Efficient Holistic Control: Self-awareness across Controllers and Wireless Networks

YEHAN MA and CHENYANG LU, Washington University in St. Louis, USA
YEBIN WANG, Mitsubishi Electric Research Laboratories, USA

Industrial automation is embracing wireless sensor-actuator networks (WSANs). Despite the success of WSANs for monitoring applications, feedback control poses significant challenges due to data loss and stringent energy constraints in WSANs. *Holistic control* adopts a cyber-physical system approach to overcome the challenges by orchestrating network reconfiguration and process control at run time. Fundamentally, it leverages self-awareness across control and wireless boundaries to enhance the resiliency of wireless control systems. In this article, we explore efficient holistic control designs to maintain control performance while reducing the communication cost. The contributions of this work are five-fold: (1) We introduce a holistic control architecture that integrates Low-power Wireless Bus (LWB) and two control strategies, *rate adaptation* and *self-triggered control*; (2) We present heuristics-based and optimal rate selection algorithms for rate adaptation; (3) We design novel network adaptation mechanisms to support rate adaptation and self-triggered control in a multi-hop WSAN; (4) We build WCPS-RT, a real-time network-in-the-loop simulator that integrates MATLAB/Simulink and a physical WSAN testbed to evaluate wireless control systems; (5) We empirically explore the tradeoff between communication cost and control performance in holistic control approaches. Our studies show that rate adaptation and self-triggered control offer advantages in control performance and energy efficiency, respectively, in normal operating conditions. The advantage in energy efficiency of self-triggered control, however, may diminish under harsh physical and wireless conditions due to the cost of recovering from data loss and physical disturbances.

CCS Concepts: • **Networks** → **Sensor networks**; • **Computer systems organization** → **Sensor networks**;

Additional Key Words and Phrases: Industrial wireless control, multi-hop mesh network, network reconfiguration, network-in-the-loop simulation, cyber-physical systems, rate adaptation

ACM Reference format:
Yehan Ma, Chenyang Lu, and Yebin Wang. 2020. Efficient Holistic Control: Self-awareness across Controllers and Wireless Networks. *ACM Trans. Cyber-Phys. Syst.* 4, 4, Article 41 (June 2020), 27 pages.
https://doi.org/10.1145/3371500

This work was sponsored by NSF through grants 1320921 (NeTS) and 1646579 (CPS) and by the Fullgraf Foundation.
Authors' addresses: Y. Ma and C. Lu, Department of Computer Science and Engineering, Washington University in St. Louis, MO, 63130; emails: {yehan.ma, lu}@wustl.edu; Y. Wang, Mitsubishi Electric Research Laboratories, Cambridge, MA, 02139; email: yebinwang@ieee.org.

2378-962X/2020/06-ART41 $15.00
https://doi.org/10.1145/3371500

1 INTRODUCTION

Wireless sensor-actuator networks (WSAN) are being adopted in industrial process automation for their advantages in reducing deployment and maintenance cost. While existing WSANs are usually used for monitoring, it remains challenging to support feedback control loops over WSANs, which is referred to as wireless networked control systems (WNCS) [1]. First, compared with traditional networked control systems (NCS) with wired networks, the control performance of WNCS can be compromised by data losses due to dynamic channel conditions in WSANs. This is undesirable, because control performance is closely related not only to the factory's profits, but also machine operator's safety and the environment. Second, a wireless device that requires a power cord is often impractical in industry settings [2, 3]. In practice, an independent and reliable power supply (e.g., battery) is often mandatory. Given the difficulty to replace batteries in harsh industrial environments, to ensure a reliable connection between the controllers and the sensors and actuators despite the long distance, the key to the design of field devices and wireless standards (e.g., WirelessHART) is to maximize the battery life of the devices such that they could be battery powered for 4 to 10 years. Therefore, it is crucial to improve the energy efficiency of WSANs while maintaining control performance in a WNCS. Finally, WNCS must be resilient to both disturbance to the physical plant and interference to the wireless networks. Therefore, a practical and dependable industrial WNCS must meet the following requirements: (1) *control performance*, which brings economic benefits; (2) *energy efficiency*, which reduces maintenance cost; and (3) *resiliency*, which prevents accidents.

Traditionally, the wireless network and the physical process are managed separately in a WNCS at run time. The lack of coordination between network and plant management forces conservative designs that trade energy for control performance. For example, a WNCS may rely on high sampling rates to guarantee control performance under worst-case conditions, even though the same sampling rates may result in excessive communication cost under normal conditions. Conversely, a less conservative design may result in a fragile system vulnerable to physical disturbance and/or wireless interference. In contrast to the traditional approach, the *holistic control* approach aims to enhance the resiliency and efficiency of WNCS by *cojoining* network reconfiguration and process control [4].

In this work, we explore efficient holistic control designs to maintain control performance at low energy cost. We develop holistic control approaches that incorporate two alternative strategies, *rate adaptation (RA)* and *self-triggered control (ST)*. We note that RA introduces adaptation in a traditional *time-driven* control framework, while ST is a representative *event-driven* control approach. Exploring both strategies in holistic control allow us to investigate the design tradeoff involved in holistic control design. Specifically, the contributions of this work are five-fold:

- We introduce a new *holistic control architecture* that integrates multi-hop wireless networks running the Low-power Wireless Bus (LWB) protocol [5] and two alternative control strategies, RA and ST;
- We present two online RA approaches based on heuristics and optimal rate selections, respectively, and establish stability of the resultant closed-loop control system;
- We design *robust network adaptation* mechanisms to support RA and ST, respectively, in multi-hop LWB networks;
- We build *WCPS-RT*, a real-time network-in-the-loop simulator that integrates MATLAB/Simulink and a physical WSAN testbed to evaluate wireless control systems;
- We empirically explore the tradeoff between communication cost and control performance under alternative holistic control approaches.

Our case studies show that RA and ST offer advantages in control performance and energy efficiency, respectively, under normal operating conditions. The advantage in energy efficiency of ST, however, may diminish under harsh physical and wireless conditions due to the cost of recovering from data loss and physical disturbance.

The rest of the article is organized as follows: Section 2 reviews related works on WNCS designs. Section 3 introduces the system architecture of holistic control systems. Sections 4 and 5 detail the control and network designs of RA and ST. Section 6 presents the real-time wireless cyber-physical simulator (WCPS-RT), and Section 7 analyzes the experimental results.

2 RELATED WORK

WNCS are composed of distributed sensors, actuators, and controllers communicating through wireless networks. Due to the benefits of flexibility and low deployment and maintenance cost, WNCS are expanding their applications over industry processes, autonomous warehouses, and smart factories [6]. However, WNCS face serious challenges due to the inherent dynamics in wireless conditions and limited energy resources in wireless networks [7]. The problem of resilient and efficient wireless control has been investigated in the fields of control theory, wireless networks, and more recently network-control co-designs [1].

In control theory, state observers [8] (e.g., extended Kalman filter) have been introduced to handle packet loss and communication latency in WNCS. To reduce communication cost, aperiodic control has been proposed as an alternative to periodic control. Examples include event-triggered control [9, 10] and self-triggered control [11]. However, existing implementation of aperiodic control was based on a single-hop wireless network [12] instead of the multi-hop WSANs that are widely adopted in process industries due to their flexibility and scalability in industrial environments. Supporting aperiodic control on a multi-hop WSAN is challenging, because industrial WSAN standards usually employ TDMA protocols for predictable communication. The aperiodic communication triggered by aperiodic control is incompatible with the periodic, time-driven nature of communication in industrial multi-hop WSANs.

In wireless networks, given the latency, packet delivery, and energy consumption bounds by control designers, network designs can achieve optimized energy-efficiency [13], reliability [14], load balancing [15], and real-time performance [16] under various wireless channel conditions and network topologies. Breath [13] is proposed to minimize the energy cost while ensuring a desired packet delivery rate and delay of the WSAN by adjusting routing, MAC, radio power, and sleeping discipline. SchedEX [14] is proposed to minimize delay while providing reliability guarantees by producing the TDMA schedule. QU-RPL [15] achieves load balancing and improves end-to-end reliability based on queue utilization. Blink [16] supports hard real-time communication in multi-hop WSAN at low energy cost. However, few of those protocols are cognizant of control performance directly. Better network performance does not always imply good control performance of the physical plant. Indeed, the internal properties of the physical plant, such as its stability, inherently influence the impact of improvements in network communication on control performance.

Recent effort on network-control co-design aims to jointly optimize the network and control at design time. Previous works on sampling rate optimization [17–21] exploit the freedom of sampling rates to optimize control performance under various network protocols and system settings. For wired control, Li et al. [17] minimize useful information loss under network bandwidth constraints. Our project differs from this work in the objective of optimizing control performance while lowering energy cost of WSAN. Goswami et al. [18] handle both real-time and control performance constraints by modeling ECUs over a FlexRay bus. While the work is based on a wired network, FlexRay shares similarities to LWB used in our system in their TDMA-based scheduling approach. Our project differs from the work in our focus on online rate adaptation, while Goswami et al. tackled the optimization problem of offline optimization. Furthermore, we also address self-triggered control approaches and network adaptation protocols.

For wireless control, Demirel et al. [19] design packet-forwarding policies over an unreliable and energy-constrained WSAN; Saifullah et al. [20] optimize sampling rates under the end-to-end deadline constraints of data flows in a WirelessHART mesh network; Kim et al. [21] focus on control over IEEE 802.11 networks. Asymmetric routing [22] enhances control performance and network efficiency by applying different routing strategies to sensing and actuation data flows, since sensing and actuation can have different levels of robustness to packet loss.

However, all these efforts focus on offline designs instead of online adaptation, which limits the resiliency and efficiency of WNCS operating in dynamic conditions (e.g., under network interference and physical disturbance; under transient state or steady state). Online rate optimization has been investigated in References [23–25] for different objectives. Specifically, Bai et al. [23] minimize tracking error under the constraints of network capacity and delay requirement; Bao et al. [24] optimize the control performance over noisy channels under total bit-rate constraint; Colandairaj et al. [25] adapt sampling rates using a static sampling policy based on control performance and network performance in an IEEE 802.11b network.

This article considers the energy cost of WSANs and the design and implementation of the network reconfiguration mechanisms for RA over a multi-hop WSAN under the LWB protocol, which are not addressed by these previous works. In prior work [4], we proposed the concept of holistic control that co-joins network management and physical control at run time. As a simple proof of concept, we presented a holistic control example that adjusts the numbers of transmissions (Txs) based on physical states. In this article, we generalize the designs of holistic control by incorporating more sophisticated control approaches, namely, RA and ST. The new control approaches require more sophisticated network reconfiguration mechanisms that are both efficient and robust. Furthermore, the alternative control approaches (time-driven vs. event-driven) allow us to explore the design tradeoff involved in holistic control in multi-hop WSANs.

It is challenging to conduct experiments on industrial control systems in the field, especially under cyber and physical disturbances. Lab-scale equipment, however, is usually too small for realistic cyber-physical experiments, particularly for multi-hop wireless networks. Therefore, simulation tools are of vital importance to WNCS. Truetime [26] is a MATLAB/Simulink-based tool that enables simulations of CPU scheduling, communication, and control algorithms. NCSWT [27] integrates MATLAB/Simulink and NS-2 for modeling and simulation of NCSs. Neither of the native wireless simulations of Truetime nor the NS-2 simulator can accurately model the probabilistic and irregular packet receptions of WSANs [28, 29]. WCPS [30] integrates MATLAB/Simulink and TOSSIM [31], specifically designed to emulate complex temporal link dynamics of WSANs. However, given the complexity of wireless communication in physical environments, simulators cannot always capture the real-world behavior of WSANs. Network-in-the-loop simulations have recently been developed to address the limitation of wireless simulations by incorporating physical wireless networks [32]. Experiments presented in Reference [12] integrate two double-tank systems with a single-hop wireless network. Baumann et al. [33–35] integrate two real inverted pendulums and a 13-node multi-hop WSAN testbed, achieving sampling rates of tens of milliseconds. However, the physical plants in laboratory settings used in those experiments cannot represent large-scale industrial processes and are limited to the specific lab-scale physical plants. In this work, we design and implement a general network-in-the-loop simulator, which integrates MATLAB/Simulink simulations and a 70-node WSAN testbed.

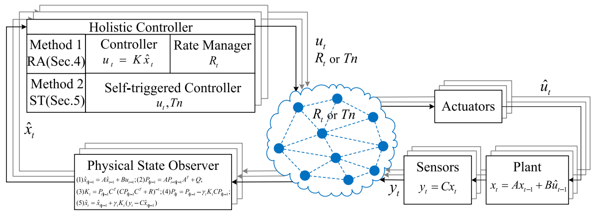


Fig. 1. Holistic WNCS architecture.

3 WIRELESS CONTROL SYSTEM ARCHITECTURE

Figure 1 depicts the holistic wireless control architecture. The holistic controllers (1) control the physical plants by communicating with sensors and actuators through a multi-hop WSAN and (2) reconfigure the WSAN based on control needs at run time. Multiple control loops share the same WSAN, which is the most common deployment in the field [7, 36, 37]. As shown in Figure 1, at time t, a sensor sends its measurements y_t to a remote holistic controller over the multi-hop WSAN. A state observer [8] estimates the state of the plant. Based on the estimated state $\hat{x}_t$, the holistic controller generates both (1) the control commands (u_t) and (2) the network reconfiguration signal (R_t or Tn). Two instances of holistic controller, namely, RA and ST, are introduced. For RA (or ST), the control commands u_t and the updated sampling rate R_t (or next event time Tn) generated by the holistic controller are sent to the WSAN through flooding. For the control commands, the actuator receives u_t and applies $\hat{u}_t$ to the physical plant. If u_t fails to be delivered by the deadline, the actuator reuses the control input received in the last period, $\hat{u}_{t-1}$. For network reconfiguration, every node in the network reconfigures its communication schedule based on R_t or Tn. The details of control and network designs for RA and ST are presented in Section 4 and Section 5, respectively.

3.1 Physical Control System

In this article, control design and analysis are performed for the physical plant, which can be modelled as a linear time-invariant system (LTI) as follows:

$$x_{t+1} = Ax_t + Bu_t, y_t = Cx_t, \tag{1}$$

where t is the time index, $x_t \in \mathbb{R}^n$ is the state vector, $u_t \in \mathbb{R}^m$ is the input vector, $y_t \in \mathbb{R}^p$ is the output vector, $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, and $C \in \mathbb{R}^{p\times n}$. We assume that the pair (A, B) is controllable and that the pair (A, C) is observable. This implies the existence of a linear state feedback controller $u_t = Kx_t$, which renders the closed-loop control system asymptotically stable. Note that the proposed wireless network reconfiguration mechanisms, however, are not limited to LTI systems and are applicable to nonlinear and time-varying systems.

The stability analysis of the resultant control system can be conducted by using the Lyapunov theory. System (1) is stable if there exists a positive definite Lyapunov function [38]

$$V(x_t) = x_t^\top Px_t, \tag{2}$$

such that

$$V(x_{t+1}) - V(x_t) = x_t^\top \Big((A + BK)^\top P(A + BK) - P\Big)x_t = -x_t^\top Qx_t, \tag{3}$$

where P, Q are positive definite matrices. P and Q satisfy the discrete-time Lyapunov equation:

$$(A + BK)^\top P(A + BK) - P = -Q. \tag{4}$$

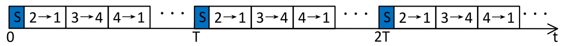


Fig. 2. LWB with static global schedule. ($f_{1,1}$, node2 → node1, $\frac{1}{T}$ Hz; $f_{2,1}$, 3 → 4, $\frac{1}{T}$ Hz; $f_{3,1}$, 4 → 1, $\frac{1}{T}$ Hz.)

3.2 Wireless Sensor-actuator Network

3.2.1 Low-power Wireless Bus (LWB). The WSAN extends the *LWB* [5] protocol to support data communication and network reconfiguration for holistic control. LWB is based on *Glossy* [39], a fast-flooding protocol that exploits the constructive interference among concurrent transmissions of radios compatible with the IEEE 802.15.4 standard. The flooding process is entirely driven by radio events, i.e., a transmission is triggered by completing a packet reception, which drastically speeds up the process and provides microsecond-level WSAN synchronization. Under LWB, nodes take turns to flood their packets in a time-triggered fashion using Glossy flooding according to a single global schedule. A sink node is responsible for disseminating the schedule to all the nodes in the network. Thus, the multi-hop many-to-all communication can be regarded as a single communication resource (shared bus) that runs on a single clock [16].

Adopting LWB as the underlying communication protocol brings significant benefits. Thanks to Glossy flooding, communication in LWB is topology-independent. Besides, LWB is a wireless protocol that provides deterministic end-to-end latency given a global schedule [5], which largely simplifies the analysis of system stability. Additionally, fast Glossy flooding achieves propagation latency within 10 ms over 100 nodes (8 hops, 3 Txs). We can take the advantage to realize fast network reconfiguration by quickly flooding network configurations across the entire network, an important feature, as network reconfiguration is a key element of holistic control.

3.2.2 Implicit Scheduling of Multi-rate LWB. Unlike prior work [16], which uses a centralized scheduler node to operate scheduling algorithms, we tailored *LWB* for implicit scheduling. All nodes schedule themselves based on information from holistic controllers, such as flooding rates or next event timers of each control loop. We define a data flow of WSAN as $f_{i,j}$, which transmits data from a source node $s_{i,j}$ to a destination node $d_{i,j}$, where $i \in \{1, 2, \ldots, n\}$ is the control loop index, and $j \in \{1, 2, \ldots, m_i\}$ is the flow index of the control loop i (l_i). Accordingly, n is the number of control loops, and m_i is the number of data flows in l_i. For example, the control loop l_1 has two data flows $f_{1,1}$ and $f_{1,2}$, among which $f_{1,1}$ is a sensing flow transmitting measurements from a sensor node ($s_{1,1}$) to a controller node ($d_{1,1}$), and $f_{1,2}$ is an actuation flow transmitting control command from a controller node ($s_{1,2}$) to an actuator ($d_{1,2}$). A MIMO control loop can have multiple sensing and actuation flows. The update rate of control commands in the control loop l_i is denoted as R_i. The operation period of l_i is $T_i = \frac{1}{R_i}$. We assume the rates of the flows in one control loop are equal.

In implicit scheduling of data flows, each node stores a static global schedule of all data flows, denoted by entries $f_{i,j}[s_{i,j},\ d_{i,j},\ t_{i,j}]$, $t_{i,j}$ is the relative time slot reserved for flow $f_{i,j}$ in LWB period $T = \frac{1}{R}$. LWB operates at the highest rate of all the control loops, $R = \max_{1 \leq i \leq n} R_i$. Figure 2 shows a simple static schedule. We assume there are three control loops and each loop has one flow. All loops have same rate $R_1 = R_2 = R_3 = \frac{1}{T}$. Thus, the rate of LWB is $R = \frac{1}{T}$. Therefore, we get the static schedule entries: $f_{1,1}[2, 1, 1]$, $f_{2,1}[3, 4, 2]$, $f_{3,1}[4, 1, 3]$. In each period T, the synchronization message S is flooded by the sink node in the beginning of every period, followed by three data slots assigned for three flows.

This static schedule is calculated assuming each control loop runs at its highest candidate rate. The static schedule can be calculated offline using any scheduling algorithm, e.g., EDF or RM. In practice, industrial process control systems usually run at sampling rates lower than 1 Hz [40]. By adopting fast Glossy flooding (flooding a packet over 100 nodes within 10 ms [39]), WSAN

Fig. 3. Implicit scheduling. ($f_{1,1}$, node2 $\rightarrow$ node1, $\frac{1}{T}$ Hz; $f_{2,1}$, 3 $\rightarrow$ 4, $\frac{1}{2T}$ Hz; $f_{3,1}$, 4 $\rightarrow$ 1, $\frac{1}{4T}$ Hz.)

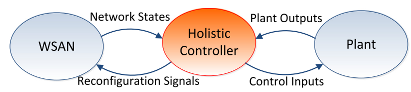


Fig. 4. Holistic management of WNCS.

can guarantee the schedulability of tens of data flows, which suggests the feasibility of the static schedule. We refer interested readers to References [16, 41] for network designs with tighter real-time requirements.

To implement multi-rate LWB using implicit scheduling, besides the static global schedule, the only information that all nodes need are the rates of all the control loops R_i. To make the *implicit scheduling* work properly, the potential T_i of all the loops should be set to integral multiples of the shortest period T. Then each node can independently decide whether to flood $f_{i,j}$ or sleep at $t_{i,j}$ within the time interval $[(k-1)T, kT]$, $k = 1, 2, 3, \ldots$, depending on R_i. Figure 3 shows an example of the *implicit scheduling* with the static schedule in Figure 2, where $R_1 = \frac{1}{T}$, $R_2 = \frac{1}{2T}$, and $R_3 = \frac{1}{4T}$. All nodes flood $f_{1,1}$ at the first data slot of every period T, flood $f_{2,1}$ at the second data slot every other period T, and flood $f_{3,1}$ at the third data slot every 4T. They sleep at the rest blank data slots.

In implicit scheduling, since each node stores the static schedule, the network reconfiguration commands can be generated by any source nodes in WSAN distributively, in contrast to *centralized scheduling* in which the whole schedule is sent by the sink in the beginning of each period T. We will present how network reconfiguration signals, such as R_i, are disseminated in Sections 4.2 and 5.2.

3.3 Holistic Management

As shown in Figure 4, we develop a holistic control architecture that bridges the gap between the plant control and WSAN management. Based on the current status of physical plants and WSAN, the holistic controller generates two kinds of commands at the same time, one for dynamically adjusting the network configuration and the other for operating the physical plants. In the following two sections, we focus on two specific efficient holistic control designs: rate adaptation and self-triggered control over a multi-hop mesh network.

4 RATE ADAPTATION

The data flow rates of a WSAN have direct impacts on control performance and energy cost. The higher the rates, the better the control performance, but the higher the energy cost [19]. In this section, to ensure the control performance while reducing the network energy cost, we adjust the rates of the WSAN based on control performance during run time. We introduce the holistic controller design and the network design of rate adaptation (RA).

4.1 Control Design

We propose two online RA strategies. First is a heuristic-based RA, which selects rate based on physical states and customized thresholds. Second is an optimal RA by minimizing a certain performance metric characterizing the control performance and communication cost. Finally, the stability of the resultant closed-loop control system is established. Please note that, in this article, the sampling rate of multiple loops is adapted in a distributed way. That is, each loop has its own holis-

ALGORITHM 1: Heuristic rate adaptation algorithm for loop i

Input: x_t, t, τ, $t_0 = t$, λ, candidate rates (ascending): $\{R_{i,1}, R_{i,2}, \ldots, R_{i,s}\}$, current $R_i = R_{i,j}$, A_i, B_i, K, P, Q
Output: updated R_i
Calculate $V(x_t)$ as defined in (2), and V_{Dth}, V_{Ith};
if $V(x_t)$ *remains below* V_{Dth} *for a time interval of* τ, **and** $R_i > R_{i,1}$ **then**
  $R_i \leftarrow R_{i,j-1}$;`//`R_i `decreases`
else if $V(x_t) > V_{Ith}$ **and** $R_i < R_{i,s}$ **then**
  if *last rate adaptation is a decrease* **then**
    $t_0 \leftarrow t$; $R_i \leftarrow R_{i,j+1}$;`//`R_i `increases`
  if *last rate adaptation is an increase* **and** $V(x_t) > (1 - \frac{\beta}{\alpha_2})^{t-t_0} V(x_{t_0})$ **then**
    $t_0 \leftarrow t$; $R_i \leftarrow R_{i,j+1}$;`//succeeding` R_i `increases`
else
  R_i remains constant

tic controller. Each loop determines its own sampling rate (R_i) individually. The rate is calculated and potentially adapted every sampling period. We discuss the RA strategies for loop i.

4.1.1 Heuristic Rate Adaptation. We employ a similar adaptation algorithm proposed in Reference [4] (Algorithm 1). The value of the Lyapunov function $V(x_t)$ in Equation (2), the metric of the control performance, provides the bounds of the state error. Given Equation (2),

$$\alpha_1 ||x_t||^2 \leq V(x_t) \leq \alpha_2 ||x_t||^2, \tag{5}$$

where α_1 and α_2 are the smallest and largest eigenvalues of P, respectively. The value of $V(x_t)$ is used to update the rate. Given a customized state error bound, denoted as $se = ||x_{se}||^2$, we set the rate increasing threshold $V_{Ith} = \alpha_1 ||x_{se}||^2$. Based on Equation (5), we have $||x_t||^2 \leq ||x_{se}||^2$, if $V(x_t) \leq V_{Ith}$. Furthermore, we adopt a more stringent decreasing threshold V_{Dth} to indicate that the system performs well, $V_{Dth} = \lambda \alpha_1 ||x_{se}||^2, \lambda \in (0, 1)$. If $V(x_t)$ remains below V_{Dth} for a customized time interval τ, the control system is regarded in good condition. Given Equation (3),

$$V(x_{t+1}) - V(x_t) \leq -\beta ||x_t||^2, \tag{6}$$

where β is the smallest eigenvalue of Q. Given Equations (5) and (6), we can get the upper bound of the ideal Lyapunov function, described by Equation (7). We set this bound as the trigger of succeeding rate increases:

$$V(x_{t+j}) \leq (1 - \beta/\alpha_2)^j V(x_t). \tag{7}$$

The heuristic RA algorithm of a holistic controller is presented in Algorithm 1. Its complexity is $O(1)$.

4.1.2 Optimal Rate Adaptation. A disadvantage of the aforementioned heuristics-based RA is that it requires hand-tuning, which can be challenging for complex control systems. Furthermore, it does not offer a systematic way to balance control system performance and communication cost, the two important and conflicting concerns in wireless control systems. Henceforth, we formulate rate selection as an optimization problem. The objective of the optimization problem is to minimize a cost function that incorporates both control performance and communication cost.

As described in Section 3.2.2, each candidate period of a feedback control loop is an integral multiple of the smallest sampling period T. Let $T_s = n_s T$ be the least common multiple of all candidate periods of a feedback control loop. To compare the control performance resulting from different rates, we rewrite all possible systems with different sampling rates in the slowest time frame T_s in a process referred to as *lifting* [42].

By lifting the system in the slowest time frame T_s, the system is given by

$$x_{t+n_sT} = A_{d1}^{n_s} x_t + \left[A_{d1}^{n_s-1} B_{d1} \ \dots \ B_{d1}\right] \begin{bmatrix} u_{t,1} \\ \vdots \\ u_{t+(n_s-1)T,1} \end{bmatrix}, \tag{8}$$

where $u_{t+iT,1}$ is the control input during time interval $\left[t+iT, t+(i+1)T\right)$, and $A_{d1} = e^{A_c T}$, $B_{d1} = \int_0^T e^{A_c(T-\tau)} B_c d\tau$, where A_c and B_c are the system matrices of the original continuous system dynamics $\dot{x} = A_c x + B_c u$. For the lowest sampling rate $1/T_s$, the corresponding system does not need lifting and has the dynamics

$$x_{t+n_sT} = A_{dn_s} x_t + B_{dn_s} u_{t,n_s}, \tag{9}$$

where u_{t,n_s} is defined over $[t, t+n_sT)$, and $A_{dn_s} = e^{n_s AT}, \quad B_{dn_s} = \int_0^{n_sT} e^{A(n_sT-\tau)} B d\tau$. To make a fair evaluation for systems resultant from different rates, we rewrite the slowest system (9) as

$$x_{t+n_sT} = A_{dn_s} x_t + \left[A_{d1}^{n_s-1} B_{d1} \quad \dots \quad B_{d1}\right] \begin{bmatrix} u_{t,n_s} \\ \vdots \\ u_{t+(n_s-1)T,n_s} \end{bmatrix} = A_{d1}^{n_s} x_t + \sum_{i=0}^{n_s-1} A_{d1}^{i} B_{d1} u_{t,n_s} \quad , \tag{10}$$

where $u_{t,n_s} = u_{t+iT,n_s}, i \in \{0, \dots, n_s-1\}$, and $A_{dn_s} = A_{d1}^{n_s}$. Finally, we can rewrite the system dynamics of loop i, depending on the rate R_i, as follows:

$$x_{t+n_sT} = \begin{cases} A_{d1}^{n_s} x_t + \sum_{i=1}^{n_s-1} A_{d1}^{i} B_{d1} u_{t+iT,1}, \ \text{if } R_i = 1/T \\ \vdots \\ A_{d1}^{n_s} x_t + \sum_{i=1}^{n_s-1} A_{d1}^{i} B_{d1} u_{t+iT,k}, \ \text{if } R_i = 1/(kT) \\ \vdots \\ A_{d1}^{n_s} x_t + \sum_{i=1}^{n_s-1} A_{d1}^{i} B_{d1} u_{t,n_s}, \ \text{if } R_i = 1/T_s \end{cases} , \tag{11}$$

Based on Equation (11), the states and inputs of systems with all candidate rates are lifted to the lowest rate. We are now ready to formulate rate selection as an optimization problem. Let us evaluate the cost function over a horizon of N sample periods corresponding to the lowest sample rate, i.e., the horizon for performance evaluation lasts NT_s seconds. Since each loop can select its rate individually, we can formulate n independent optimization problems, where n is the number of feedback control loops. Coordination between different control loops is part of our future work. The optimization problem for loop i has decision variables of an N-dimensional vector $\mathbf{R}_i(k)$, where kth element $\mathbf{R}_i(k)$ represents sample rate during time interval $\left[t+(k-1)T_s, t+kT_s\right)$. Finally, cost function is defined as a weighted combination of control performance and communication cost:

$$\mathcal{J}(x_t, \mathbf{R}_i) = \sum_{j=0}^{N-1} \left\{ x_{t+jT_s}\left(\mathbf{R}_i(j)\right)^\top W_Q x_{t+j}\left(\mathbf{R}_i(j)\right) + \epsilon_1 u_{t+j}\left(\mathbf{R}_i(j)\right)^\top W_R u_{t+j}\left(\mathbf{R}_i(j)\right) + \epsilon_2 \mathbf{R}_i(j) \right\}, \tag{12}$$

where x_{t+j} is predicted based on Equation (11) given x_t and control law, $x_{t+j}\left(\mathbf{R}_i(j)\right)^\top W_Q x_{t+j}\left(\mathbf{R}_i(j)\right) + \epsilon_1 u_{t+j}\left(\mathbf{R}_i(j)\right)^\top W_R u_{t+j}\left(\mathbf{R}_i(j)\right)$ represents control performance including state cost and control cost, W_Q, W_R set relative weights of state deviation and control effort, x_{t+j} denotes x_{t+jT_s}, and

$$u_{t+j} = \begin{cases} [u_{t+jT_s,1}, \dots, u_{t+jT_s+(n_s-1)T,1}]^\top, \ \text{if } \mathbf{R}_i(j) = 1/T \\ \vdots \\ [u_{t+jT_s,n_s}, \dots, u_{t+jT_s+(n_s-1)T,n_s}]^\top, \ \text{if } \mathbf{R}_i(j) = 1/T_s. \end{cases}$$

In Equation (12), the communication cost is linearly proportional to sampling rate $\mathbf{R}_i(j)$. Constant ϵ_1 is to weight state error versus control cost, and ϵ_2 is to weight control performance versus communication cost. When ϵ_2 approaches 0, which means that network energy cost is ignored, the WNCS is prone to stay at the fastest sampling rate to achieve better control performance. We define this scenario as *cheap network* in analogy with *cheap control*, which is the case $W_R = 0$ [43] when control performance is evaluated. As a result, the optimization problem of loop i can be written as follows:

$$\underset{\mathbf{R}_i}{\text{minimize}} \quad \mathcal{J}(x_t, \mathbf{R}_i) \tag{13a}$$

$$\text{subject to} \quad \mathbf{R}_i = [\mathbf{R}_i(1), \ldots, \mathbf{R}_i(N)], \text{ with } \mathbf{R}_i(k) \in \{R_{i,1}, \ldots, R_{i,s}\} \tag{13b}$$

$$x_{t+j} = \begin{cases} A_{d1}^{n_s} x_{t+j-1} + \sum_{i=1}^{n_s-1} A_{d1}^{i} B_{d1} u_{t+(j-1)T_s+iT,1}, & \text{if } \mathbf{R}_i(j) = 1/T \\ \vdots & \\ A_{d1}^{n_s} x_{t+j-1} + \sum_{i=1}^{n_s-1} A_{d1}^{i} B_{d1} u_{t+(j-1)T_s,n_s}, & \text{if } \mathbf{R}_i(j) = 1/T_s \end{cases} \tag{13c}$$

$$u_t = Kx_t. \tag{13d}$$

The optimization problem (13) has N integer decision variables. Since the decision variables $\mathbf{R}_i(j)$ belong to a finite set of candidate rates, the optimal rate adaptation problem is an integer programming problem, which could be computationally expensive to solve at every sampling period. To reduce the computational complexity, we simplify Equation (13) by assuming that the control system stays at the same rate over the horizon, i.e., $\mathbf{R}_i(1) = \cdots = \mathbf{R}_i(N) = R_i$. Accordingly, the cost function is given by

$$\mathcal{J}(x_t, R_i) = \sum_{j=0}^{N-1} \{x_{t+jT_s}(R_i)^\top W_Q x_{t+j}(R_i) + \epsilon_1 u_{t+j}(R_i)^\top W_R u_{t+j}(R_i) + \epsilon_2 R_i\}. \tag{14}$$

The simplified optimization problem takes the following formulation:

$$\underset{R_i}{\text{minimize}} \quad \mathcal{J}(x_t, R_i) \tag{15a}$$

$$\text{subject to} \quad R_i \in \{R_{i,1}, \ldots, R_{i,s}\}. \tag{15b}$$

$$x_{t+j} = \begin{cases} A_{d1}^{n_s} x_{t+j-1} + \sum_{i=1}^{n_s-1} A_{d1}^{i} B_{d1} u_{t+(j-1)T_s+iT,1}, & \text{if } R_i = 1/T \\ \vdots & \\ A_{d1}^{n_s} x_{t+j-1} + \sum_{i=1}^{n_s-1} A_{d1}^{i} B_{d1} u_{t+(j-1)T_s,n_s}, & \text{if } R_i = 1/T_s \end{cases} \tag{15c}$$

$$u_t = Kx_t. \tag{15d}$$

Although the simplified optimization problem (15) is an integer programming problem, for each loop i it has only one scalar decision variable R_i (instead of N in Equation (13)). Furthermore, the number of candidate rates is usually small in practice, which significantly reduces the computation complexity. We solve the optimization problem by brute force search. Note that the system matrices of rate lifting can be calculated offline. Given a horizon of N, M candidate rates, and $n_s = \frac{T_s}{T}$, the computation complexity is $O(MNn_s)$. We also evaluate the computation cost in MATLAB/Simulink on a 2.5 GHz Intel Core i7 processor. The settings of the experiments are the same as in Section 7.2.2 ($n_s = 2^{M-1}$). Figure 5 shows the execution time of solving (15) for 2K times. As shown in Figure 5(a), with candidate rates $M = 3$, the median and worst-case execution time when horizon $N < 25$ is below 1 ms and 2.1 ms, respectively. As shown in Figure 5(b), with $N = 10$, the median and worst-case execution time when $M \leq 6$ is below 4 ms and 11 ms, respectively. The execution time is negligible compared with the 1 s sampling period. These results show that the problem (15) is online solvable.

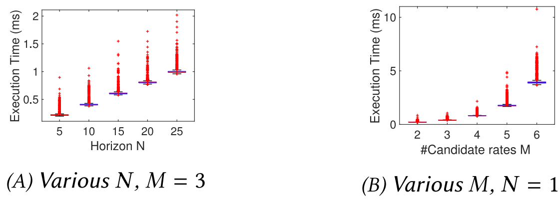


Fig. 5. Execution time of solving Equation (15) with various horizon N and number of candidate rates M.

Remark 4.1. Since we target industrial process control systems with sampling rates lower than 1 Hz [40], we tailor the rate selection for our WSAN design with an assumption of the existence of "worst-case-guaranteed" schedule in Section 3.2. Hence, there is no network resource/schedulability constraint, and the rate selections of multiple loops can be done individually.

For systems with schedulability constraints, we can provide schedulability guarantee by generalizing the optimal RA problems (13) and (15) to incorporate schedulability constraints. We replace the objective function in Reference [20] Equation (8) by $\sum_{i=1}^{n} \mathcal{J}(x_t, \mathbf{R}_i)$ and adding system dynamic constraints (11) of all loops. Since we apply LWB, as studied in Reference [16], the real-time scheduling constraints can be simplified from multi-processor task scheduling in Reference [20] to uni-processor case. However, given that the configuration space of the corresponding centralized optimization problem is much larger than Equations (13) and (15), and the introduction of schedulability constraints, the resultant optimization problem can be computationally expensive to solve online. In addition, this approach requires centralized management of the control loops. Extending our work to consider schedulability constraint is part of future work.

4.1.3 *Stability Analysis.* Deploying the aforementioned RA algorithms renders the closed-loop control system being a switched system, whereas the switch is governed by the RA algorithm. Since it is difficult, if not impossible, to formulate the analytic formula of the switching sequence, we borrow the stability result for switched systems with arbitrary switching. Stability analysis tools for switched systems can be found in Reference [44] and references therein. This work performs stability analysis and control design based on a well-received result: If there exists a common Lyapunov function for all subsystems, then the stability of the switched system is guaranteed under arbitrary switching. It is revealed that the construction of such a common Lyapunov function among all candidate rates can be formulated as a Linear Matrix Inequality (LMI) problem: solves for P satisfying

$$\Big(A(R_i) + B(R_i)K\Big)^\top P\Big(A(R_i) + B(R_i)K\Big) - P < 0, \quad \forall R_i \in \{R_{i,1}, \ldots, R_{i,s}\}, \tag{16}$$

where $A(R_i)$ and $B(R_i)$ are discretized system matrices of loop i corresponding to the sample rate R_i. If there is a feasible solution for the LMI problem (16), then $V(x_t) = {x_t}^\top P x_t$ is the common Lyapunov function of all candidate rates, and the stability is established.

The aforementioned stability analysis, in a deterministic setting, provides a strategy to search for a common Lyapunov function $V(x_t)$. As described in Section 3.2.1, the latency bound of LWB is deterministic [5, 39]. In our test cases, the latency is shorter than one sampling period. Stability analysis under network latency of below one sampling period is well studied. We refer interested readers to References [33, 45]. The stability analysis can be generalized to take indeterministic network latency and packet loss into account, which consequently leads to stochastic stability. Stability analysis under different network scenarios has been intensively studied in control community and is not the focus of this article. We refer interested readers to stability analysis addressing network latency [45, 46] and packet loss with different distribution patterns [47, 48]. Despite the

Table 1. Schedule Examples for Loop 1 with Candidate Rates of $R_{1,1} = \frac{1}{T}$ Hz, $R_{1,2} = \frac{1}{2T}$ Hz, $R_{1,3} = \frac{1}{4T}$ Hz

Rate	$t_{1,1}$	$t_{1,1}+T$	$t_{1,1}+2T$	$t_{1,1}+3T$	$t_{1,1}+4T$	$t_{1,1}+5T$	$t_{1,1}+6T$	$t_{1,1}+7T$	$t_{1,1}+8T$	$t_{1,1}+9T$
$R_{1,1} = \frac{1}{T}$ Hz										
$R_{1,2} = \frac{1}{2T}$ Hz										
$R_{1,3} = \frac{1}{4T}$ Hz										

simplifications, our stability analysis provides practical guidance towards balancing the closed-loop control performance and network rate in real-world scenarios involving network latency and packet loss, as shown in case studies under network and physical interference in Sections 7.4–7.6.

Remark 4.2. The existence of a single common Lyapunov function (16) for all candidate rates is a conservative but easy-to-check stability condition. The relaxation of conservativeness has been intensively studied in control community and leads to numerous results [49–51]. For example, Reference [49] proposed to replace the single common Lyapunov function with a switched Lyapunov function and established a sufficient condition of stability as stated in the following theorem:

THEOREM 4.3 ([49], THEOREM 4). *If there exist symmetric matrices $S(R_i)$, matrices $G(R_i)$ and $U(R_i)$ such that $\forall (R_i, R_j)$*

$$\begin{bmatrix} G(R_i) + G(R_i)^\top - S(R_i) & \left(A(R_i)G(R_i) + B(R_i)U(R_i)\right)^\top \\ A(R_i)G(R_i) + B(R_i)U(R_i) & S(R_j) \end{bmatrix} > 0, \tag{17}$$

then state feedback control gain $K(R_i) = U(R_i)G(R_i), \forall R_i \in \{R_{i,1}, \ldots, R_{i,s}\}$ stabilizes the system.

The results in Reference [49] show the tradeoff between a single Lyapunov function for simplicity and a switched Lyapunov function that is less conservative but numerically hard to check.

4.2 Network Reconfiguration

In this section, we present a run-time RA protocol for a mesh WSAN. Packet loss has non-negligible impacts on WNCS, especially in network reconfiguration. We also discuss its packet loss recovery.

4.2.1 Candidate Rates Selection. Section 4.1 considers how to adjust the rate of each loop. The candidate rates are also important design factors. To ensure that the rate transient processes work properly, the potential rates of each loop need to be designed intentionally. First, according to Section 3.2.2, when the offline scheduler schedules data flow $f_{i,j}$, it reserves time slots for fastest rate R. Second, the candidate periods of all the loops should be integral multiples of the shortest period $T = \frac{1}{R}$. Third, to ensure that the RA works properly with packet loss recovery, which will be discussed later in Section 4.2.3, the candidate rates of each loop should be harmonic, e.g., $(\frac{1}{T}, \frac{1}{2T}, \frac{1}{4T})$ or $(\frac{1}{T}, \frac{1}{3T}, \frac{1}{9T})$. Schedule examples for $(\frac{1}{T}, \frac{1}{2T}, \frac{1}{4T})$ are in Table 1. A filled unit indicates that a packet is sent in that time slot. According to Section 4.1.3, to guarantee stability, a common Lyapunov function should exist by solving the LMI problem formulated by subsystems induced by all candidate rates.

4.2.2 Network Reconfiguration Based on Piggyback. The holistic controller of l_i adopts a *piggyback* mechanism to disseminate a newly computed R_i for data flow $f_{i,j}$. The holistic controller of l_i piggybacks R_i with the actuation command. The data field of the actuation packet is [l_i, R_i, Data$_i$]. Because of the flooding nature of LWB, all nodes in the network can receive this update. Once a node receives an updated R_i, it will calculate a new schedule based on R_i, as described in Section 3.2.2.

The distributed network reconfiguration based on piggyback has several benefits over the conventional centralized network reconfiguration. First, this piggyback mechanism helps reduce energy cost by utilizing existing actuation data flows, saving the time and energy needed to calculate

Table 2. Packet Loss Recovery for Rate Adaptation of Flow f_{11}

	$t_{1,1}$	$t_{1,1}+T$	$t_{1,1}+2T$	$t_{1,1}+3T$	$t_{1,1}+4T$	$t_{1,1}+5T$
Updated rate R_i	$R_{1,1}$	$R_{1,2}$	$R_{1,2}$		$R_{1,2}$	
Node 2*	$R_{1,1}(1)$	$R_{1,1}(1) \rightarrow R_{1,2}$	$R_{1,2}(1)$	$R_{1,2}(/)$	$R_{1,2}(1)$	$R_{1,2}(/)$
Node 3	$R_{1,1}(1)$	$R_{1,1}(1) \rightarrow R_{1,2}$	$R_{1,2}(1)$	$R_{1,2}(/)$	$R_{1,2}(1)$	$R_{1,2}(/)$
Node 4	$R_{1,1}(1)$	$R_{1,1}(0)$	$R_{1,1}(1) \rightarrow R_{1,2}$	$R_{1,2}(/)$	$R_{1,2}(1)$	$R_{1,2}(/)$

and deliver the whole schedule in every period. Second, the network reconfiguration commands can be flooded by any source nodes in WSAN distributively, in contrast to *centralized scheduling*, in which the whole schedule should be sent by the sink. In addition, implicit and distributed scheduling using piggyback is more reliable than a centralized scheduler. Packet loss in implicit scheduling affects only one loop, but the packet loss of centralized scheduling can affect all data flows.

4.2.3 Packet Loss Recovery. If a node loses the packet with the updated rate of l_i, it will use the current R_i until another packet of l_i is received. Therefore, it is possible that, at the same time, different nodes along the route of a flow are using different rates. Nevertheless, it is still possible for nodes to eventually receive the update. The transmissions of three nodes in Table 2 represent an example of packet loss recovery for flow $f_{1,1}$ (source $s_{1,1}$ is node 2) when a holistic controller updates the rate from $R_{1,1}$ to $R_{1,2}$ at the second period ($t_{1,1}+T$). $\{1, 0, /\}$ in brackets following $R_{1,j}$ indicate that the node receives a packet, loses a packet, and remains sleeping, separately, corresponding to schedule of $R_{1,j}$ as shown in Table 1. The update rate is received by nodes 2 and 3, but fails to arrive at node 4 within the second period ($t_{1,1}+T$) due to packet loss. Hence, the rates of nodes 2 and 3 switch to $R_{1,2}$, while node 4 continues to use $R_{1,1}$. Although node 4 uses different rate, it is still possible for it to receive update rate in the third period ($t_{1,1}+2T$), since relative slot $t_{1,1}$ in the third period is the common slot shared by $R_{1,1}$ and $R_{1,2}$. If all candidate rates are harmonic, i.e., share as many common slots as possible, the node will recover faster from packet loss.

5 SELF-TRIGGERED CONTROL

Self-triggered control (ST) [11], an aperiodic event-driven control design, improves the efficiency of the network. The first single-hop wireless network protocol for aperiodic control is presented in Reference [12]. However, due to the lack of network protocol, aperiodic control designs have not been adopted in multi-hop mesh networks. In this section, we, respectively, introduce control design and network design of ST.

5.1 Control Design

In event-triggered control, trigger condition is checked in every sampling period. The time of actuation event cannot be known in advance, which requires the network to reserve resource for unknown events. The ST relaxes this requirement by predicting the future events based on system models. Intuitively, ST triggers sensing and actuation events only when certain control performance is predicted to be lost. The self-triggered strategy we present in this article is motivated by Reference [12]. Since a decreasing Lyapunov function $V(x_t) = x_t^\top P x_t$ is the certificate of stability (P is achieved in Section 4.1.3), the desired control performance is defined by a decreasing function $S(x_t)$, upper bounding the evolution of Lyapunov function $V(x_t)$: $V(x_t) \le S(x_t)$. Provided that $V(x_t) \le S(x_t)$ holds and $S(x_t)$ is decaying over time, the closed-loop system is stabilized [11, 12]. The predicted time of the next sensing and actuation events is $t_k = \min\{t > t_{k-1} | V(x_t) - S(x_t) \ge$

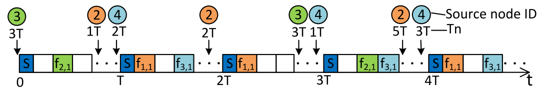


Fig. 6. Self-triggered transmission based on LWB ($f_{1,1}$, node2 → node1; $f_{2,1}$, 3 → 4; $f_{3,1}$, 4 → 1).

0}. Here, we adopt a feasible decreasing $S(x_t)$, as follows:

$$S(x_t) = V(x_{t_{k-1}})e^{-\gamma V(x_{t_{k-1}})^{\delta}(t-t_{k-1})}. \tag{18}$$

We induce the term $\gamma V(x_{t_{k-1}})^{\delta}$, $\gamma, \delta > 0$, which makes the decreasing rate of $S(x_t)$ adapt to the value of the Lyapunov function (state error). That is, when $V(x_{t_{k-1}})$ is large, which indicates severe state error, the $S(x_t)$ decreases faster. Therefore, the sensing and actuation events are more likely to be triggered. However, when $V(x_{t_{k-1}})$ is small, which indicates the current states are close to equilibrium point, the $S(x_t)$ decreases slower. The sensing and actuation events are unnecessary and less likely to be triggered. Please note that, unlike event-triggered control, the trigger condition of which is checked in every sampling period, self-triggered control checks the trigger condition based on predictions based on system model, which makes it less resilient to disturbance. To provide robustness guarantees of the self-triggered control, an upper bound of the inter-transmission interval should be customized based on References [12, 52, 53].

5.2 Network Protocol for Self-triggered Control

5.2.1 Self-triggered Transmissions. Due to the predictive nature of ST, the network knows *a priori* when the event will be triggered by the holistic controllers. Therefore, nodes know the next time when they should wake up and flood data. Within the inter-transmission interval, the nodes sleep. Thus, the energy costs of nodes can be reduced compared with periodic control at the highest rate.

Similar to the network protocol of RA, the holistic controller uses the piggyback mechanism to disseminate a newly computed time of next transmission Tn_i for all data flows of l_i. Again, Tn_i should be integral multiples of T. The holistic controller piggybacks Tn_i with the actuation command. Therefore, the data field of the actuation packet is $[l_i, Tn_i, \text{Data}_i]$. Because of flooding, all nodes in the network can receive this update. In a node, each data flow has an event timer. Once a node receives a Tn_i, it will set the value of $Timer_{i,j}$ to Tn_i and start counting down from the next period. If the $Timer_{i,j}$ expires, the node will wake up and flood in the pre-assigned relative slots $t_{i,j}$ within T. Figure 6 shows an example of self-triggered transmissions based on LWB. At the first period, $f_{2,1}$ is flooded, and node 3, which is the *source* of $f_{2,1}$, receives and floods $Tn_2 = 3T$ at slot that is assigned for $f_{2,1}$. Therefore, the next $f_{2,1}$ is reserved and transmitted $3T$ later at the fourth period. At the second period, $f_{1,1}$ and $f_{3,1}$ are transmitted, and $Tn_1 = T$, $Tn_3 = 2T$, respectively. Therefore, the next $f_{1,1}$ is reserved and transmitted at the third period and $f_{3,1}$ at the fourth period.

5.2.2 Why Not Event-triggered Transmissions. We adopt ST instead of event-triggered control. In event-triggered control, trigger conditions are checked every sampling period. Source node is aware of whether the event is triggered in current period, and it does not flood if the trigger condition is not violated. However, other nodes in mesh WSAN do not know the trigger time in advance; they still wake up and keep listening in case certain events are triggered in current period. Therefore, event-trigger control systems over a multi-hop mesh network cannot reduce duty cycle of the network. As shown in Figure 7, at the first period, the source node of $f_{2,1}$, node 3, notices the event is triggered. It floods $f_{2,1}$ in the second relative time slot. Since all other nodes are listening, they receive and forward $f_{2,1}$. Different from ST, all nodes keep awake in the first and third relative time slots in case the trigger conditions of $f_{1,1}$ and $f_{3,1}$ are violated.

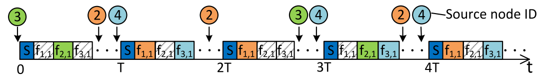


Fig. 7. Event-triggered transmission based on LWB.

Table 3. Impact of Packet Loss on Self-triggered Control of Flow $f_{1,1}$

	$t_{1,1}$	$t_{1,1}+T$	$t_{1,1}+2T$	$t_{1,1}+3T$	$t_{1,1}+4T$	$t_{1,1}+5T$
Updated inter-transmission time: Tn_1	T	$2T$		$2T$		$2T$
Node 2*	$2T(1)\rightarrow T$	$T(1)\rightarrow 2T$	$2T(/)$	$2T(1)$	$2T(/)$	$2T(1)$
Node 3	$2T(1)\rightarrow T$	$T(1)\rightarrow 2T$	$2T(/)$	$2T(1)$	$2T(/)$	$2T(1)$
Node 4	$2T(0)$	$2T(/)$	$2T(0)$	$2T(/)$	$2T(0)$	$2T(/)$

5.2.3 Packet Loss Recovery for ST. If all nodes receive Tn_i and are synchronized well, they wake up and flood $f_{i,j}$ at the same time. However, unlike rate adaptation based on *LWB*, which can self-recover from packet loss, self-triggered transmissions based on LWB are less resilient to packet loss. If a node fails to receive Tn_i, it is possible that it will not wake up at the right time for the next transmission and will become unsynchronized with other nodes for $f_{i,j}$ forever. Table 3 shows an example of the impact of packet loss on flow $f_{1,1}$ when a holistic controller predicts a series of inter-transmission intervals (Tn_1). $\{1, 0, /\}$ in brackets following Tn_1 indicate that the node receives a packet, loses a packet, and remains sleeping, separately, corresponding to its inter-transmission interval Tn_1. In this example, the update inter-transmission time $Tn_1 = T$ is received by nodes 2 and 3, but fails to arrive at node 4 in first period ($t_{1,1}$) due to packet loss. Hence, the nodes 2 and 3 schedule next transmissions in the second period ($t_{1,1} + T$), while node 4 uses last $Tn_1 = 2T$ and schedules next transmission in the third period ($t_{1,1} + 2T$). In the second period, nodes 2 and 3 receive new $Tn_1 = 2T$, and schedule the next transmissions in the fourth period ($t_{1,1} + 3T$). Node 4 sleeps at this period and loses the updated inter-transmission time again. If the system goes on like this, node 4 becomes unsynchornized with other nodes and loses all packets. Therefore, it is of vital importance to come up with effective and efficient strategies to recover from packet loss. We propose the following packet loss recovery strategy: If a node wakes up but does not receive a packet with Tn_i, it should re-awake at the highest rate R until another packet with Tn_i is received.

6 WCPS REAL-TIME

To experiment with wireless control over real-world WSANs, we develop wireless cyber-physical simulator real-time (WCPS-RT).

6.1 Architecture of WCPS-RT

WCPS-RT integrates *MATLAB/Simulink Desktop Real-time (SLDRT)* [54] and a *three-floor WSAN testbed* [55, 56]. The architecture of WCPS-RT is shown in Figure 8. Note that this figure shows the architecture of one wireless control loop. Several control loops can share the same WSAN.

SLDRT is used to simulate the physical part of the WNCS: physical plants, controllers, state observers, and physical disturbance. In practice, industrial plants usually operate continuously or at very high rates. However, the wireless communication and controller execute at a relatively low rate because of the communication and computation latencies. Therefore, SLDRT modules are operated at different rates in our design.

The *three-floor WSAN testbed* is deployed on the third to fifth floors of Jolley Hall at Washington University in St. Louis, as in Figure 9. It consists of 70 TelosB motes. Each mote is equipped with Chipcon CC2420 radio compliant with the IEEE 802.15.4 standard and a TI MSP430 microcontroller. 40 Raspberry Pis with a backplane network are used for the management of the WSAN [7].

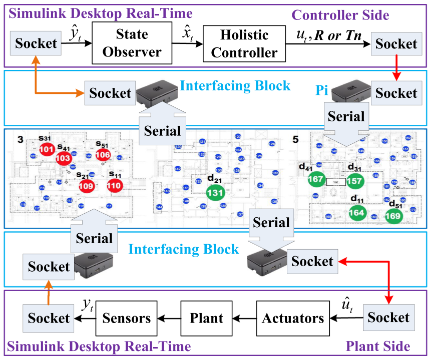


Fig. 8. Architecture of WCPS-RT.

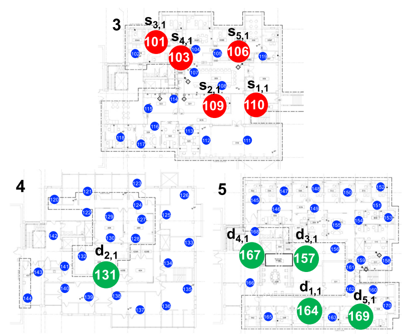


Fig. 9. Three-floor WSAN testbed in Jolley Hall of Washington University in St. Louis.

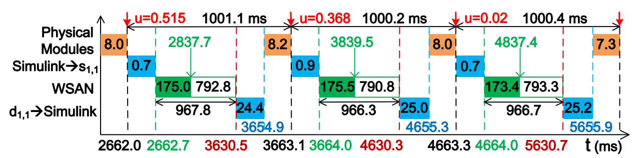


Fig. 10. Timeline of WCPS-RT.

The *interfaces* between SLDRT and WSAN are socket connections between the PCs that run SLDRT and the Pis, and serial connections between the Pis and the end nodes. In this way, the end nodes $s_{i,j}$, $d_{i,j}$ of the sensing and actuation flows $f_{i,j}$ can be any nodes in the testbed.

6.2 Real-time Network-in-the-loop Simulation

Both SLDRT and the three-floor WSAN testbed operate in real-time. To evaluate the real-time performance of the WCPS-RT, we measure the latency caused by each module. In our design, sensing and actuation flows have the same overhead induced by interfaces, since they have the same types of interfaces between physical parts and WSAN, as in Figure 8, and all data flows share the same WSAN with independent interfaces.

We use the latencies of one actuation flow as an illustrative example. First, we adopt the *Precision Time Protocol (PTP)* to synchronize the PC that runs SLDRT and the Pis. *PTP* is a protocol used to synchronize clocks throughout a network. It achieves clock accuracy in the sub-microsecond range [57]. Then, we record the completion timestamps of each module on corresponding machines (1) the physical modules, (2) the actuation flow from Simulink to $s_{1,1}$, (3) the transmissions in WSAN, (4) the actuation flow from $d_{1,1}$ to Simulink. Finally, we draw the timeline of WCPS-RT and analyze the latencies, as shown in Figure 10. We set the sampling period to 1*s*, which is the fastest update time supported by most industrial WSAN products. From the timeline, the total overhead induced by interfaces between Simulink, and the node is less than 26 ms (2.6%). More than 966 ms are reserved for communication over the WSAN in each period, among which around 175 ms are utilized for transmissions in this example. The results validate the real-time performance of WCPS-RT. Please note that 26 ms overhead is acceptable when we use WCPS-RT to simulate industrial processes such as oil refinery and mining, sampling periods of which are usually longer than 1 s [40]. However, it is not acceptable in faster sampling period of tens of milliseconds. We will work on shortening this overhead in the future. We refer interested readers to References [16, 33, 41] for network and WNCS designs with tighter time requirements.

7 EVALUATION

In this section, we describe systematic trials of our wireless control designs using WCPS-RT. On the physical side, to represent an industrial process system, we use up to five 4-state load positioning systems that share the same WSAN. On the WSAN side, we evaluate the proposed network protocols over a 70-node WSAN testbed [55, 56].

Because the state observer provides robust and theoretically sound protection against loss of sensing information [8, 58, 59], the WNCS are more sensitive to packet loss on the actuation side of WSAN [22]. Thus, we focus on comprehensive actuation-network-in-the-loop simulations. We then empirically evaluate the tradeoff between rate adaptation (RA) and self-triggered control (ST) in communication cost and control performance under different operating conditions.

7.1 Systems Settings

7.1.1 Physical System Settings. We run simulations of a realistic load positioning system [60, 61], which positions a load (L) using a motor with a ballscrew transmission. The motor is attached rigidly to a movable base platform (B). The load positioning is a 4-state nonlinear system as described in Reference [61]. When the system is operated at low rates, as in real industrial applications, the stiffness of the ballscrew and the potential energy stored in it are neglected in the model. The system can be simplified as a 4-state linear system [60]:

$$\dot{x}_t = A_c x_t + B_c u_t, y_t = C_c x_t, \tag{19}$$

$$A_c = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -d_L\left(\frac{1}{m_L} + \frac{1}{m_B}\right) & \frac{k_B}{m_B} & \frac{d_B}{m_B} \\ 0 & 0 & 0 & 1 \\ 0 & \frac{d_L}{m_B} & -\frac{k_B}{m_B} & -\frac{d_B}{m_B} \end{bmatrix}, B_c = \begin{bmatrix} 0 \\ \frac{1}{m_L} + \frac{1}{m_B} \\ 0 \\ -\frac{1}{m_B} \end{bmatrix}, C_c = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}.$$

Here, d_L, m_L, d_B, m_B, and k_B are parameters of the load and base platforms, such as the mass, damping, and stiffness. The state vector is defined as $x_t = [x_L(t)\ \dot{x}_L(t)\ x_B(t)\ \dot{x}_B(t)]^T$, where x_L is the displacement of the load *relative* to the base platform, x_B is the *absolute* displacement of the base platform, and $\dot{x}_L$ and $\dot{x}_B$ are the speeds of the relative and absolute movements. We will stabilize the states of the load positioning system to the origin.

There are two kinds of plants. The first kind is denoted as PLANT1, $d_L = 15$, $m_L = 100$, $d_B = 10$, $m_B = 10$, $k_B = 5$, and $K = [-1.9393\ \ -13.1373\ 0.0842\ \ -13.0264]$. The second kind is denoted as PLANT2, $d_L = 10$, $m_L = 15$, $d_B = 3$, $m_B = 5$, $k_B = 2$, and $K = [-1.0076\ -0.6317\ -0.1954\ -0.3814]$. The second kind has lower mass and damping, therefore their response time is shorter than that of PLANT1. In holistic controller, we discretize the continuous-time models (19) using *step-invariant transformation* at its corresponding sampling sampling period T_i: $A_{T_i} = e^{A_c T_i}$, $B_{T_i} = \int_0^{T_i} e^{A_c \tau} d\tau B_c$.

For each control loop, given the discrete-time model, K, and Q, we can get P, α_1, α_2, and β according to Equations (16), (5), and (6), respectively. For all loops, $Q = I_4$, $W_Q = I_4$, $W_R = 1$, $\gamma = 1$, and $\delta = 2$. We will adjust and evaluate some parameter selections of RA, such as V_{Ith}, λ and τ of heuristic RA algorithm, and ϵ_2 of the optimal RA problem.

7.1.2 WSAN Settings. The network protocols for RA and ST use Contiki [62]. The LWB operates at the rate $R = 1$ Hz. The global static schedule has one synchronization slot, with a length of 25 ms, and 2–5 data slots, with lengths of 18 ms. 70 nodes participate in the transmissions. The synchronization packet is disseminated by the sink node (node 164) every 1 s. The synchronization packet size is 6 bytes, and the data packets are 25 bytes. Each data slot is used to transmit the control

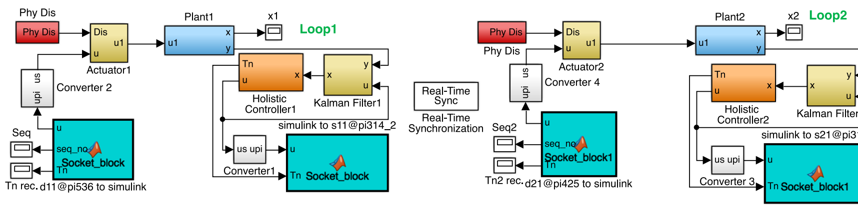


Fig. 11. SLDRT modules of WCPS-RT.

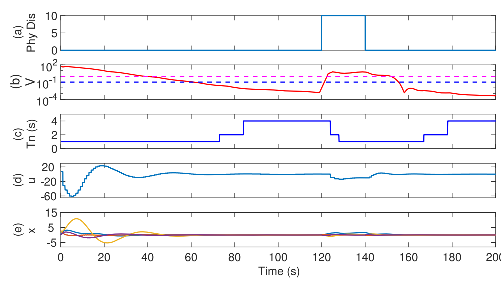


Fig. 12. Response curve of heuristic rate adaptation.

command u_t and network reconfiguration signals R or Tn of each control loop. Figure 9 shows the source and destination pairs of five actuation flows over three-floor WSAN. The Tx power is 0 dbm, and the retransmission number is 3.

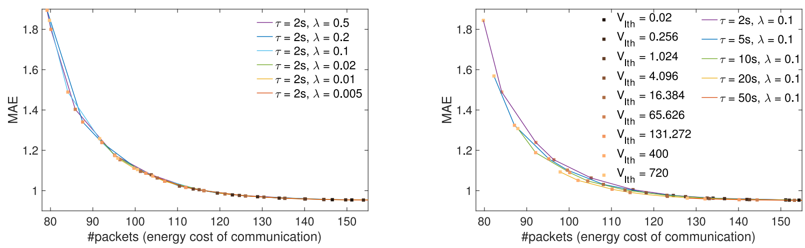
7.1.3 WCPS-RT Settings. We simulate the WNCS using WCPS-RT, which integrates a 70-node WSAN and SLDRT. We simulated two control loops sharing a WSAN for statistical results from Section 7.3 to Section 7.6. Loop l_1 controls a PLANT1. Loop l_2 controls a PLANT2. The SLDRT modules of two loops are shown in Figure 11. Each loop has its own holistic controller, and the controllers and the actuators communicate via actuation flows sharing the same WSAN. And we simulate five control loops sharing a WSAN to show the scalability of WCPS-RT in Section 7.7. Loops l_1, l_3, and l_5 control three PLANT1s separately. Loops l_2 and l_4 control two PLANT2s.

As presented in Section 6.1, modules in Figure 11 operate at different rates. The physical plants run at 100 Hz. Kalman filters and actuators run at 1 Hz. The "worst-case-guaranteed" WSAN and controllers run at 1 Hz, and WSAN and controllers can adjust their rates and operate ST during run time, based on control needs. In RA, we choose candidate rates: 1 Hz, 0.5 Hz, 0.25 Hz, which are reasonable rates for our load positioning systems with time constants of roughly 30 s. And they are also typical rates in industrial process control [40]. To provide robustness guarantees of the self-triggered control [12], we set the upper bound of the inter-transmission interval as 10 s.

7.2 Evaluation of Optimal and Heuristic Rate Adaptation Algorithms

We first evaluate the optimal and heuristic RA algorithms. Since they are control designs, we temporarily run simulations under ideal network (100% packet delivery ratio and no latency) in Section 7.2. We then run network-in-the-loop simulations with different holistic control approaches under different physical and network conditions in Sections 7.3–7.7.

7.2.1 Heuristic Rate Adaptation. We first evaluate the online heuristic RA. Figure 12 shows how heuristic RA works. Take PLANT1 as an example. We introduce physical disturbance by injecting a constant bias into the actuator from 120 to 140 s, as shown in plot (a). Plot (b) shows the Lyapunov



(A) Various λ *(B) Various test interval τ*

Fig. 13. Impacts of parameters in heuristic RA.

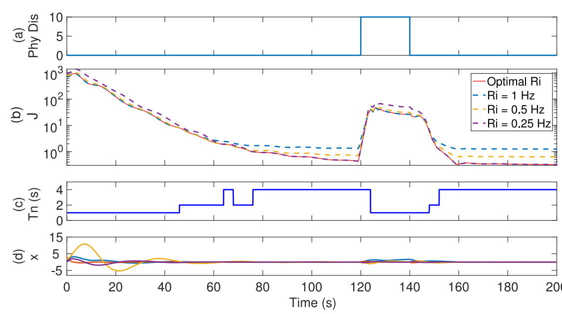


Fig. 14. Response curve of optimal RA.

function $V(x_t)$. The two dashed lines, from upper to lower, are the thresholds for increase and decrease of rate. Plot (c) shows the sampling rate adaptation. Tn indicates the time till the next packet Tx, i.e., sampling period in RA. Plots (d) and (e) show the control command u_t and physical states x_t, respectively. During the transient ($0 - 60$ s and $120 - 160$ s), the control performance is poor, which is reflected by a large value of $V(x_t)$. When $V(x_t)$ is above the increase threshold, the holistic controller increases the rate. When x is approaching the origin ($80 - 120$ s and $170 - 200$ s), as indicated by the decreases of $V(x_t)$, and $V(x_t)$ is below the decrease threshold for $\tau = 10$ s, the rate of the WSAN decreases, as shown in (c).

Figure 13 shows the impact of parameter tuning in heuristic rate selection, i.e., the increased threshold V_{Ith}, decreased threshold coefficient λ, and the test time interval τ. Each marker in this figure is obtained by carrying out 20 rounds of simulations. We use the mean absolute error (MAE) as the metric of control performance, and the number of packets sent through WSAN as the metric of energy cost. The value of X-axis is the mean of MAEs in 20 rounds of simulations, and the value of Y-axis is the mean of total number of packets. As described in Section 4.1.1, $V_{Ith} = \alpha_1||x_{se}||^2$, $V_{Dth} = \lambda\alpha_1||x_{se}||^2$. Thus, the intuition is that smaller V_{Ith} makes rate increase more often, and smaller λ and longer τ make rate decrease less often. We can see that with fixed τ and λ, the MAE decreases at the cost of more network energy consumption as V_{Ith} becomes smaller. There is a diminishing return of control performance improving as increase of energy cost. Figure 13(a) shows that λ mostly does not affect the trajectory of curves, which indicate relationship between MAE and network energy cost. With fixed V_{Ith} and τ, MAE decreases at more energy cost when λ is smaller. The same holds for τ in Figure 13(b), that MAE decreases at more energy cost when τ is longer.

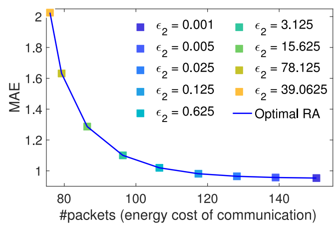


Fig. 15. Impact of ϵ_2 in optimal RA.

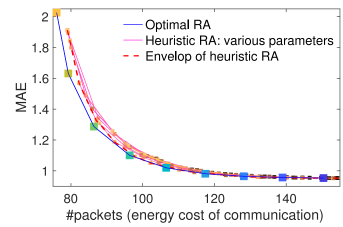


Fig. 16. Compare optimal RA and heuristic RA.

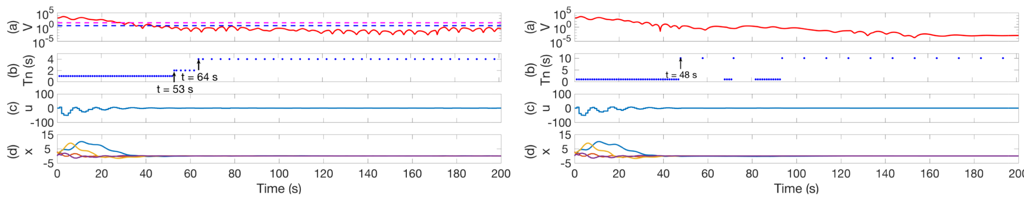
7.2.2 Optimal Rate Adaptation. We evaluate the optimal RA. Figure 14 shows how optimal RA works. Plot (b) shows the values of objective function $\mathcal{J}$ of three candidate rates and the optimal rate solution of the optimization problem (15). During the transient processes around $0 - 50$ s and $125 - 150$ s, the control performance dominates $\mathcal{J}$. Therefore, high sampling rate minimizes $\mathcal{J}$. While when the system is stable during $80 - 120$ s and $180 - 200$ s, the communication cost dominates $\mathcal{J}$. Thus, low rate minimizes $\mathcal{J}$. In this way, optimal rate selection facilitates a systematic balance between energy cost and control performance through adjusting the weight (ϵ_2) in $\mathcal{J}$. Figure 15 shows the relationship between MAE, communication cost, and ϵ_2. Each marker is obtained by 20 rounds of simulations. Larger ϵ_2 reduces energy cost at the cost of control performance and vice versa. Considering the diminishing return of MAE improvement, proper ϵ_2 could be chosen to achieve small MAE at the cost of reasonable network energy cost.

7.2.3 Comparison between Optimal and Heuristic Approaches. The optimal RA is able to systematically balance energy and control performance. It does not need any threshold compared to heuristic RA. However, since we propose to adapt the rate at run time, computational complexity of the algorithms matters. Optimal RA problem is an integer programming problem. Its computational complexity is higher than the heuristic approach with the complexity of $O(1)$. Figure 16 compares the performance of optimal and heuristic approaches. The markers on the lower left of the figure indicate better performance that can achieve smaller MAE with less energy cost. We can see that the optimal RA has slightly better performance than the best envelop of the heuristic approach. The envelop is achieved by arbitrarily tuning combinations of parameters 90 times as shown in Figure 13. However, tuning ϵ_2 in optimal RA is more efficient to balance energy and control performance. However, we can also see that the advantage of optimal RA is less remarkable when the requirement of control performance is stringent, as shown in the right bottom part of Figure 16. Since we have a specific and stringent requirement on control performance, i.e., $||x_{se}||^2 = 0.1$, $\lambda = 0.1$ and $\tau = 10$ s, we choose to adopt heuristic RA in the rest of sections.

7.3 Normal Network and Physical Conditions

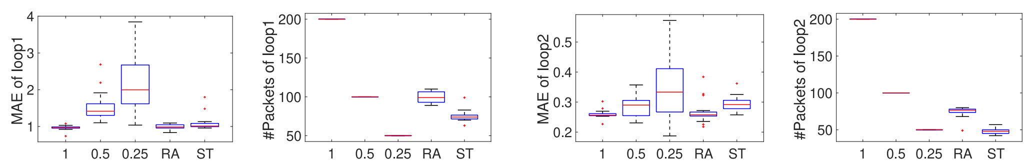
We then run network-in-the-loop simulations. We evaluate the WNCS under normal conditions. The WSAN operated on IEEE 802.15.4's channel 26. The average packet delivery ratio is 99.15%. And there is no physical disturbance. We present the results of five sets of network-in-the-loop simulations under the different management approaches:

(1) RA: Figure 17(a) shows the response curves of loop 1. In plot (b), each dot indicates Txs of one packet, and the y-axis of the dot is the time till the next Tx. When x is approaching the origin, as indicated by the decreases of $V(x_t)$, and $V(x_t)$ is below the decrease threshold for $\tau = 10$ s, the rate of the WSAN starts to decrease, as shown in (b). The rate changes from 1 Hz (1 Tx every 1 s) to 0.5 Hz (1 Tx every 2 s) at $t = 53$ s, then to 0.25 Hz (1 Tx every 4 s) at $t = 64$ s.



(A) *time-driven control with rate adaptation* (B) *self-triggered control*

Fig. 17. Response curve under normal condition.



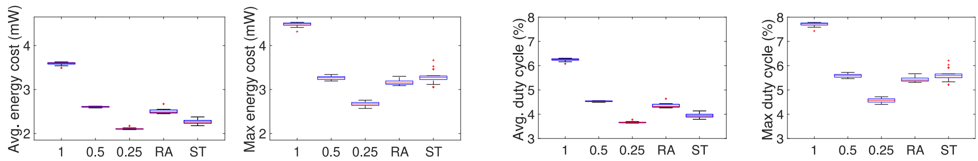
(A) *performance of loop 1* (B) *performance of loop 2*

Fig. 18. Performance under normal condition.

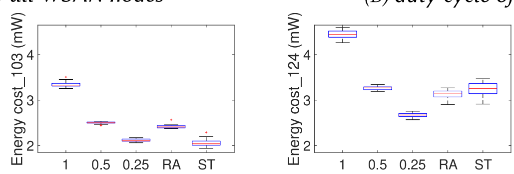
(2) ST: Figure 17(b) shows the response curve of ST. In (b), since $V(x_t)$ decreases, the inter-transmission interval changes from 1 s to 10 s at $t = 48$ s. When $V(x_t)$ increases at around 60 s to 90 s, Tn reduces to 1 s as soon as the timer expires.
(3) Fixed rate time-driven control: Existing WSANs typically employ time-drive transmissions with fixed rates, so we use three fixed rates of 1 Hz, 0.5 Hz, and 0.25 Hz, denoted by 1, 2, 4 in following statistical results.

Next, we run each experiment for 20 rounds with different initial values to statistically compare different approaches. Figure 18 shows the performances of two loops. Both RA and ST can achieve similar control performances with fixed rate of 1 Hz, with a network cost (# of packets) reduction of more than 50%. Loop 2 has network cost reduction of more than 62%, since it has shorter time constant. For both loops, ST is more aggressive in saving network cost than RA.

In reality, the total energy cost, including the synchronization cost, is of interest. Therefore, we analyze power cost over the WSAN in detail. We collect the time spent in transmitting and listening per node per second using the Energest module [63] provided by Contiki OS. The sum of transmitting and listening time is the radio-on time of the collection period, and the node sleeps in the rest of the period. We adopt the energy model in Reference [64] to estimate the energy cost. Figure 19(a) and Figure 19(b) show that the energy costs are consistent with duty cycle. Figure 19(a) shows the average energy cost of all 70 nodes is consistent with the number of packets going through WSAN. RA and ST save 40% energy, which is higher than energy cost of loop1 and loop2 alone in Figure 18, since energy estimation includes the cost of synchronization every second. However, in the case of the maximum energy cost, ST costs more than RA, which can be explained by the fact that the node incurs the maximum energy cost due to packet loss. Facing packet loss, the node with the ST protocol keeps listening at a high energy cost because of its recovery mechanism. Whereas the node with the RA protocol applies self-recovery mechanism without extra energy cost. To verify this difference, we analyze the power cost of two nodes. Node 103 has a higher packet reception ratio than node 124. Figure 19(c) shows that ST transmissions are not as efficient as RA for node 124, due to its recovery mechanism. Figure 20 shows the relationship between MAE and energy cost under normal condition. Each data point indicates the MAE and energy cost



(A) *energy cost of all WSAN nodes* (B) *duty cycle of all WSAN nodes*



(C) *energy cost of different nodes*

Fig. 19. Energy cost under normal condition.

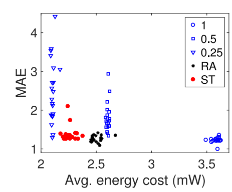
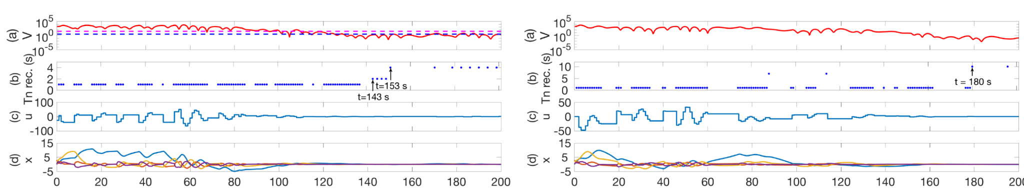


Fig. 20. Relationship between MAE and energy cost under normal condition.



(A) *time-driven control with rate adaptation* (B) *self-triggered control*

Fig. 21. Response curve under network interference.

of one round of experiment. Data points of RA and ST are concentrated in the bottom left of the figure, which indicates that those approaches achieve smaller MAE with lower energy cost.

7.4 Network Interference

We operate WSAN over channel 22 (2.460 GHz) of IEEE.802.15.4, and we introduce network interference by continuously sending jamming packets over an overlapping channel 11 (2.462 GHz) of Wi-Fi. The average packet delivery ratio is reduced to 65.9%. Figure 21 shows the response curves of RA and ST. In plot (b), each dot indicates that the actuator receives a packet. Both methods stay longer at high rate than in normal condition to compensate the impact of interferences. And both the network protocols can recover from packet drops. Figure 22 shows the statistical results under network interference. In this case, both RA and ST guarantee the control performance, at the cost of more energy consumption than Section 7.3. ST consumes more energy than RA, due to its packet loss recovery mechanism. Figure 23 shows the relationship between MAE and energy cost under network interference. Data points of RA and ST are concentrated in the bottom middle, which indicates that those approaches achieve smaller MAE with higher energy cost than normal

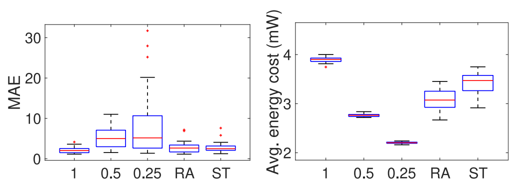


Fig. 22. Performance under network interference.

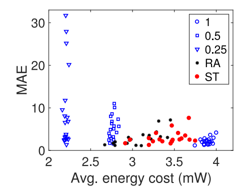
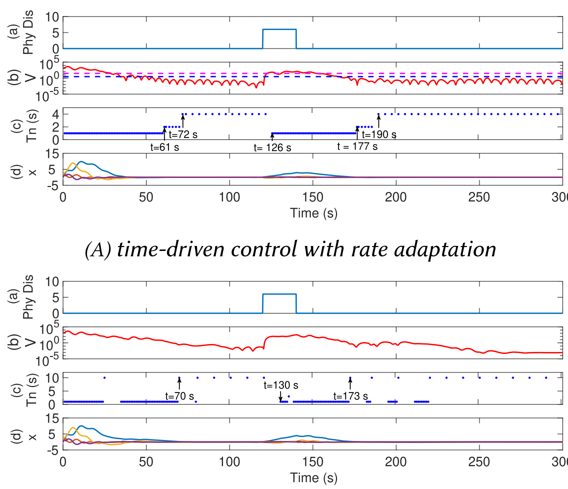


Fig. 23. Relationship between MAE and energy cost under network interference.



(A) time-driven control with rate adaptation

(B) self-triggered control

Fig. 24. Response curve under physical interference.

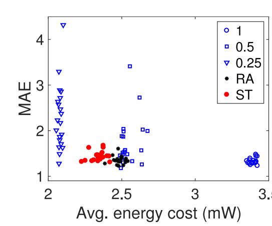
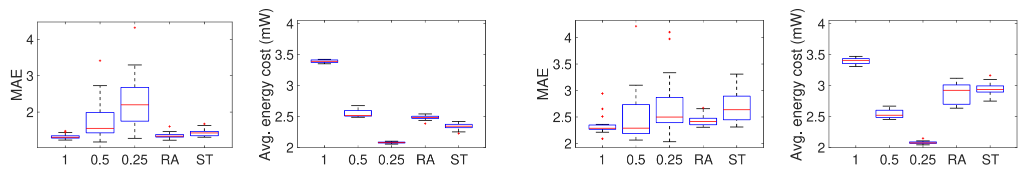


Fig. 25. Relationship between MAE and energy cost under physical disturbance.

case due to recovery from network interference, but still lower than 1 Hz sampling. The simultaneous increase of both MAE and energy cost can be explained by the intuition of efficient holistic control that poorer system performance will cause the increase of the rates and number of events. However, no extra energy is cost when the system is in good condition. This trend indicates that network resources are adapted well based on the states of the physical plants.

7.5 Physical Disturbance

We introduce physical disturbance by adding a constant bias to actuators from 120 s to 140 s. As shown in Figure 24, both RA and ST adapt rates to 1 Hz under the physical disturbance. However, the time ST (t = 130 s) reacts to the disturbance is later than RA (t = 126 s), since ST has longer Tn (10 s). Figure 26 shows the statistical results. In Figure 26(a), both RA and ST have similar MAE with a fixed rate of 1 Hz and can save more than 30% of the energy. However, in Figure 26(b), the ST performs worse than RA within the interference interval. The longer Tn (10 s) makes ST response to disturbance slower than time-driven management. Figure 25 shows the relationship between MAE and energy cost under physical disturbance. Data points of RA and ST are concentrated in



(A) time interval of one round: 0 s − 300 s *(B) time interval of physical interference: 120s − 180s*

Fig. 26. Performance under physical interference.

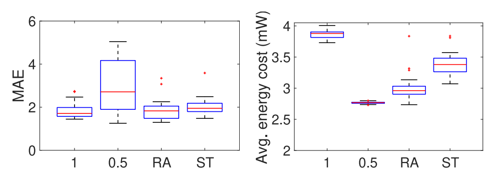


Fig. 27. Performance under network and physical interferences.

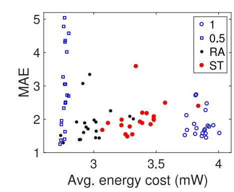


Fig. 28. Relationship between MAE and energy cost under both network and physical interferences.

the bottom left of the figure, which indicates that those approaches achieve smaller MAE with lower energy cost.

7.6 Both Network and Physical Interferences

We run experiments with both network and physical interferences in Section 7.4 and Section 7.5. Fixed rate of 0.25 Hz causes the instability of the system. Therefore, we do not show the results of 0.25 Hz. Figure 27 shows the statistical results that both RA and ST guarantee the control performance at the costs of more energy consumption than in Section 7.5. ST costs more energy than RA due to the recovery mechanism. Figure 28 shows the relationship between MAE and energy cost under both network and physical interferences. Data points RA and ST are concentrated in the bottom left, which indicates that those approaches achieve smaller MAE with lower energy cost. The simultaneous increase of both MAE and energy cost indicate that network resources are allocated properly based on the states of the physical plants.

To summarize, in normal physical and network condition, RA and ST can achieve similar control performance to a conventional fixed rate of 1 Hz while improving energy efficiency. Besides, ST is more aggressive in energy saving than RA. However, when there are interferences, RA has better performance and energy efficiency than ST, because ST has an embedded recovery mechanism, which costs more energy under packet loss, and a longer inter-transmission interval, which makes ST response slowly to disturbance.

7.7 Scalability and Flexibility of WCPS-RT

Although above experimental results are based on two control loops. WCPS-RT has the scalability to operate more control loops. In addition, it has the flexibility that end nodes of the data flows can be any nodes in the testbed. As an example, we simulate five control loops sharing a WSAN. Loops l_1, l_3, and l_5 control three PLANT1s. Loops l_2 and l_4 control two PLANT2s. Figure 9 shows the source and destination pairs of five actuation flows over three-floor WSAN. Table 4 shows the MAEs and energy costs in one round (200 s) of network-in-the-loop simulation under normal

Table 4. Performance of Five-loop Simulation

	MAE1	MAE2	MAE3	MAE4	MAE5	Energy (mW)
1	0.9666	0.2891	0.9509	0.2292	0.9630	5.2730
2	1.2529	0.3158	1.2800	0.2723	1.6537	3.0461
4	1.5129	0.3131	1.6886	0.2701	1.8859	2.0233
RA	0.9435	0.2623	0.9458	0.2987	0.9671	2.7966
ST	0.9764	0.3148	1.0243	0.3151	0.9943	2.5209

condition. Loops l_1, l_3, and l_5 have larger MAEs and are more sensitive to different rates than l_2 and l_4, since l_2 and l_4 with lower mass and damping are easier and faster to stabilize. Although there is some randomness in single simulation, it is obvious that RA and ST can achieve similar control performance with fixed rate of 1 Hz while saving energy for more than 47%.

8 CONCLUSIONS

Wireless control faces significant challenges due to data loss and energy constraints in wireless networks. In this article, we present efficient holistic control approaches based on rate adaptation (RA) and self-triggered control (ST). The holistic control architecture can not only ensure control performance under wireless and physical interferences, but also reduce network energy consumption. Furthermore, we design network reconfiguration mechanisms based on LWB to support RA and ST. In addition, we build WCPS-RT that integrates MATLAB/Simulink and a three-floor WSAN testbed for experimental validation of control over real-world WSANs. Our empirical studies show that both RA and ST result in improvement of control performance and energy efficiency when compared to traditional control systems at fixed sampling rates. The advantage in energy efficiency of ST, however, diminishes under harsh physical and wireless network conditions due to the cost of recovering from data loss and physical disturbance.

REFERENCES

[1] Pangun Park et al. 2018. Wireless network design for control systems: A survey. *Commun. Surv. Tut.* 20, 2 (2018).

[2] ABB. 2020. WirelessHART Networks: 7 Myths That Cloud Their Consideration for Process Control Measurement Made Easy. Retrieved from https://new.abb.com/products/measurement-products/pl/produkty-i-rozwiazania-wireless/wirelesshart-networks-seven-myths-that-cloud-their-consideration-for-process-controll-en.

[3] Emerson^{TM}. 2017. White Paper: Emerson Wireless Security – WirelessHart and Wi-Fi. Retrieved from https://www.emerson.com/documents/automation/white-paper-emerson-wireless-security-wirelesshart-wi-fi-security-deltav-en-41260.pdf.

[4] Yehan Ma et al. 2018. Holistic cyber-physical management for dependable wireless control systems. *ACM Trans. Cyber-Phys. Syst.* 3, 1 (2018).

[5] Federico Ferrari et al. 2012. Low-power wireless bus. In *Proceedings of the ACM Conference on Embedded Network Sensor Systems.*

[6] Xiaomin Li et al. 2017. A review of industrial wireless networks in the context of Industry 4.0. *Wirel. Netw.* 23, 1 (2017).

[7] Chenyang Lu et al. 2016. Real-time wireless sensor-actuator networks for industrial cyber-physical systems. *Proc. IEEE* 104, 5 (2016).

[8] Bruno Sinopoli et al. 2004. Kalman filtering with intermittent observations. *IEEE Trans. Autom. Control* 49, 9 (2004).

[9] Manuel Mazo et al. 2008. On event-triggered and self-triggered control over sensor/actuator networks. In *Proceedings of the IEEE Conference on Decision and Control.*

[10] Manuel Mazo et al. 2011. Decentralized event-triggered control over wireless sensor/actuator networks. *IEEE Trans. Autom. Contr.* 56, 10 (2011).

[11] Manuel Mazo Jr et al. 2010. An ISS self-triggered implementation of linear controllers. *Automatica* 46, 8 (2010).

[12] José Araújo et al. 2014. System architectures, protocols and algorithms for aperiodic wireless control systems. *IEEE Trans. Ind. Inf.* 10, 1 (2014).

[13] Pangun Park et al. 2011. Breath: An adaptive protocol for industrial control applications using wireless sensor networks. *IEEE Trans. Mob. Comput.* 10, 6 (2011).
[14] Felix Dobslaw et al. 2016. End-to-end reliability-aware scheduling for wireless sensor networks. *IEEE Trans. Ind. Inf.* 12, 2 (2016).
[15] Hyung-Sin Kim et al. 2017. Load balancing under heavy traffic in RPL routing protocol for low power and lossy networks. *IEEE Trans. Mob. Comput.* 16, 4 (2017).
[16] Marco Zimmerling et al. 2017. Adaptive real-time communication for wireless cyber-physical systems. *ACM Trans. Cyber-Phys. Syst.* 1, 2 (2017).
[17] Zheng Li et al. 2006. Adaptive multiple sampling rate scheduling of real-time networked supervisory control system-part II. In *Proceedings of the Annual Conference on IEEE Industrial Electronics*. IEEE.
[18] Dip Goswami et al. 2012. Time-triggered implementations of mixed-criticality automotive software. In *Proceedings of the Conference on Design, Automation and Test in Europe*. EDA Consortium.
[19] Burak Demirel et al. 2014. Modular design of jointly optimal controllers and forwarding policies for wireless control. *IEEE Trans. Autom. Contr.* 59, 12 (2014).
[20] Abusayeed Saifullah et al. 2014. Near optimal rate selection for wireless control systems. *ACM Trans. Embed. Comput. Syst.* 13, 4s (2014).
[21] Dohwan Kim et al. 2019. Sampling rate optimization for IEEE 802.11 wireless control systems. In *Proceedings of the ACM/IEEE International Conference on Cyber-Physical Systems*.
[22] Bo Li et al. 2016. Wireless routing and control: A cyber-physical case study. In *Proceedings of the ACM/IEEE International Conference on Cyber-Physical Systems*.
[23] Jia Bai et al. 2012. Optimal cross-layer design of sampling rate adaptation and network scheduling for wireless networked control systems. In *Proceedings of the ACM/IEEE International Conference on Cyber-Physical Systems*.
[24] Lei Bao et al. 2009. Rate allocation for quantized control over noisy channels. In *Proceedings of the International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*. IEEE.
[25] J. Colandairaj et al. 2007. Wireless networked control systems with QoS-based sampling. *IET Contr. Theor. Applic.* 1, 1 (2007).
[26] Anton Cervin et al. 2003. How does control timing affect performance? Analysis and simulation of timing using Jitterbug and TrueTime. *IEEE Contr. Syst. Mag.* 23, 3 (2003).
[27] Emeka Eyisi et al. 2012. NCSWT: An integrated modeling and simulation tool for networked control systems. *Simul. Model. Pract. Theor.* 27 (2012).
[28] Kannan Srinivasan et al. 2010. An empirical study of low-power wireless. *ACM Trans. Sens. Netw.* 6, 2 (2010).
[29] Carlos Santos et al. 2015. Aperiodic linear networked control considering variable channel delays: Application to robots coordination. *Sensors* 15, 6 (2015).
[30] Bo Li et al. 2015. Incorporating emergency alarms in reliable wireless process control. In *Proceedings of the ACM/IEEE International Conference on Cyber-Physical Systems*.
[31] Philip Levis et al. 2003. TOSSIM: Accurate and scalable simulation of entire TinyOS applications. In *Proceedings of the International Conference on Embedded Networked Sensor Systems*. ACM.
[32] Miroslav Pajic et al. 2012. Closing the loop: A simple distributed method for control over wireless networks. In *Proceedings of the ACM/IEEE International Conference on Information Processing in Sensor Networks*. IEEE.
[33] Fabian Mager et al. 2019. Feedback control goes wireless: Guaranteed stability over low-power multi-hop networks. In *Proceedings of the ACM/IEEE International Conference on Cyber-Physical Systems*.
[34] Dominik Baumann et al. 2018. Evaluating low-power wireless cyber-physical systems. In *Proceedings of the IEEE Workshop on Benchmarking Cyber-Physical Networks and Systems*.
[35] Dominik Baumann et al. 2019. Control-guided communication: Efficient resource arbitration and allocation in multi-hop wireless control systems. *IEEE Contr. Syst. Lett.* 4, 1 (2019).
[36] Anders Ahlen et al. 2019. Toward wireless control in industrial process automation: A case study at a paper mill. *IEEE Contr. Syst. Mag.* 39, 5 (2019).
[37] Yehan Ma et al. 2019. Optimal dynamic scheduling of wireless networked control systems. In *Proceedings of the ACM/IEEE International Conference on Cyber-Physical Systems*.
[38] Kemin Zhou et al. 1996. *Robust and Optimal Control*. Prentice Hall, Upper Saddle River, NJ.
[39] Federico Ferrari et al. 2011. Efficient network flooding and time synchronization with glossy. In *Proceedings of the International Conference on Information Processing in Sensor Networks*. IEEE.
[40] Gang Zhao. 2011. Wireless sensor networks for industrial process monitoring and control: A survey. *Netw. Prot. Algor.* 3, 1 (2011).
[41] Timofei Istomin et al. 2016. Data prediction+ synchronous transmissions= ultra-low power wireless sensor networks. In *Proceedings of the ACM Conference on Embedded Network Sensor Systems*. ACM.
[42] Tongwen Chen et al. 2012. *Optimal Sampled-data Control Systems*. Springer Science & Business Media.

[43] Bruce Francis. 1979. The optimal linear-quadratic time-invariant regulator with cheap control. *IEEE Trans. Autom. Contr.* 24, 4 (1979).
[44] Hai Lin et al. 2009. Stability and stabilizability of switched linear systems: A survey of recent results. *IEEE Trans. Autom. Contr.* 54, 2 (2009).
[45] Wei Zhang et al. 2001. Stability of networked control systems. *IEEE Contr. Syst. Mag.* 21, 1 (2001).
[46] Huijun Gao et al. 2008. A new delay system approach to network-based control. *Automatica* 44, 1 (2008).
[47] Merid Lješnjanin et al. 2014. Packetized MPC with dynamic scheduling constraints and bounded packet dropouts. *Automatica* 50, 3 (2014).
[48] Jing Wu et al. 2007. Design of networked control systems with packet dropouts. *IEEE Trans. Autom. Contr.* 52, 7 (2007).
[49] Jamal Daafouz et al. 2002. Stability analysis and control synthesis for switched systems: A switched Lyapunov function approach. *IEEE Trans. Autom. Contr.* 47, 11 (2002).
[50] Daniel Liberzon et al. 1999. Basic problems in stability and design of switched systems. *IEEE Contr. Syst. Mag.* 19, 5 (1999).
[51] S.-H. Lee et al. 2000. A new stability analysis of switched systems. *Automatica* 36, 6 (2000).
[52] Dragan Nesic et al. 2004. Input-output stability properties of networked control systems. *IEEE Trans. Autom. Contr.* 49, 10 (2004).
[53] Bin Hu et al. 2019. Co-design of safe and efficient networked control systems in factory automation with state-dependent wireless fading channels. *Automatica* 105 (2019).
[54] Simulink. 2020. Simulink Desktop Real-Time. Retrieved from https://www.mathworks.com/products/simulink-desktop-real-time.html.
[55] Mo Sha et al. 2015. Implementation and experimentation of industrial wireless sensor-actuator network protocols. In *Proceedings of the European Conference on Wireless Sensor Networks*. Springer.
[56] Mo Sha et al. 2017. Empirical study and enhancements of industrial wireless sensor–actuator network protocols. *IEEE Internet Things J.* (2017).
[57] PTP. 2020. PTP Protocol. Retrieved from https://www.nist.gov/el/intelligent-systems-division-73500/ieee-1588.
[58] Xiangheng Liu et al. 2004. Kalman filtering with partial observation losses. In *Proceedings of the Conference on Decision and Control*. IEEE.
[59] Yang Shi et al. 2010. Kalman filter-based identification for systems with randomly missing measurements in a network environment. *Int. J. Contr.* (2010).
[60] Vijay Shilpiekandula et al. 2012. Load positioning in the presence of base vibrations. In *Proceedings of the American Control Conference*.
[61] Yu Jiang et al. 2015. Optimal codesign of nonlinear control systems based on a modified policy iteration method. *IEEE Trans. Neural Netw. Learn. Syst.* (2015).
[62] Chayan Sarkar. 2016. LWB and FS-LWB implementation for Sky platform using Contiki. *arXiv preprint arXiv:1607.06622* (2016).
[63] Adam Dunkels et al. 2007. Software-based on-line energy estimation for sensor nodes. In *Proceedings of the Workshop on Embedded Networked Sensors*. ACM.
[64] Chengjie Wu et al. 2016. Maximizing network lifetime of WirelessHART networks under graph routing. In *Proceedings of the IEEE International Conference on Internet-of-Things Design and Implementation*.

Received December 2018; revised August 2019; accepted November 2019