the controller (C), orange circles are actuators (A), while light blue and green circles are flow (F) and height (H) sensors, respectively. Nodes acting as forwarders (R) are in grey. Nodes 21–22 are disabled to increase the network diameter.

Table 2. Protocol parameters.

Slot parameters, defined for every slot type	
W	Slot duration
N	Number of packet retransmissions within the slot
Epoch parameters	
R	Max. number of TA pairs in the recovery phase
C	Number of command dissemination slots CTRL
E	Number of event slots EV (only WCB-E)

Table 3. Reliability of the WCB configuration.

Slot type	HALL			DEPT		
	N	W	$\overline{PDR}$	N	W	$\overline{PDR}$
S	3	7	0.99996	3	10	0.99993
T	2	6	0.9994	2	9	0.99914
A	3	8	1.0	3	11	0.99994
CTRL	2	8	0.99987	2	11	0.9998

Slot parameters. Table 3 shows the configuration we select along with the corresponding mean packet delivery rate $\overline{PDR}$ for the *whole* network (opposed to the sink only). EV slots are not reported here as they are used only in WCB-E; they are analyzed at the end of the section.

A value $N \in \{2, 3\}$ ensures very good reliability; higher values increase consumption without much improvement. We select $N = 3$ for S and A slots as these are *i)* crucial to the overall reliability of WCB, and *ii)* scheduled *once* per epoch, bearing a moderate impact on energy consumption w.r.t. $N = 2$. As for T slots, they *i)* are the largest component of an epoch active portion, always present in WCB-P and dynamically triggered in WCB-E, and *ii)* benefit from the safety net of acknowledgements and retransmissions scheduled on-demand during the recovery phase. Therefore, we privilege energy consumption over reliability and use $N = 2$. Nevertheless, Table 3 shows that this value still achieves a remarkable three-nine reliability of T slots *over the entire network*.

Knowledge of this reliability enables us to estimate analytically the probability to collect at the sink *all* the K sensor readings, assuming packet loss modeled as a series of independent and identically distributed (i.i.d.) Bernoulli trials [56]. In our case (§4), this yields a probability to deliver all $K = 10$ sensor readings of 99.3% and 99.6% in HALL and DEPT, respectively. In other words, at least one reading is lost only in 4–7 epochs out of 1000. In these relatively rare cases, the recovery phase is automatically triggered, and the lost packets retrieved when needed, much more efficiently than by increasing the reliability (and consumption) of *every* T flood.

Epoch parameters. In the recovery phase, R is the number of TA pairs enabling nodes to retransmit packets not acknowledged by the sink, if any. This parameter directly affects the reliability of data collection but also the latency of actuation commands, as their dissemination is always scheduled after the maximum duration of the recovery phase (Figure 1). Hereafter, we use $R = 3$ as we verified experimentally that, in our setup, the probability to lose >3 packets in the collection phase is $<10^{-7}$.

On the other hand, the dissemination phase must also be reliable in addition to timely, as it is crucial to the control operation that actuation commands are correctly received network-wide. Nevertheless, a safety net of acknowledgments and retransmissions, akin to the one supporting many-to-one data collection traffic, would be inefficient for one-to-many dissemination. Fortunately, a simple and effective redundancy strategy where the CTRL slot containing actuation commands is always repeated C times is possible. Table 3 shows that $N = 2$ already makes it unlikely that an actuation message is lost network-wide. The probability that the packet is lost *multiple* times in a row is therefore very low; we verified empirically and analytically that the value $C = 2$ used hereafter is sufficient to obtain between 6- and 7-nine reliability in our testbeds.

Event phase. The reliability of the event phase in WCB-E is crucial to the correct and timely operation of ETC. Nevertheless, the EV slots constituting this phase have peculiar characteristics. First, they are *shared*; several sensor nodes may detect at the same time a violation of the triggering condition and decide to signal an event by concurrently transmitting in the same EV slot. Second, their reception triggers a reaction *at the sink and all sensor nodes*, signaling the need to perform a

Table 4. Reliability of the EV phase in WCB-E.

Table 5. WCB configuration in §7. The values W_x are in ms.

	N	W	U	$\overline{PDR}$		$\overline{SDR}$	
				E =1	E =2	E =1	E =2
HALL	2	4	1	0.9993	0.9999990	1.0	1.0
	2	4	2	0.992	0.99973	0.9986	0.99999
	2	4	3	0.985	0.9988	0.997	0.99994
	2	4	5	0.973	0.995	0.991	0.9995
	2	4	7	0.969	0.993	0.988	0.999
	2	4	10	0.976	0.997	0.989	0.999
DEPT	2	6	1	0.9988	0.999997	1.0	1.0
	2	6	2	0.996	0.99996	0.9994	0.999997
	2	6	3	0.993	0.99986	0.9988	0.999993
	2	6	5	0.991	0.9997	0.9984	0.99998
	2	6	7	0.987	0.9991	0.997	0.9998
	2	6	10	0.97	0.995	0.989	0.998

Parameter	HALL	DEPT
N_S	3	3
W_S	7	10
N_{EV}	2	2
W_{EV}	4	6
N_T	2	2
W_T	6	9
N_A	3	3
W_A	8	11
N_{CTRL}	2	2
W_{CTRL}	8	11
E	2	
R	3	
C	2	

collection phase. Third, as in the case of actuation commands, this traffic pattern is not amenable to acknowledgments, and therefore must rely on alternative reliability mechanisms.

Table 4 analyzes the reliability of EV slots, similarly to what reported for the other slots in Table 3, this time considering also a number U of randomly-selected sensor nodes transmitting in the same shared slot. Results show that while most of the network, including the sink, enjoys near-perfect reliability, a few nodes instead experience repeated losses. This is exacerbated as U increases, with a minimum network-wide $\overline{PDR} = 97\%$. Unfortunately, losing 3 events out of 100 is unacceptable, as it could hamper ETC performance.

A redundant strategy, similar to the one adopted for the dissemination phase, mitigates the problem; repeating the EV slot for $E = 2$ times improves reliability in all configurations and yields a minimum $\overline{PDR} = 99.3\%$. Increasing E would improve further, but also severely reduce the energy efficiency of the ETC system, as the event phase is scheduled in every epoch of WCB-E.

However, an alternative, energy-efficient technique is possible. We observe that event packets do not carry data; their mere reception is what informs nodes that an event has been reported. Consequently, instead of requiring correct reception of event packets, we consider the reception of *any* IEEE 802.15.4 frame (even corrupted ones) in an EV slot as an indication of an event detection.

The impact of this technique is beneficial, as shown in the right-hand side of Table 4, reporting the average, network-wide signal detection rate $\overline{SDR}$. Reliability is increased in all configurations, with a minimum $\overline{SDR} = 99.8\%$ with $U = 10$ senders in DEPT. Further, reliability rapidly increases as U decreases, achieving or approaching 5 nines. In practice, in our representative test case the number of sensors concurrently detecting events is <1.2 on average, and always <6.

On the other hand, relying on corrupted packets in the EV slot may lead nodes to falsely presume an event has been detected, wasting energy by incorrectly triggering data collection. We verified empirically both in our dedicated experiments as well as in the overall evaluation (§7) that the rate of these false positives is <0.003%, bearing a negligible impact on energy consumption.

Table 5 summarizes the configuration used in the evaluation.

7 ETC OVER WCB: A TESTBED EVALUATION

We now ascertain the ability of WCB to efficiently support ETC by fulfilling its peculiar requirements in terms of reliability and latency, necessary to a correct and efficient control, while retaining the

energy savings enabled by ETC adaptive sampling. To offer a concrete and complete application of ETC over WCB, we focus on the WIS test case and execute in our cyber-physical testbed (§5) the control strategy we outlined (§4) atop the WCB-E variant properly configured (§6). Each experiment has a duration of one full day (1440 epochs) of simulated time, repeated multiple times.

We compare against periodic control over WCB-P. Although a comparison of the latter against the state of the art in networking for periodic control is outside the scope of this paper, we argue that WCB-P is more performant than the existing CTX-based solutions we survey in §8—themselves outperforming conventional ones—due to the different design and reliability mechanisms, whose beneficial impact we show here. In any case, given that WCB-P is essentially a degenerate case of WCB-E (§3.3) our choice compares both control strategies against the same protocol framework, elucidating the key differences without the bias a completely different network stack would induce.

7.1 Control Performance

Each simulated day starts with $x_{1,i} = 0.05$ m, $x_{2,i} = x_{3,i} = 0$ m for each pool i and no disturbance. Off-take step disturbances are added at pool 5 as in [32]: $0 \rightarrow 16$ m^3/min at minute 180, $16 \rightarrow 34$ m^3/min at 450, and $34 \rightarrow 0$ m^3/min at 600. As the system has time to settle in between and after disturbances, we observe it both in steady state and during transient, when perturbed.

We consider *i)* an ideal scenario where sensors yield perfect readings, and *ii)* one where independent normally-distributed pseudo-random white noise is added to both level and flow measurements, with zero mean and standard deviation of 0.001 m and 1 m^3/min, respectively. In the ideal scenario, the only source of randomness is the network, allowing us to isolate the impact of the protocol stack on control performance. In the second scenario, the added noise introduces variability (and degradation) of the ETC sampling performance, enabling a more realistic assessment.

Metrics. We focus on the number of samples generated as well as on two metrics based on the *integral average error* (IAE) of a signal $x(t)$ w.r.t. its reference x^*

$$\text{IAE}(x, x^*, T_{exp}) := \frac{1}{T_{exp}} \int_0^{T_{exp}} |x(t) - x^*| dt. \tag{16}$$

This standard control performance metric measures the accumulated tracking error; the smaller its value, the faster states converge to their references. In our case $T_{exp} = 1440$ minutes, the duration of the experiments. Since height references are already accounted for in the variables $x_{1,i}$, we set $x^* = 0$, yielding the metrics $\text{IAE}_i := \text{IAE}(x_{1,i}, 0, T_{exp})$. For each simulation, we compute the sums and maxima of IAEs over the pools, with the following shortened notations:

$$\text{IAE}_{\Sigma} := \sum_{i=1}^{5} \text{IAE}_i, \quad \text{IAE}_{\max} := \max_{i \in \{1,\dots,5\}} \text{IAE}_i, \tag{17}$$

Results. The pool heights follow a similar trajectory under both control strategies (Figure 9, top) and with a similar performance in reference tracking (Table 6), confirming the desirable property that ETC yields essentially the same control output of periodic control. However, *ETC generates significantly fewer samples than periodic control*, almost 90% less in the ideal scenario and only slightly more, 87% less on average, with measurement noise (Table 6). The sample pattern for ETC (Figure 9, bottom) highlights that, as expected, sampling is more frequent when transients are stronger, and becomes sporadic as the system approaches steady state.

It is important to remark that the savings ETC can provide w.r.t. periodic control are highly dependent on the control problem at hand, as the average PETC sampling frequency depends in non-trivial ways on the system dynamics, control design, and triggering mechanism. The formal computation of this value has only recently been made possible [15]. Nevertheless, we observe that
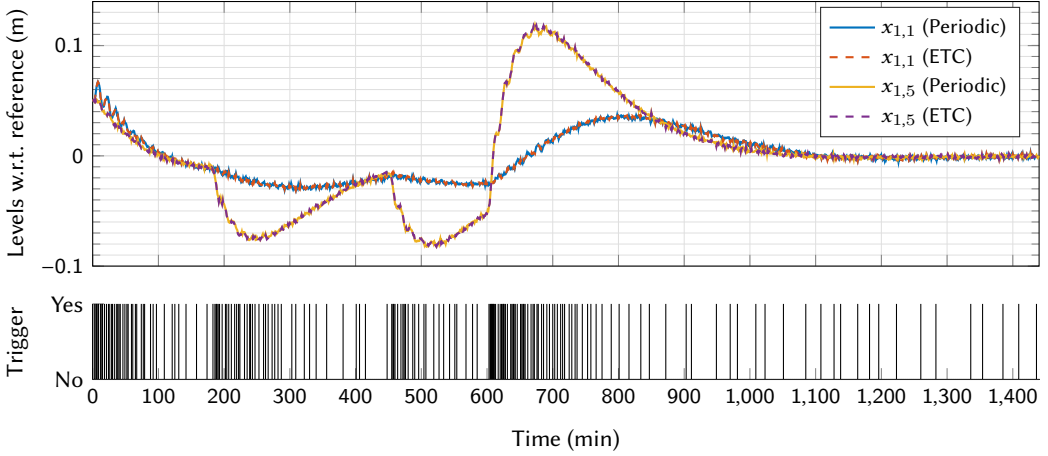
Fig. 9. ETC vs. periodic control, both over WCB in DEPT. Top: Level w.r.t. reference for the 1st and 5th pools, 1-day executions with measurement noise. Bottom: Sampling instants for the ETC case.

Table 6. Sampling and control performance metrics from experiments: mean (standard deviation when different from 0) over 8 executions of 1 day of plant operations each.

Scenario	Testbed	Sampling	Sample count	$IAE_\sum$ (m)	IAE_{max} (m)
Without noise	HALL	ETC	149	0.1084	0.03283
		Periodic	1440	0.1085 ($<10^{-6}$)	0.03293 ($<10^{-6}$)
	DEPT	ETC	148	0.1088 ($<10^{-6}$)	0.03286
		Periodic	1440	0.1085 ($<10^{-6}$)	0.03293 ($<10^{-6}$)
With noise	HALL	ETC	186.1 (5.743)	0.1091 (1.21×10^{-4})	0.03311 (6.1×10^{-5})
		Periodic	1440	0.1088 (3.8×10^{-5})	0.033 (2.1×10^{-5})
	DEPT	ETC	185.4 (4.984)	0.109 (1.39×10^{-4})	0.03308 (4.7×10^{-5})
		Periodic	1440	0.1088 (3.8×10^{-5})	0.033 (2.1×10^{-5})

the *variation* of sample count across experiments, captured by the standard deviation (Table 6), appears in ETC only in the scenario with measurement noise and is completely absent in the ideal one. This is a witness of the consistent performance of WCB-E in terms of reliability and latency, analyzed next: *practical control aspects like measurement noise induce significantly higher variations in ETC sampling than the vagaries of the wireless communication.* Notably, this enables a desirable separation of concerns during system development, as the assessment of the benefits ETC provides, and therefore the decision on whether or not to employ it for the specific case at hand, can be performed accurately and entirely during the control design phase.

7.2 Network Performance

The reliability of event detection, sensor reading collection and command dissemination, together with the actuation latency, are crucial to the control performance we observed.

Table 7 reports the average of these metrics across 16 test runs, i.e., 1440×16=23040 epochs in each row. WCB achieves zero packet losses regardless of the functionality, protocol variant, and testbed considered, confirming the effectiveness of its strategy (§3) and configuration (§6). Recovery mechanisms are key to achieve this result. Log inspection shows that, for data collection, they

Table 7. Performance of WCB in hardware-in-the-loop testbed experiments: mean (and standard deviation when non-zero) over 16 executions of 1440 epochs each, i.e., 1 day of plant operation.

Testbed	Protocol	Event detection reliability [%]	Data collection reliability [%]	Actuation reliability [%]	Latency of actuation commands [ms]
Hall	WCB-E	100	100	100	192.021 (0.04)
	WCB-P	—	100	100	180.023 (0.02)
Dept	WCB-E	100	100	100	253
	WCB-P	—	100	100	237.017 (0.012)

are triggered ~1% of the times; while small in absolute terms, this fraction of lost packets, if not recovered, would make ETC trigger more often than needed and potentially degrade performance.

These losses, mainly caused by interference from WiFi access points and devices in the indoor office spaces where our testbeds are deployed, are effectively and efficiently mitigated by our use of channel hopping. Our technique is directly inspired by work on Crystal [26] where it has been shown capable to withstand significantly stronger interference. Further, *all* of the top-three winning systems (including Crystal) in the 2018 and 2019 editions of the EWSN Dependability Competition rely on some form of channel hopping to overcome its nearly unreasonable noise levels. Therefore, while we cannot offer an evaluation of WCB under strong interference as in [26], prohibitive both in terms of testbed logistics and text limitations, we incorporate in our system the state-of-the-art techniques for interference resilience, an aspect entirely neglected by existing network stacks for ETC [5, 17, 27, 50] whose other shortcomings we discuss in §8. Moreover, although the sparser and aperiodic traffic induced by ETC increases the importance of each packet, this control strategy is intrinsically resilient to packet loss. When this occurs, the effect is simply the triggering of more events due to incomplete information at the controller; this transiently impacts energy consumption but not the correctness of control, as long as the required sensor readings are delivered at the controller within a maximum tolerable delay.

The latency between the beginning of an epoch and the delivery of the *last* actuation command is also very small, especially if compared to the sampling period (hundreds of ms vs. 60 s). Further, it has minimal jitter, as commands usually reach actuators in the first CTRL slot. Interestingly, the different network diameter of the two testbeds induces an inevitable difference in the latency of actuation commands. Although this difference is very small (<61 ms) w.r.t. the system dynamics (hours), the ETC sampling patterns are known to be sensitive to small perturbations over the long run; however, the net effect is only a small difference in the ETC sample count (Table 6).

Finally, as expected, WCB-E is slightly slower (~6.7%) than WCB-P due to the additional event detection phase, although the absolute difference is negligible w.r.t. the sampling period and does not affect the control output, as already mentioned (Figure 9, Table 6).

7.3 Energy Consumption

The wireless transceiver is notoriously the most power-hungry component in networked embedded systems, and the one whose contribution ETC seeks to minimize. Therefore, we compare ETC vs. periodic control in terms of the radio duty-cycle $DC = \frac{T_{on}}{T_{exp}}$, i.e., the per-node radio-on time over experiment duration, a metric commonly accepted as a reliable proxy for energy consumption.

Key finding. Table 8 confirms that our embodiment of ETC consumes significantly less than periodic control—one of our goals. The reason lies precisely in the *interplay* between ETC and the network stack supporting its operation, WCB-E. By design, ETC abates traffic by triggering sensor data transmissions only when needed for control. In our test case, >89% of the periodic samples are

Table 8. Sampling and duty-cycle performance of ETC and periodic control vs. presence of measurement noise. Results are average percentages over 8 executions of 1440 epochs each, i.e., 1 day of plant operation.

Testbed	Control	No measurement noise			With measurement noise		
		DC	reduction		DC	reduction	
			sampling	DC		sampling	DC
Hall	ETC	0.0319	89.65	67.84	0.0341	87.02	65.45
	Periodic	0.0992			0.0987		
Dept	ETC	0.0413	89.72	64.58	0.0438	87.13	62.47
	Periodic	0.1166			0.1167		

Table 9. Average per-epoch radio-on time T_{on} and duty-cycle DC without measurement noise. Values are the average over 8 executions of 1440 epochs each, i.e., 1 day of plant operation.

Testbed	Metric	WCB-E					WCB-P
		No event	Event	Transient	Steady state	1 day	1 day
		detected	detected	(600−750)	(1000−1440)	(0−1440)	(0−1440)
Hall	T_{on} [ms]	13.81	65.51	29.58	14.40	19.16	59.50
	DC [%]	0.0230	0.1092	0.0493	0.0240	0.0319	0.0992
Dept	T_{on} [ms]	18.82	76.93	36.58	19.60	24.79	69.98
	DC [%]	0.0314	0.1282	0.0610	0.0327	0.0413	0.1166

suppressed in the ideal case, and >87% in the noisy one. In general, this traffic suppression does not automatically translate in energy savings. Nevertheless, WCB-E minimizes consumption when the system is in steady state while ensuring timely and reliable communication when required to support control. In our case, this yields a DC reduction >62%, with marginal differences in the two testbeds due to their different network diameter. Therefore, *WCB-E effectively translates the significant reduction of control traffic achieved by ETC into corresponding savings in energy consumption.* This is a significant leap forward w.r.t. state-of-the-art ETC literature [20, 21, 39, 48, 52] whose energy reduction is hampered by inefficient protocols and limited to small-scale star topologies.

Dissecting the energy contribution. Figure 10 highlights where energy savings arise from, by comparing the average DC per epoch of WCB-P and WCB-E across one day of plant operation. The behavior of the periodic controller is invariant w.r.t. system conditions. Therefore, WCB-P must acquire sensors readings and disseminate actuation commands in *every* epoch, resulting in a nearly-constant DC; the small spikes correspond to occasional recovery phases. In contrast, the adaptive ETC controller triggers communication via WCB-E only when needed. This results in a pattern similar to Figure 9, although here we focus on the ideal case as it simplifies observations concerned with communication by separating them from measurement noise. After the initial settling phase,
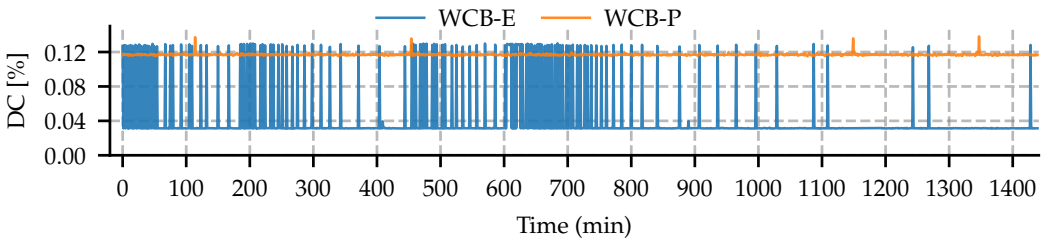

Fig. 10. Comparison of the average network duty-cycle per-epoch of WCB-E and WCB-P during one day of plant operations in Dept in absence of measurement noise.
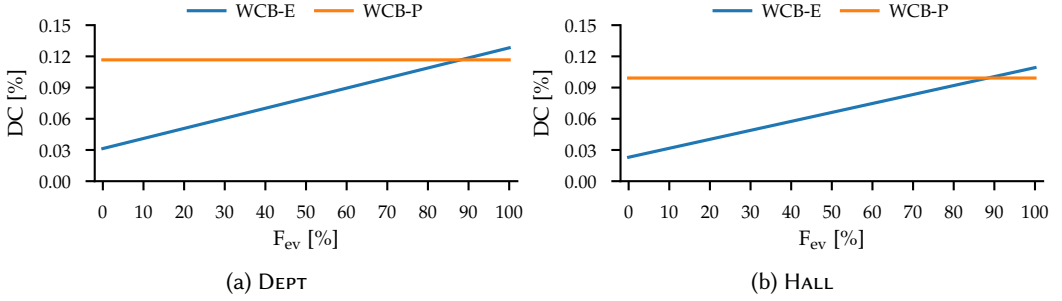
Fig. 11. Comparison of WCB-P and WCB-E vs. the frequency of epochs with events, in both testbeds.

Figure 10 clearly shows how DC increases in conjunction with off-take step disturbances (minute 180, 450, and 600) and reduces when the system approaches stability (1000−1440).

Table 9 offers additional insights on T_{on} and DC, by comparing the invariant control operation of WCB-P against the various stages of ETC operation over WCB-E. In epochs where no event is detected WCB-E saves 73.1% and 76.8% w.r.t. WCB-P in DEPT and HALL, respectively. Energy is minimized by putting the network to sleep right after the EV phase (Figure 1). Otherwise, when an event is detected WCB-E is slightly more active ($\leq 10.1\%$) due to the extra EV slots.

Generalizing to other scenarios. These results show how the efficiency of ETC over WCB-E ultimately depends on how frequently the triggering condition is violated. As long as events are relatively *rare*, the energy savings in steady-state outweigh the extra cost of the EV phase.

System designers must ascertain this tradeoff in the early stages of development, to select the most appropriate control strategy and the corresponding network stack supporting it. Luckily, analytical models for the energy consumption of both WCB variants can be easily derived, as all nodes follow the same, global, periodic schedule. Once the average network-wide radio-on time $t_{on,X}$ of each slot type is estimated as in [25] and §6, the overall per-epoch radio-on time $T_{on,P}$ of WCB-P is simply the sum of $t_{on,X}$ across slots in each protocol phase, invariant w.r.t. event detection. The one for WCB-E is then derived as:

$$T_{on,E} = F_{ev} \times (T_{on,P} + E \times t_{on,EV}) + (1 - F_{ev}) \times (t_{on,S} + E \times t_{on,EV})$$

where F_{ev} is the average frequency of epochs with at least one event and E the number of EV slots (§6). DC is computed for both cases by dividing the radio-on time by the epoch duration T_{epoch}.

Figure 11 exemplifies the tradeoffs at stake by reusing the parameters from our evaluation except for the frequency F_{ev}, whose value here is varied to represent, in an abstract setting, the DC resulting from several hypothetical control problems. The charts show how, in these conditions, periodic control over WCB-P becomes preferable vs. ETC over WCB-E only when $F_{ev} \geq 90\%$; the latter enables energy savings even when F_{ev} approaches this break-even point. For instance, when $F_{ev} \approx 70\%$, DC is reduced by nearly 15%, which becomes 25% when $F_{ev} \approx 60\%$, still extending system lifetime significantly. Overall, this confirms that ETC over WCB-E supports a wide range of real-world control problems and systems where it unlocks remarkable energy savings, ultimately pushing the envelope of the application of cyber-physical systems to untethered scenarios.

Implications of epoch duration selection. The value of T_{epoch} is a crucial parameter that determines a trade-off between control responsiveness and energy consumption. From a control design standpoint, T_{epoch} should be as small as possible to achieve the best control performance; however, from a communication standpoint, this causes a corresponding increase in duty cycle for both

Table 10. Effects of the epoch duration on duty cycle, for ETC and periodic control. The ETC savings in DC relative to periodic are computed as $1 - \frac{DC_{ETC}}{DC_{periodic}}$.

T_{epoch} (s)	# Events	# Epochs	F_{ev} (%)	DC_{ETC} (%)	$DC_{periodic}$ (%)	ETC savings in DC (%)
60	187	1440	13	0.034	0.099	65.7
45	195	1920	10.1	0.042	0.132	68.2
30	211	2880	7.3	0.059	0.198	70.2
15	234	5760	4	0.106	0.397	73.3
5	237	17280	1.4	0.290	1.190	75.6
1	268	86400	0.3	1.397	5.950	76.5

ETC and periodic control. Another aspect to be considered is that a small T_{epoch} typically leads to fewer events generated in the epoch, i.e., a smaller F_{ev}, increasing the relative benefit of ETC w.r.t. periodic sampling (Figure 11). Table 10 offers a concrete example of these tradeoffs by showing how duty cycle changes in our scenario with an epoch duration smaller than the value $T_{epoch} = 60$ s used here. The values of DC for ETC and periodic control are estimated from simulations using the model for DEPT in Figure 11. As shown in the table, when aiming to minimize energy consumption, a general guideline would be to set T_{epoch} to the highest value ensuring that control performance is within specifications. This is the criterion we adopted here, setting the value $T_{epoch} = 60$ s to match the fundamental sampling period h recommended in the literature [54] (§4.2), which guarantees a control performance within 1% of the nominal continuous-time one.

8 RELATED WORK

The adaptive control strategy of ETC raised a lot of interest in the last decade, with several researchers tackling the design of new triggering conditions and other strategies to reduce communication further [20, 52], improve applicability on digital platforms [21], and decentralize triggering conditions [39]. An overview of the state of the art in ETC can be found in [22, 40].

However, the benefits unleashed in theory by ETC must be confirmed in practice by real-world testbeds. This is true in general [35] and even more poignant for ETC, given the peculiar challenges it poses to communication (§1, [8]). Unfortunately, only few works investigate ETC performance via prototypes. These use IEEE 802.15.4 [5, 27], WiFi [50], or G5 (IEEE 801.11.p) [17], but always in a single-hop topology with at most 5 nodes, hardly representative of staple real-world use cases for WNCS.

In contrast, the work described here is validated with a realistic setup that combines a model of the system under control with a real, multi-hop low-power wireless network, yielding a significant level of realism to the evaluation. These testbeds are unfortunately rare in the literature. The closest is the one proposed in [36], featuring a similar combination of modeled system and real network. Nevertheless, the concise description does not detail if and how network-induced random delays are mitigated; further, it relies on the PTP protocol for time synchronization, requiring dedicated, expensive hardware. In contrast, our testbed explicitly targets random delays with an architecture (§5) that, in addition, provides the extra flexibility to speed up or slow down the real-time execution. Moreover, it uses commonplace devices and is therefore easily replicable by other researchers.

Apart from providing a realistic evaluation, in this paper we have tackled the crux of the matter by proposing a network stack expressly targeting the peculiar traffic patterns and requirements induced by ETC. For these, the stacks commonly used in industrial control, e.g., WirelessHART [1], ISA100.11.a [2], 6TiSCH [51], do not offer the necessary guarantees in terms of timeliness, reliability, and energy-efficiency, especially in multi-hop configurations. Research proposals exist that cater for dynamically changing traffic demands, crucial in ETC; however, they are based on expensive

centralized or distributed scheduling (e.g., as in DistributedHART [41]) or unrealistic assumptions about link reliability (e.g., as in D^2-PaS [55]). In contrast, CTX-based stacks do not require explicit routing, per-link scheduling, or continuous link monitoring. This enables excellent performance along the three performance dimensions above but has also been exploited to adapt to dynamically changing traffic demands [25, 49]. This state of the art directly motivates our work and specifically the use of CTX.

In this respect, the design of WCB is inspired by two systems: the Low-power Wireless Bus (LWB) [18] and Crystal [25]. The former was the first to make explicit the potential of CTX for abstracting communication into a network-wide bus, generating several follow-up variants. For instance, Blink [57] targets hard real-time communications by equipping LWB with a real-time scheduler based on earliest deadline first. eLWB extends LWB with the ability to handle events, as a side contribution of a more general architecture targeting an acoustic emission monitoring system [47]. In eLWB, the reaction to the event is centralized at the controller, while in WCB it is decentralized at sensor nodes, yielding lower latency. Further, eLWB focuses on monitoring rather than control, without dedicated reliability mechanisms, crucial in ETC and discussed later.

LWB has been exploited also specifically for control. The system in [7] supports feedback control, stability guarantees, and mode changes over multi-hop wireless networks for systems with fast dynamics (tens of ms). Latency is therefore the main focus rather than reliability, for which dedicated mechanisms are not provided. The paper exploits a periodic controller. Another work by the same group explores instead self-triggered control [9] where, contrary to ETC, nodes *predict* when they expect to trigger an event; this information is exploited to reserve the required communication slots with LWB. Self-triggered control is also studied in [36], and compared against rate adaptation; in both control strategies, the necessary communication is provided by a variant of LWB.

The aperiodic, unpredictable communication patterns of ETC are significantly more challenging than the pre-defined or predictable ones induced by periodic and self-triggered control. ETC in principle enables minimal network overhead during quiescent, steady-state periods, but also demands both timely and reliable communication otherwise, to guarantee correctness and performance. LWB does not cater for aperiodic communication, let apart guaranteeing its conflicting requirements w.r.t. timeliness, reliability, and energy-efficiency, as we do instead in WCB. Therefore, none of the stacks above, directly built atop LWB support these requirements either; further, none of them provides dedicated mechanisms *expressly* targeting reliability, as in our case.

Instead, these conflicting requirements have been reconciled in Crystal [25, 26]. Aperiodic communication "makes each packet count", as it is transmitted unpredictably and sporadically, implicitly carrying more information. Crystal focuses on data collection and exploits the capture effect to support concurrent, reliable transmission of sensor readings, individually acknowledged by a Glossy flood. This pattern directly inspires the T and A slots in WCB, where they are combined differently. In Crystal, concurrent senders are a priori unknown; in the worst case where all U nodes transmit, at least $2U$ Glossy floods are required. In WCB, data collection occurs only if and when an event signaling a violation of the ETC triggering condition is disseminated. As this occurs reliably and in a distributed fashion, it eliminates contention and triggers collection, always from *all* sensor nodes, using only $U + 1$ floods. The recovery phase, reminiscent of the TA pairs of Crystal, must therefore retrieve only an occasional missed packet, rather than all competing ones, limiting overhead and bounding the recovery duration, crucial for predictable control operation.

9 CONCLUSIONS AND FUTURE WORK

We presented the Wireless Control Bus (WCB), the first network stack efficiently supporting the peculiar communication requirements induced by ETC. Unlike the few prototypes reported in the literature, WCB expressly targets multi-hop, low-power wireless networks, and advances the state

of the art by significantly reducing the gap between communication savings and energy savings—a well-known issue hampering ETC adoption. We design a centralized state feedback controller using a novel, modified decentralized periodic ETC suited for step disturbance rejection, combine it with WCB, and evaluate its performance in network-in-the-loop setups emulating a 15-state water irrigation system over a real-world multi-hop network. Our results show that w.r.t. periodic control, also implemented over WCB: *i)* ETC reduces samples by >87%, translated by WCB into energy savings >62%, and *ii)* control performance is essentially equivalent in the two strategies and consistent across experiments, witnessing the extreme dependability of the network layer.

We intend to release publicly WCB as open source. We believe that the availability and performance of WCB, unlocking the full potential of ETC, may fuel new research on this topic. Our own agenda includes exploring the combination of WCB with other decentralized ETC frameworks [21], implementing theta-adaptation [39], and using traffic models [30] to further reduce energy consumption by scheduling longer periods of sensor node sleep, along the lines of [14]. Concerning our test case of water irrigation systems, we are working on alternate control architectures, like the robust output-feedback controllers in [32], and developing a testbed using a scaled-down irrigation channel, to investigate other practical aspects of wireless ETC. Finally, the exploitation of CTX on radios other than IEEE 802.15.4 opens intriguing opportunities. For instance, an ultra-fast data collection layer has recently been proposed for ultra-wideband (UWB) radios [49], whose adaptation to the ETC context could potentially unlock additional performance improvements.

ACKNOWLEDGMENTS

REFERENCES

[1] 2007. *HART Field Communication Protocol Specification.* Revision 7.0. HART Communication Foundation.

[2] 2009. *Wireless Systems for Industrial Automation: Process Control and Related Applications, ISA-100.11a.* Standard. International Society of Automation.

[3] K. J. Åström and B. Bernhardsson. 2002. Comparison of Riemann and Lebesgue Sampling for First Order Stochastic systems. In *Proc. of the 41st IEEE Conference on Decision and Control (CDC).*

[4] A. Ahlen, J. Akerberg, M. Eriksson, A. J. Isaksson, T. Iwaki, K. H. Johansson, S. Knorn, T. Lindh, and H. Sandberg. 2019. Toward Wireless Control in Industrial Process Automation: A Case Study at a Paper Mill. *IEEE Control Systems Magazine* 39, 5 (2019), 36–57.

[5] J. Araújo, M. Mazo Jr., A. Anta, P. Tabuada, and K. H. Johansson. 2014. System Architectures, Protocols and Algorithms for Aperiodic Wireless Control Systems. *IEEE Transactions on Industrial Informatics* 10, 1 (2014), 175–184.

[6] J. Araújo, A. Anta, M. Mazo, J. Faria, A. Hernandez, P. Tabuada, and K. H. Johansson. 2011. Self-triggered Control over Wireless Sensor and Actuator Networks. In *Proc. of Int. Conf. on Distributed Computing in Sensor Systems and Workshops (DCOSS).*

[7] D. Baumann, F. Mager, R. Jacob, L. Thiele, M. Zimmerling, and S. Trimpe. 2019. Fast Feedback Control over Multi-Hop Wireless Networks with Mode Changes and Stability Guarantees. *ACM Trans. on Cyber-Physical Systems* 4, 2 (2019).

[8] D. Baumann, F. Mager, U. Wetzker, L. Thiele, M. Zimmerling, and S. Trimpe. 2020. Wireless Control for Smart Manufacturing: Recent Approaches and Open Challenges (to appear). *Proc. IEEE* (2020), 1–27.

[9] D. Baumann, F. Mager, M. Zimmerling, and S. Trimpe. 2020. Control-Guided Communication: Efficient Resource Arbitration and Allocation in Multi-Hop Wireless Control Systems. *IEEE Control Systems Letters* 4, 1 (2020), 127–132.

[10] M. Cantoni, E. Weyer, Y. Li, S. K. Ooi, I. Mareels, and M. Ryan. 2007. Control of Large-scale Irrigation Networks. *Proc. of the IEEE* 95, 1 (2007), 75–91.

[11] R. Cardell-Oliver, K. Smettem, M. Kranz, and K. Mayer. 2004. Field Testing a Wireless Sensor Network for Reactive Environmental Monitoring. In *Proc. of the Intelligent Sensors, Sensor Networks and Information Processing Conf (ISSNIP).*

[12] M. Ceriotti, M. Corrà, L. D'Orazio, R. Doriguzzi, D. Facchin, S. T. Gună, G. P. Jesi, R. L. Cigno, L. Mottola, A. L. Murphy, M. Pescalli, G. P. Picco, D. Pregnolato, and C. Torghele. 2011. Is There Light at the Ends of the Tunnel? Wireless Sensor Networks for Adaptive lighting in Road Tunnels. In *Proc. of the Int. Conf. on Information Processing in Sensor Networks (IPSN).*

[13] T. Chang, T. Watteyne, X. Vilajosana, and P. H. Gomes. 2019. Constructive Interference in 802.15.4: A Tutorial. *IEEE Communications Surveys Tutorials* 21, 1 (2019), 217–237.

[14] G. de A. Gleizer and M. Mazo Jr. 2020. Scalable Traffic Nodels for Scheduling of Linear Periodic Event-triggered Controllers. *21st IFAC World Congress* (2020). https://arxiv.org/abs/2003.07642.

[15] G. de A. Gleizer and M. Mazo Jr. 2021. Computing the Sampling Performance of Event-triggered Control. In *Proc. of the 24th Int. Conf. on Hybrid Systems: Computation and Control* (Nashville, TN, USA) *(HSCC '21)*. ACM, Article 20, 7 pages.

[16] M. Doddavenkatappa, M. C. Chan, and A. L. Ananda. 2011. Indriya: A Low-cost, 3D Wireless Sensor Network Testbed. In *Proc. of the Int. Conf. on Testbeds and Research Infrastructures (TridentCom)*.

[17] V. Dolk, J. Ploeg, and W. Heemels. 2017. Event-triggered Control for String-stable Vehicle Platooning. *IEEE Transactions on Intelligent Transportation Systems* 18, 12 (2017), 3486–3500.

[18] F. Ferrari, M. Zimmerling, L. Mottola, and L. Thiele. 2012. Low-power Wireless Bus. In *Proc. of ACM Conf. on Embedded Network Sensor Systems (SenSys)*.

[19] F. Ferrari, M. Zimmerling, L. Thiele, and O. Saukh. 2011. Efficient Network Flooding and Time Synchronization with Glossy. In *Proc. of the Int. Conf. on Information Processing in Sensor Networks (IPSN)*.

[20] Antoine Girard. 2014. Dynamic Triggering Mechanisms for Event-triggered Control. *IEEE Trans. Automat. Control* 60, 7 (2014), 1992–1997.

[21] W. Heemels, M. Donkers, and A. Teel. 2013. Periodic Event-triggered Control for Linear Systems. *IEEE Trans. Automat. Control* 58, 4 (2013), 847–861.

[22] W. Heemels, K. H. Johansson, and P. Tabuada. 2012. An Introduction to Event-triggered and Self-triggered Control. In *Proc. of the IEEE Conf. on Decision and Control (CDC)*.

[23] K. C. Hewage, S. Raza, and T. Voigt. 2017. Protecting Glossy-Based Wireless Networks from Packet Injection Attacks. In *Proc. of the Int. Conf. on Mobile Ad Hoc and Sensor Systems (MASS)*.

[24] Zolertia Inc. 2020. Zolertia Firefly. Online: https://zolertia.io/product/firefly [Accessed: 18-12-2020].

[25] T. Istomin, A. L. Murphy, G. P. Picco, and U. Raza. 2016. Data Prediction + Synchronous Transmissions = Ultra-low Power Wireless Sensor Networks. In *Proc. of the ACM Conf. on Embedded Network Sensor Systems (SenSys)*.

[26] T. Istomin, M. Trobinger, A. L. Murphy, and G. P. Picco. 2018. Interference-resilient Ultra-low Power Aperiodic Data Collection. In *Proc. of the 17th ACM/IEEE Int. Conf. on Information Processing in Sensor Networks (IPSN)*.

[27] S. Kartakis, A. Fu, M. Mazo Jr., and J. A. McCann. 2018. Communication Schemes for Centralized and Decentralized Event-triggered Control Systems. *IEEE Transactions on Control Systems Technology* 26, 6 (2018), 2035–2048.

[28] K. Khakpour and M. H. Shenassa. 2008. Industrial Control using Wireless Sensor Networks. In *Proc. of the 3rd Int. Conf. on Information and Communication Technologies: From Theory to Applications*.

[29] Hassan K. Khalil. 2001. *Nonlinear Systems* (3 ed.). Pearson.

[30] A. S. Kolarijani and M. Mazo Jr. 2016. A Formal Traffic Characterization of LTI Event-triggered Control Systems. *IEEE Transactions on Control of Network Systems* 5, 1 (2016), 274–283.

[31] Y. Li and M. Cantoni. 2008. Distributed Controller Design for Open Water Channels. *IFAC Proceedings Volumes* 41, 2 (2008), 10033–10038.

[32] Y. Li and B. de Schutter. 2011. Stability and Performance Analysis of an Irrigation Channel with Distributed Control. *Control Engineering Practice* 19, 10 (2011), 1147–1156.

[33] R. Lim, F. Ferrari, M. Zimmerling, C. Walser, P. Sommer, and J. Beutel. 2013. FlockLab: A Testbed for Distributed, Synchronized Tracing and profiling of Wireless Embedded Systems. In *Proc. of Int. Conf. on Information Processing in Sensor Networks (IPSN)*.

[34] T. Liu and Z.-P. Jiang. 2015. A Small-gain Approach to Robust Event-triggered Control of Nonlinear Systems. *IEEE Trans. Automat. Control* 60, 8 (2015), 2072–2085.

[35] C. Lu, A. Saifullah, B. Li, M. Sha, H. Gonzalez, D. Gunatilaka, C. Wu, L. Nie, and Y. Chen. 2016. Real-time Wireless Sensor–Actuator Networks for Industrial Cyber-physical Systems. *Proc. IEEE* 104, 5 (2016).

[36] Y. Ma and C. Lu. 2018. Efficient Holistic Control over Industrial Wireless Sensor-Actuator Networks. In *Proc. of IEEE International Conference on Industrial Internet (ICII)*.

[37] F. Mager, D. Baumann, R. Jacob, L. Thiele, S. Trimpe, and M. Zimmerling. 2019. Feedback Control Goes Wireless: Guaranteed Stability over Low-power Multi-hop Networks. In *Proc. of the Int. Conf. on Cyber-Physical Systems (ICCPS)*.

[38] J. Malmborg and J. Eker. 1997. Hybrid Control of a Double Tank System. In *Proc. of Int. Conf. on Control Applications*.

[39] M. Mazo Jr. and P. Tabuada. 2011. Decentralized Event-triggered Control over Wireless Sensor/Actuator Networks. *IEEE Trans. Automat. Control* 56, 10 (2011), 2456–2461.

[40] M. Miskowicz. 2018. *Event-based Control and Signal Processing*. CRC Press.

[41] Venkata Prashant Modekurthy, Abusayeed Saifullah, and Sanjay Madria. 2019. DistributedHART: A Distributed Real-Time Scheduling System for WirelessHART Networks. In *2019 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*.

[42] G. P. Picco, D. Molteni, A. L. Murphy, F. Ossi, F. Cagnacci, M. Corrà, and S. Nicoloso. 2015. Geo-referenced Proximity Detection of Wildlife with WildScope: Design and Characterization. In *Proc. of Int. Conf. on Information Processing in Sensor Networks (IPSN)*.

[43] R. Postoyan, A. Anta, W. P. M. H. Heemels, P. Tabuada, and D. Nešić. 2013. Periodic Event-triggered Control for Nonlinear Systems. In *52nd IEEE conference on decision and control*. IEEE, 7397–7402.

[44] G. Pujolle. 2006. An Autonomic-oriented Architecture for the Internet of Things. *Proc. of the IEEE John Vincent Atanasoff Int. Symp. on Modern Computing (JVA)*.

[45] C. Ramesh, H. Sandberg, and K. H. Johansson. 2013. Design of State-based Schedulers for a Network of Control Loops. *IEEE Trans. Automat. Control* 58, 8 (2013), 1962–1975.

[46] M. Schuß, C. A. Boano, M. Weber, and K. Römer. 2017. A Competition to Push the Dependability of Low-Power Wireless Protocols to the Edge. In *Proc. of the 14th Int. Conf. on Embedded Wireless Systems and Networks (EWSN)*.

[47] F. Sutton et al. 2017. The Design of a Responsive and Energy-Efficient Event-Triggered Wireless Sensing System. In *Proc. of Int. Conf. on Embedded Wireless Systems and Networks (EWSN)*.

[48] Paulo Tabuada. 2007. Event-triggered Real-time Scheduling of Stabilizing Control Tasks. *IEEE Trans. Automat. Control* 52, 9 (2007), 1680–1685.

[49] M. Trobinger, D. Vecchia, D. Lobba, T. Istomin, and G.P. Picco. 2020. One Flood to Route Them All: Ultra-Fast Convergecast of Concurrent Flows over UWB. In *Proc. of ACM Conf. on Embedded Networked Sensor Systems (SenSys)*.

[50] B. van Eekelen, N. Rao, B. A. Khashooei, D. J. Antunes, and W. P. M. H. Heemels. 2016. Experimental Validation of an Event-triggered Policy for Remote Sensing and Control with Performance Guarantees. In *Proc. of Int. Conf. on Event-based Control, Communication, and Signal Processing (EBCCSP)*.

[51] X. Vilajosana, T. Watteyne, T. Chang, M. Vučinić, S. Duquennoy, and P. Thubert. 2020. IETF 6TiSCH: A Tutorial. *IEEE Communications Surveys Tutorials* 22, 1 (2020), 595–615.

[52] X. Wang and M. D. Lemmon. 2008. Event Design in Event-triggered Feedback Control Systems. In *Proc. of the 47th IEEE Conference on Decision and Control (CDC)*.

[53] Rubicon Water. 2019. FlumeGate – Flow Control and Measurement Gate. https://www.rubiconwater.com/catalogue/flumegate [Accessed 25-11-2019].

[54] Erik Weyer. 2001. System Identification of an Open Water Channel. *Control engineering practice* 9, 12 (2001), 1289–1299.

[55] Tianyu Zhang, Tao Gong, Chuancai Gu, Huayi Ji, Song Han, Qingxu Deng, and Xiaobo Sharon Hu. 2017. Distributed Dynamic Packet Scheduling for Handling Disturbances in Real-Time Wireless Networks. In *2017 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*.

[56] M. Zimmerling, F. Ferrari, L. Mottola, and L. Thiele. 2013. On Modeling Low-Power Wireless Protocols Based on Synchronous Packet Transmissions. In *Proc. of Int. Symp. on Modelling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*.

[57] M. Zimmerling, L. Mottola, P. Kumar, F. Ferrari, and L. Thiele. 2017. Adaptive Real-Time Communication for Wireless Cyber-Physical Systems. *ACM Transaction on Cyber-Physical Systems* 1, 2, Article 8 (2017).

[58] M. Zimmerling, L. Mottola, and S. Santini. 2020. Synchronous Transmissions in Low-Power Wireless: A Survey of Communication Protocols and Network Services. *Comput. Surveys* 53, 6, Article 121 (2020).