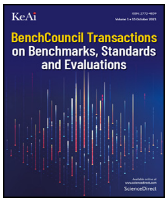
BenchCouncil Transactions on Benchmarks, Standards and Evaluations 2 (2022) 100036

Contents lists available at ScienceDirect

BenchCouncil Transactions on Benchmarks, Standards and Evaluations

journal homepage: https://www.keaipublishing.com/en/journals/benchcouncil-transactions-on-benchmarks-standards-and-evaluations/



Challenges and recent advances in the design of real-time wireless Cyber-Physical Systems

Romain Jacob

ETH Zurich, Switzerland

ARTICLE INFO

Keywords:
Synchronous transmissions
Wireless Cyber-Physical Systems
Replicability
Real-time systems

ABSTRACT

Cyber-Physical Systems (CPS) refer to systems where some intelligence is embedded into devices that interact with their environment. Using wireless technology in such systems is desirable for better flexibility, improved maintainability, and cost reduction, among others. Moreover, CPS applications often specify deadlines; that is, maximal tolerable delays between the execution of distributed tasks. Systems that guarantee to meet such deadlines are called real-time systems. In the past few years, a technique known as synchronous transmissions (ST) has been shown to enable reliable and energy efficient communication, which is promising for the design of real-time wireless CPS.

We identify at least three issues that limit the adoption of ST in this domain: (i) ST is difficult to use due to stringent time synchronization requirements (in the order of µs). There is a lack of tools to facilitate the implementation of ST by CPS engineers, which are often not wireless communication experts. (ii) There are only few examples showcasing the use of ST for CPS applications and academic works based on ST tend to focus on communication rather than applications. Convincing proof-of-concept CPS applications are missing. (iii) The inherent variability of the wireless environment makes performance evaluation challenging. The lack of an agreed-upon methodology hinders experiment reproducibility and limits the confidence in the performance claims. This paper synthesizes recent advances what address these three problems, thereby enabling significant progress for future applications of low-power wireless technology in real-time CPS.

1. Introduction

Cyber-Physical Systems (CPS) are understood as systems where "*physical and software components are deeply intertwined, each operating on different spatial and temporal scales, exhibiting multiple and distinct behavioral modalities, and interacting with each other in a myriad of ways that change with context*" [1]. The domains of application of CPS are very diverse: *e.g.*, robotics, distributed monitoring, process control, power-grid management [2–4].

It is important to realize that the design of CPS encompasses three main aspects, mapping to as many research fields, with their own purpose and goals: The *embedded hardware design* aims to extend the amount of computational resources available (*e.g.*, processing power, memory, sensors and actuators) while limiting the cost, form factor, and energy consumption of a device. The *communication*, either wired or wireless, aims to transmit messages between distributed devices efficiently; that is, quickly and using little energy. Finally, the *distributed system design* realizes the implementation of the CPS functions, such as *e.g.*, remote monitoring and control of distributed processes.

The goal of the overall design is to reliably provide the specified CPS functions. Achieving this goal relies on hardware and communication; however reaching "perfect" communication, such as 100% packet reception rate, is not a goal in itself; it is merely a mean to an end. What truly matters is to fulfill the system functionality. Typically, CPS design aims to provide end-to-end performance guarantees, such as meeting hard deadlines between the execution of distributed tasks; *e.g.*, between the start of a sensing task to the end of the corresponding actuation tasks (Fig. 1). Meeting such deadlines is called *providing real-time guarantees*.

The potential benefits of wireless communication for CPS applications are well-known and include simpler deployment and maintenance, cheaper operational costs, lighter weight [5]. Furthermore, wireless is the only viable option in application domains including highly mobile nodes, such as an automated warehouse with transport robots [6] or teams of drones [7]. However, CPS applications have challenging performance requirements [8], which are hard to fulfill with a wireless design.

E-mail address: jacobr@ethz.ch.
URL: https://www.romainjacob.net/.

https://doi.org/10.1016/j.tbench.2022.100036
Received 13 January 2022; Received in revised form 17 March 2022; Accepted 17 March 2022

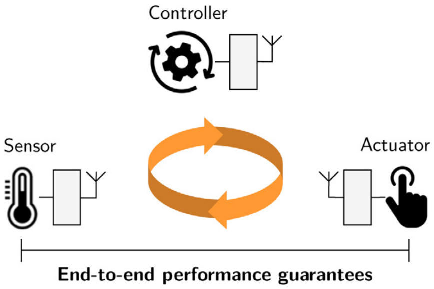


Fig. 1. The prime objective of CPS design is to provide end-to-end performance guarantees for distributed applications. In this paper, we consider synchronous transmissions, a recent development in low-power wireless communication, and demonstrate how to leverage the technique to provide real-time guarantees in wireless CPS.

1.1. *Requirements of wireless CPS*

CPS applications are subject to different types of requirements, such as the specified end-to-end latency, bandwidth, or number of devices; the precise performance level for these requirements depends on the application context. Generally, CPS requirements belong to one of the following classes:

Reliability A large ratio of messages is successfully transmitted wirelessly.

Adaptability The system adapts to runtime changes in resource demands.

Mobility The system supports mobile devices.

Timeliness Applications meet their deadlines, which are often specified end-to-end. Depending on the class of systems, deadlines can be either soft or hard [9].

Efficiency The system supports short end-to-end latency, scales in terms of system size, and optimizes its energy and bandwidth utilization.

These requirements are mutually conflicting. For example, reducing the energy consumption is typically achieved by keeping the radio turned off whenever possible. However, this directly conflicts with *Adaptability*, as the system cannot adapt reliably without exchanging some extra messages. In general, there is a price to pay in terms of *Efficiency* for meeting any of the other requirements. Hence, designing CPS consists in exploring the design space for relevant trade-offs; that is, the design optimizes the overall system *Efficiency* while meeting other application requirements.

1.2. *Traditional wireless networking*

Low-power wireless communication is a mature field of research, heavily studied for more than two decades. A large part of the research focused on wireless sensor networks, where low power consumption is a key requirement to enable long-term operation of the deployed networks, with specifications up to multiple years of operation on small batteries. Many successful applications and deployments include monitoring of soils [10], permafrost [11], buildings [12], or wildlife [13,14].

In these scenarios, the distributed application often remains simple (*e.g.*, collect sensor readings). The main challenge is to reliably aggregate or disseminate messages across a multi-hop network. *Single-hop* communication refers to the case where a source node is in communication range from its destination. This is a rather simple case, but the deployed networks often span large areas whereas low-power radios can typically communicate in the range of tens of meters. Thus, *multi-hop* communication is required, whereby a source node must rely on other nodes in the network to forward its messages, hop after hop, until the destination is reached. This is the same principle as in the children's game known as Chinese whispers [15]; if you ever played, you know that the original message hardly ever reaches the end of the chain successfully.

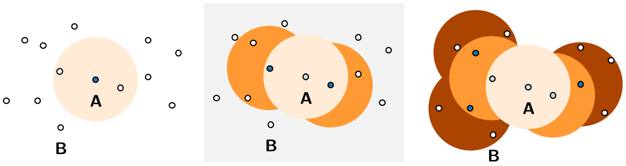
Multi-hop communication is a collaborative task for which the nodes must be coordinated. Indeed, if a node transmits a message while another wireless communication is ongoing, the transmissions will interfere and they may both fail. Furthermore, the radio frequency bands used for wireless communication cannot be isolated. Other networks are potentially exchanging messages on the same frequencies, which generates external interference and triggers packet losses. As a result, traditional multi-hop communication requires complex mechanisms for coordinating the nodes, scheduling the different transmissions to forward all messages throughout the network, and retransmitting messages that have been lost (*e.g.*, due to external interference). The complexity is further increased in mobile scenarios, where the set of neighboring nodes (which may relay a node's messages) changes frequently. The traditional wireless networking approach performs multi-hop communication by carefully planning a sequence of unicasts (*i.e.*, one-hop transmissions), usually performed along one or a few of the shortest paths possible between a message source and its destination [16–18]. Intuitively, this is efficient because only the necessary nodes are involved in relaying a message.

In practice however, multi-hop wireless network are sensitive to topology changes, external interference, and traffic congestion. These limit the reliability of communication, which has been a major obstacle to the utilization of wireless technology in CPS: for a long time, it has been considered impossible to provide the required level of reliability using wireless [19]. Synchronous transmissions have fundamentally changed that.

1.3. *Synchronous transmissions*

Synchronous transmissions (ST), also referred to as concurrent transmissions, is a technique consisting in letting multiple nodes transmit a message at the "same time" (hence the name of *synchronous* transmissions). A destination node can successfully receive (one of these) synchronous transmissions thanks to two effects taking place at the physical layer: constructive interference and the capture effect [20,21]. In a nutshell, ST is likely to be successful if the incoming messages arrive at the receiving node's antenna within a small time offset (in the range of a few symbol periods—tens of μs—depending on the physical layer and the effect considered). ST has been shown to work both analytically [22], empirically [23], and on different physical layers, such as IEEE 802.15.4 [24], Bluetooth [25], and LoRa [26].

The use of ST in low-power communication, pioneered by Glossy [23] in 2011, has triggered a paradigm shift in the low-power wireless community: ST can be leveraged to implement efficient broadcast in a multi-hop network using network-wide flooding (Fig. 2). The flooding procedure implemented by Glossy is illustrated in Fig. 3. A first node initiates the flooding process. The 1-hop neighbors of the initiator receive the message and synchronously broadcast this same message in the next time step, which is then received by the initiator's 2-hop neighbors with high probability, thanks to ST. The process repeats following the same logic: a node that receives a packet broadcasts it again in the next time slot. Each node in the network transmits each packet up to N times, after which the flood terminates. It has been shown in a wide range of scenarios that, with $N = 3$, Glossy achieves a reliability above 99.99% [23]; that is, 99.99% of the floods are successfully received by nodes in the network. With $N = 5$, the average reliability reaches 99,999% [23]. Glossy achieves such high reliability by leveraging spatio-temporal redundancy. Packets are transmitted along all possible paths; in other words, they are implicitly routed everywhere, and therefore avoid interference sources localized in space. In addition, having each node transmitting N times creates temporal redundancy,



Node **A** initiates the flood.

The 1-hop neighbors of **A** synchronously repeat.

Node **B** receives the message.

Fig. 2. Flooding of a message from node **A** to node **B**.

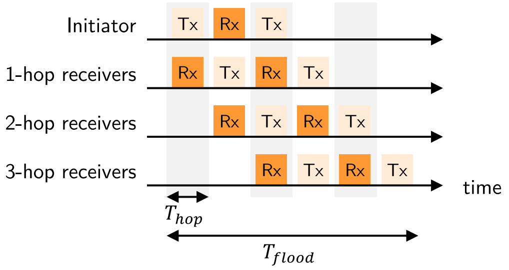


Fig. 3. Glossy operation in a 3-hop network with 2 transmissions per node (N).

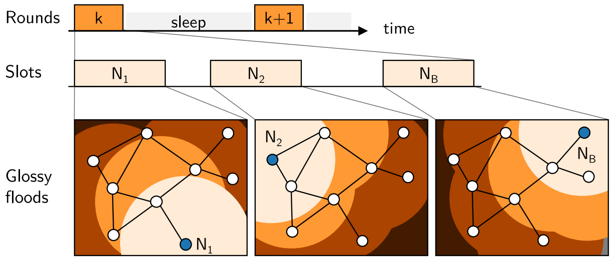


Fig. 4. Thanks to synchronous-transmissions-based flooding, a multi-hop network can be abstracted and scheduled like a shared bus. Communication is organized in rounds, composed of time slots; in each time slot, a node initiates a flood which allows to send a message to any other node(s) in bounded time. This mimics the operation of classical field bus, but with a wireless design.

thereby avoiding interference sources localized in time. Moreover, the predictability of the operation timing in ST-based flooding can be leveraged to perform distributed time synchronization. Glossy demonstrated that sub- μs synchronization accuracy can be achieved in a multi-hop network composed of tens to hundreds of nodes [23]. Since Glossy, other flooding strategies have been proposed [27–29], but the overall principle remains the same.

The key benefit of ST is that, thanks to the provided multi-hop broadcast primitive, the overall communication design can be drastically simplified. Essentially, one can abstract the underlying multi-hop topology as a *virtual single-hop network*, which can be scheduled like a shared bus: any node can send a message to any other node(s) in the network in bounded time. The only requirement is that no other node is using the "bus" at the same time. This design, first proposed with the Low-power Wireless Bus protocol [30], has been adapted into many flavors (see [31] for a recent survey) with always a similar concept: communication is organized in rounds, between which nodes keep their radio turned off to save energy. Each round is composed of time slots, which are assigned to certain nodes for communication. In each of these slots, nodes execute a flooding primitive (*e.g.*, Glossy) thereby preforming a one-to-all communication (Fig. 4). Consequently, the complexity of performing reliable multi-hop communication (see Section 1.2) is significantly relaxed. Thanks to ST, multi-hop communication is reduced to the scheduling of a single shared resource, a well-understood and relatively easy problem [9].

A priori, flooding seems to be a wasteful approach: every message sent by any node will be received and forwarded by every other node in the network. However, the simplicity and reliability of the approach actually pays off. (i) Since the flooding logic is simple, it requires little communication overhead for the coordination of the network; nodes mostly send application data. (ii) The spatio-temporal redundancy embedded in the flooding process makes it very reliable; once a flood is completed, there is hardly ever a need to further retransmit a message in a subsequent flood. (iii) Finally, since multiple nodes can transmit simultaneously, the flooding process completes quickly; very close to the theoretically optimal speed [23].

Thus, with flooding approaches based on ST, the energy cost of sending one byte of data is relatively high (since this byte will be retransmitted by all the nodes), but the *overall cost* for communication remains relatively small, thanks to the limited protocol overhead and the absence of need for further retransmissions. The energy efficiency and reliability of ST-based flooding has been demonstrated in many research contributions (*e.g.*, [23,32,33]) and showcased in the EWSN Dependability Competitions [34], where all wining solutions in the past four years (2016 to 2019) were based on ST [27,29,35–37].

The downside of ST is that it is difficult to use in more complex system designs, such as those envisioned for wireless CPS [8]. The difficulty stems from the tight timing requirements for successful ST: to be received reliably, transmissions must be initiated by the different nodes within few μs. Practically, this implies that the runtime execution of a node is governed by the communication protocol, which makes the implementation of advanced distributed tasks complex and error-prone. As a consequence, ST has thus far been mainly used for academic endeavors and mostly in wireless sensor network scenarios where the application tasks are typically simple and non-critical. Collecting a new sensor reading is a task that can usually tolerate being delayed by a few milliseconds while communication is ongoing. This is not acceptable for wireless CPS in general.

1.4. *The dual-processor platform*

In CPS, each device must perform application and communication tasks in order to fulfill the overall system functions; this poses the challenge of interference between tasks which contend for processor execution time. This interference problem can be mitigated by a new breed of embedded platforms featuring multiple processing cores, such as the NXP LPC541XX [38] or the VF3xxR [39]. On the one hand, this helps because applications and communication tasks can be processed in parallel, but on the other hand, it creates contention for the access to the resources shared between the cores. Efficient scheduling of multi-core platforms is a complex problem and a research field of its own.

Instead of resolving contention by scheduling, another approach proposed in the literature attempts to *prevent interference by design*. This principle, soberly called the Dual-Processor Platform (*DPP* [40]), consists in linking two processors with a processor interconnect called *Bolt* (Fig. 5). *Bolt* [41] provides predictable asynchronous message passing between two arbitrary processors while decoupling these processors with respect to time, power, and clock domains. The lower part of Fig. 5 shows a conceptual view of the *DPP*, including two message queues with first-in-first-out (FIFO) semantics, one for each direction, which are the only communication channels between the interconnected processors. The guiding principle of *Bolt* design is to limit the interference between the interconnected processors as much as possible, then to provide formally verified bounds on the unavoidable interference remaining. Concretely, this means that the *Bolt* API functions, used by the processors to exchange messages, have hard latency bounds. The upper part of Fig. 5 shows an early prototype of a *DPP*. Thanks to the separation of concerns between the scheduling and execution of application and communication tasks, the *DPP* concept provides an efficient and predicable architecture for CPS nodes. By

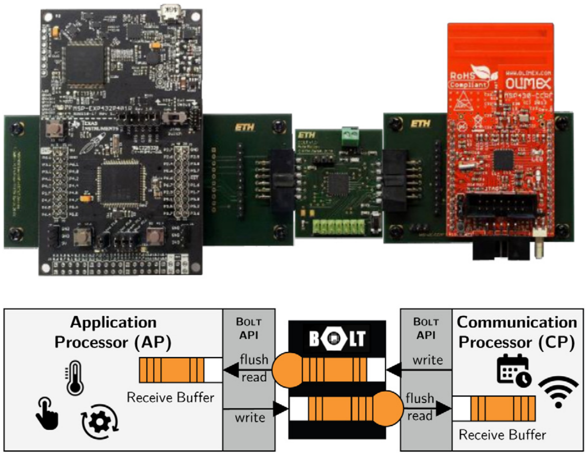


Fig. 5. Top.Example of a custom-built heterogeneous *DPP. Bolt (in the middle) interconnects a powerful application processor (TI MSP432* [42]*) on the left with a state-of-the-art communication processor (TI CC430* [43]*) on the right.* **Bottom.** Conceptual view of the *Bolt* processor interconnect. *Using the Bolt's API functions (*`write`*,* `read`*, and* `flush`*), the processors dedicated to application (AP)and communication (CP) can asynchronously exchange messages with predictable latency, while otherwise executing independently*.

entirely dedicating one processor to the application tasks and another one to wireless communication, we can decouple the timing of communication from the timing of the applications, and therefore facilitate the integration of ST in a CPS design. Furthermore, this helps to optimize performance: each processor can be customized for the specific operations it has to perform. The division of labor fosters specialization, thereby reducing the overall energy consumption and execution time; *i.e.*, maximizing the system's *Efficiency*.

1.5. *Performance evaluation in networking*

Over the past decade, low-power wireless communication has made significant progress, which are not limited to ST. The overall level of performance has increased, and it is now common to see reports of packet reception rates above 99% [23,44–46]. The more extreme the performance level, the more critical it becomes to confidently assess performance. Higher levels of confidence become necessary to argue about the differences in protocol design and quantify their performance trade-offs. Obviously, this is important for academics as it allow comparing competing approaches. But it is also important for industry: these new and promising technologies will never be adopted unless we can back up our performance claims confidently. In other words, others must be able to replicate our experiments.

In the context of wireless networking, replicable performance evaluation is made particularly challenging by the inherent variability of the experimental conditions: the uncontrollable dynamics of real-world networks [47,48] and the unsteady performance of hardware and software components [49,50] can cause a large variability in the experimental conditions, which makes it hard to quantitatively compare different solutions [51].

This reproducibility challenge (sometimes even referred to a "crisis" [52]) touches all scientific fields, and recently received significant attention in computer science [53–55]. Yet, how to practically design and execute performance evaluation experiments for wireless protocols remains a largely open question which is being debated by the community [56]. The lack of a standard for evaluating performance prevents a clear comparison of the different approaches, and therefore hinders the adoption of the technology. When everyone claims to be the best, one can hardly trust anyone.

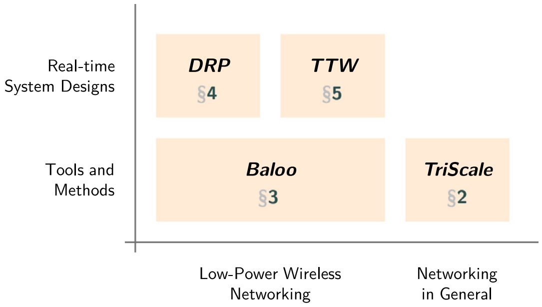


Fig. 6. Overview of the work presented in this paper.

1.6. *Recent advances*

In this paper, we summarize some of our recent work, in which we leverage recent advances in the domain of low-power wireless communication, in particular synchronous transmissions, in order to design wireless CPS providing end-to-end real-time guarantees. Fig. 6 provides a classification of the contributions, which we introduce below.

- We worked towards more rigorous and reproducible experimental networking research. In [57], we went beyond simple guidelines and proposed the first concrete methodology for designing networking experiments and analyzing their data. We leveraged this methodology to propose the first formalized definition of reproducibility for networking experiments. We implemented our methodology in *TriScale*, a first-of-its-kind tool that assists researchers by streamlining the design process and automating the data analysis (Section 2).
- We proposed and implemented *Baloo* [58], a design framework for network stacks based on synchronous transmissions (ST). *Baloo* significantly lowers the entry barrier for harnessing the efficiency, reliability and mobility support of ST: users implement their protocol through a simple yet flexible API while *Baloo* handles all the complex low-level operations based on the users' inputs (Section 3).
- We demonstrated for the first time that end-to-end real-time guarantees can be obtained in wireless CPS by leveraging the efficiency and reliability of synchronous transmissions. We proposed and implemented wireless real-time protocols for two different design objectives.
    - The Distributed Real-time Protocol (*DRP* [59]) uses contracts to maximize the flexibility of execution between application tasks (Section 4).
    - Time-Triggered Wireless (*TTW* [60]) statically co-schedules all task executions and message transfers to minimize end-to-end latency (Section 5).

The rest of this paper presents an overview of these contributions, highlights their underlying key ideas, then concludes with some directions for future work.

2. Designing replicable networking experiments with *TriScale*

The ability to replicate an experimental result is essential for making a scientifically sound claim. Without replicability[1] —the ability to

---

[1] Different terminology is used to refer to different aspects of replicability research [61,62]. In this paper, we refer to replicability as the ability of different researchers to follow the steps described in published work, collect new data using the same tools, and eventually obtain the same results, within the margins of experimental error. This is usually called replicability [63] but sometimes referred to as reproducibility.

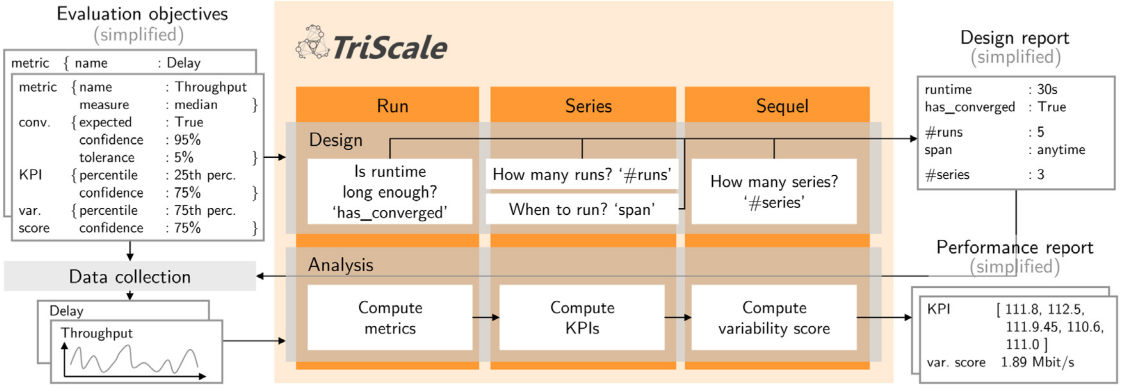


Fig. 7. Overview of *TriScale*. *TriScale is a framework supporting the design and analysis of networking experiments. TriScale assists the user in the design phase with a concrete methodology to answer important experiment design questions such as "How many runs?" and "How long should the runs be?" After the raw data are collected, TriScale supports the user by automating the data analysis. The framework implements robust statistics that handle the intrinsic variability of experimental networking data and returns expressive performance reports along with a variability score that quantifies the replicability of an experiment.*.

Table 1
A non-exhaustive list of factors hindering replicability, and selected networking references addressing them.

Focus of TriScale	
Variability in experiment design and data analysis	[54,55,64]
Other factors hindering replicability	
Lack of documentation	[55,65,66]
Artifacts unavailability/unusability	[53,67]
Uncontrollability of the exp. conditions	[47,48,68–70]
Variability of hardware and software behavior	[49,50,71]

assess the validity of claims reported by other researchers—any performance evaluation is questionable, at best. In networking, replicability is a well-recognized problem which stems from several factors (Table 1).

To be replicable, performance evaluations must account for the inherent variability of the experimental conditions—*i.e.*, the environment in which the experiment takes place—and the variability in hardware and software behavior in the system under test as well as in the measuring system. To facilitate this, the networking community has put great efforts into developing testbeds and data collection frameworks, *e.g.*, [68–70]. In addition, several calls for actions have been made to foster proper documentation [54,55] and artifact sharing [53,63] which are essential for replicability.

A more subtle but nonetheless important hindering factor for replicability are *differences in the methodology* used to design an experiment, analyze the resulting data, and draw conclusions from the evaluation. The literature related to this problem is currently limited to generic guidelines [54,55,72] and recommendations [64,66,73], which leave open several critical questions *before* (How many runs? How long should a run be?) and *after* experiments are conducted (How to process the data and analyze the results?). Without a concrete methodology, networking researchers often design and analyze similar experiments in different ways, making them hardly comparable [56]. Yet, strong claims are being made ("our system improves latency by 3×") while confidence is often discussed only in qualitative ways ("with high confidence"), if at all [70,71]. Furthermore, it is unclear how to effectively assess whether an experiment is indeed replicable. We argue that a concrete methodology is needed to help resolve this situation.

Hence, we developed such a methodology for the design and analysis of performance evaluations for networking research. In [57], we introduced *TriScale*, an implementation of our methodology into a software framework making the methodology readily applicable by researchers. While we do not claim that our methodology is fitting all situations, nor that it is the best one possible, we do find it useful in many practical cases. At a high level, our methodology features four key desirable properties.

Rationality The methodology rationalizes the experiment design by linking the design questions (*e.g.*, How many runs?) with the confidence in the performance claims.

Robustness The methodology is robust against the variability of the experimental conditions. The data analysis uses statistics that are compatible with the nature of networking data and are able to quantify the expected performance variation shall the evaluation be replicated.

Generality The methodology is applicable to a wide range of performance metrics, evaluation scenarios (emulator, testbed, in the wild), and network types (wired, wireless).

Conciseness The methodology describes the experimental design and the data analysis in a concise and unambiguous way to foster replicability while minimizing the use of highly treasured space in scientific papers.

Key idea. *TriScale*'s methodology is based on an analysis of the temporal characteristics of variability in networking experiments, which we argue can be decomposed into three timescales, which can be mapped to specific questions of the experiment design (see Fig. 7). For each timescale, *TriScale* applies a set of appropriate and rigorous statistical methods to derive performance results with *quantifiable confidence*. For each performance metric, *TriScale* computes a variability score that estimates, with a given confidence, how similar the results would be if the evaluation were replicated.

Limitations. With *TriScale*, we provide a concrete methodology that *concretely guides* networking researchers through the design of their experiments and the analysis of the gathered data, while *quantifying the replicability* of the performance evaluation. Hence, *TriScale* complements prior work toward replicable networking research that mostly focused on data collection, *e.g.*, [68–70].

Take-away. *TriScale* is implemented as a Python package [74]. For each timescale, a dedicated function takes raw data as input, performs the corresponding test or analysis, returns the results, and produces data visualizations. We aimed to make *TriScale* intuitive and easy to use. For a better impression of its usability, you can run an interactive demo directly in your web browser [75]. We expect *TriScale*'s open availability to actively encourage its use by the networking community and promote better experimentation practices in the short term.

The quest towards highly-reproducible networking experiments remains open, but we believe that *TriScale* represents an important stepping stone towards an accepted standard for experimental evaluations in networking.

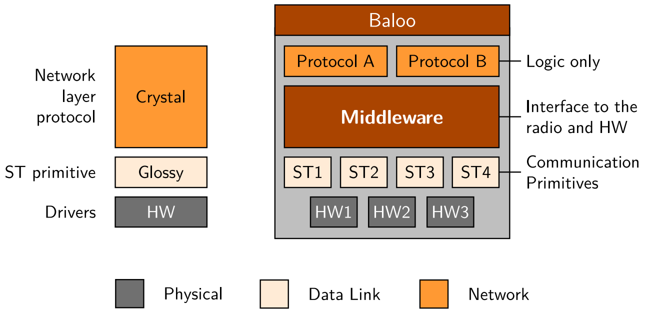
3. Synchronous transmissions made easy: Design your network stack with *Baloo*

As introduced in Section 1.3, Synchronous Transmissions (ST) is an increasingly used wireless communication technology for low-power multi-hop networks. Popularized by Glossy [23] in 2011, it has been proven to be highly reliable and energy efficient, as illustrated by the EWSN Dependability Competition [34], where all wining solutions were based on ST [27,29,35–37] (from 2016 to 2019).

A *ST primitive* refers to a protocol that efficiently realizes broadcast (*i.e.*, any-to-all communication) in bounded time, usually relying on *flooding*. Flooding is a communication strategy that realizes broadcast by having all receivers of a packet retransmit this same packet to all their neighbors; the packet is thus "flooded" through the whole network. ST makes flooding energy and time efficient by letting multiple wireless nodes transmit the packet *synchronously*, hence the name of *Synchronous Transmissions*. The successful reception of the packet can be achieved if the transmitters are tightly synchronized, thanks to *constructive interference* and the *capture effect* [20]. The synchronization requirements vary from sub- µs to tens of µs, depending on the platform and modulation scheme [20]. Such a broadcast primitive simplifies the design of network layer protocols: The underlying multi-hop network can be abstracted as a *virtual single-hop network* and thus be scheduled like a shared bus [30].

Since Glossy [23], many flavors of ST primitives have been proposed to improve performance in terms of reliability, latency, and energy consumption. To be more resilient to strong interference, Robust Flooding [27] is a primitive that modifies the RX-TX sequence from the original Glossy, whereas RedFixHop [76] uses hardware acknowledgments to minimize the number of retransmissions required. Instead, some primitives aim to minimize latency for specific traffic patterns. For example, Chaos [32] lets all nodes modify the packet being flooded to quickly aggregate information (*e.g.*, the max value of all sensor readings) or efficiently perform all-to-all data sharing to achieve distributed consensus [77]. Codecast [78] also targets many-to-many exchange for a larger amount of data. Pando [79] is another primitive focused on high throughput, which uses fountain code and packet pipelining for efficient data dissemination. Syncast [80] aims to reduce the radio on time required to save energy, while Less is More (LiM) [81] is a primitive that reduces energy consumption using learning to avoid unnecessary retransmissions during flooding.

All these primitives share the same drawback: Successful ST requires low-level control of timers and radio events in order to meet ST tight synchronization requirements (the order of µs). This degree of accuracy is difficult to achieve as it requires a detailed knowledge of the underlying hardware, low-level control of the radio operations, and a very careful management of software delays. As a result, designing a network stack based on ST is a complex and time-consuming task, for which only few solutions have been proposed. One of the first was the Low-power Wireless Bus (LWB) [30], which tries to flexibly support all kinds of traffic patterns in a balanced trade-off between latency and energy consumption. The same group designed eLWB [82], a variation of LWB tailored to event-based data collection. Sleeping Beauty [83] was later proposed to minimize energy consumption for data collection scenarios with many redundant sensor nodes. Time-Triggered-Wireless (*TTW* [60], Section 5) was designed to minimize the end-to-end latency between communicating application tasks. Finally, Crystal [46] has been proposed as a network stack specialized for sporadic data collection. All these network stacks solely rely on Glossy as ST primitive. In principle however, the same protocol logic could benefit from *multiple* primitives. For example, an LWB network could use Robust Flooding [27] in case of high interference, then revert to Glossy [23] for better time synchronization. If nodes need reprogramming, the software update can be quickly disseminated using Pando [79]. Designing a modular network stack supporting multiple ST primitives adds a new level of complexity.



(a) The implementation of the network layer protocol (Crystal) couples the interface to the underlying ST primitive (Glossy) and the protocol logic, *i.e.*, how long are the communication rounds, which radio channel is used, etc.

(b) Thanks to its additional middleware layer, *Baloo* flexibly supports multiple ST primitives and significantly reduces the efforts required to implement network layer protocols compared to traditional stacks, like LWB [34] or Crystal [46].

Fig. 8. Crystal [46] is a typical example of network stack based on ST (Fig. 8a). Conversely, *Baloo* is a flexible design framework. It is based on a middleware layer that separates the concern of timely execution of ST primitives from the implementation of the protocol logic (Fig. 8b).

Question 1 Can we facilitate the design of wireless network stacks based on Synchronous Transmission?

Question 2 Can we implement flexible and adaptive protocols, potentially leveraging multiple ST primitives, while guaranteeing that the timing requirements of ST are met?

The problem. To facilitate the network stack design (**Question 1**), a natural idea is to separate the concern of the timely execution of the primitives from the implementation of the protocol logic. One way to achieve such separation of concerns is to use a *middleware* as part of the network stack. The idea of a middleware for Wireless Sensor Networks (WSN) is not new, and the main challenge in such an endeavor is well-known. As phrased by Mottola and Picco [84], "*striking a balance between flexibility and complexity in providing access to low-level features is probably one of the toughest, yet most important, problems in WSN middleware*". The design of a middleware for ST is particularly challenging. Indeed, meeting the tight timing requirements for ST is directly conflicting with the concept of abstraction of a middleware: How to guarantee that the network layer does not hinder the timing accuracy for ST if it is itself unaware of the execution of the primitives? That is **Question 2**.

The challenge. A middleware for ST should meet the following requirements.

Usability The middleware must realize a well-defined interface enabling runtime control from the network layer (which implements the protocol logic) over the execution of the underlying ST primitives.

Generality The middleware must enable the implementation of a large variety of network layer protocols.

Versatility The middleware must enable one network layer protocol to use multiple ST primitives and switch between them at runtime.

Synchronicity The middleware must guarantee to respect the time synchronization requirements for ST (from sub- µs to tens of µs [20]).

Our solution. To address these challenges, we have designed *Baloo* [58],[2] a flexible design framework for low-power network stacks based on ST. *Baloo* provides a large set of features enabling performant protocol designs, while abstracting away low-level hardware management such as interrupt handling and radio core control. In summary:

- We proposed *Baloo*, a flexible design framework for low-power wireless network stacks based on ST (see Fig. 8).
- We presented the design of a middleware layer that meets all our requirements. This middleware forms the core component of *Baloo*.
- We showcased the usability of *Baloo* by re-implementing three well-known network stacks using ST: the Low-power Wireless Bus (LWB) [30], Sleeping Beauty [83], and Crystal [46].
- We illustrated the portability of *Baloo* by providing implementations for two platforms — the CC430 SoC [43] and the old but still heavily used TelosB mote [85].
- We demonstrated that *Baloo* induces only limited performance overhead (memory usage, radio duty cycle) compared to the original implementations.

Key idea. The core of *Baloo* is its clean API, based on callback functions, which let the users focus on implementing the protocol logic without worrying about low-level radio control (interrupt handling, timer settings, etc..). The API is generic and supports the different communication primitives. Through this API, multiple primitives can be used within the same network stack without additional complexity for the users.

Limitations. *Baloo* is a tool that facilitate the *implementation* of ST-based protocols, but it does not help to *design* them. The protocol design space is very large, with many trade-offs to consider depending on the application use case. This is still a fertile area of research, with recent proposals including [86–92].

Take-away. *Baloo* is openly available and is accompanied by a detailed documentation of its features and how to use them [93]. Our re-implementations of Crystal, Sleeping Beauty, and LWB are also available. We believe *Baloo* will be an important enabler for the development of real-world applications leveraging state-of-the-art ST technology.

4. *DRP*: Flexible real-time guarantees

As introduced in Section 1, Cyber-physical systems (CPS) tightly integrate components for sensing, actuating, and computing into distributed feedback loops to directly control physical processes [4]. As many CPS applications are mission-critical and physical processes evolve as a function of time, the communication among the sensing, actuating, and computing elements is often subject to real-time requirements, for example, to guarantee stability of the feedback loops [19]. These real-time requirements are often specified from an end-to-end application perspective. For example, a control engineer may require that sensor readings taken at time t are available for computing the control law at $t + D$, where the relative deadline D is derived from the application requirements; *e.g.*, the maximum tolerable delay between a sensing and a control task, where these tasks are typically executed on physically distributed devices.

Meeting end-to-end deadlines is non-trivial because data transfers between application tasks involves multiple other tasks (*e.g.*, operating system, networking protocols) and shared resources (*e.g.*, memories, system buses, wireless medium). The entire transmission chain of the data throughout the system must be taken into account to enable end-to-end real-time guarantees.

Question 3 Can we provide end-to-end real-time guarantees between distributed applications in wireless CPS?

Question 4 Can we do so while preserving runtime adaptability and flexibility in the timing of task executions?

The problem. Enabling real-time communication between network interfaces of sources and destinations in a low-power wireless network has been studied for more than a decade [94–96]. Today, standards such as WirelessHART [97] and ISA100.11a [98] for control applications in the process industries already exist [99], and considerable progress in real-time transmission scheduling and end-to-end delay analysis for WirelessHART networks has been made [100,101].

Unfortunately, wireless real-time protocols such as WirelessHART [97] or Blink [102] only provide guarantees for message transmissions between *network interfaces*. These protocols do not handle the application schedules; at the source, the message release is typically assumed periodic; at the destination, nothing guarantees that the application will process the message in time. Providing end-to-end guarantees between *distributed application* (**Question 3**) demands to combine a wireless real-time protocol with the rest of the system; *i.e.*, consider application schedules and handle interference on shared resources.

The challenge. To support a broad spectrum of CPS applications, a solution to this problem should fulfill the following requirements.

Timeliness All messages received by the destination application meet their end-to-end deadlines.

Reliability All messages received at the wireless network interface are successfully delivered to their destination application (*i.e.*, no buffer overflows).

Adaptability The system adapts to dynamic changes in traffic requirements at runtime.

Composability Existing hardware and software components can be freely composed to satisfy the application's needs, without altering the properties of the integrated parts.

Efficiency The solution scales to large systems and operates efficiently with respect to resources such as energy, wireless bandwidth, computing capacity, and memory.

The main challenge consists in funneling messages in real-time through tasks that run concurrently and access shared resources. Interference on such resources can delay tasks and communication arbitrarily, therefore hampering *Timeliness*, *Reliability*, and *Composability*.

Our solution. In [59], we presented *DRP*, a real-time wireless CPS that tackles interference on shared resources by defining (minimal) constraints on the application schedules. This is achieved by combining a predictable device architecture with a real-time scheduler for the entire system.

Predictable device architecture We use the Dual-Processor Platform (*DPP*) concept (Section 1.4). The *DPP* dedicates a communication processor (CP) exclusively to the real-time network protocol and executes all other tasks on an application processor (AP). The *DPP* is based on the *Bolt* interconnect [41], which decouples two processors in the time, power, and clock domains, while allowing them to asynchronously exchange messages within predictable time bounds.

Thus, on each device, we decouple the communication and application tasks, which can be independently invoked in an event- or time-triggered fashion. The *DPP* concept guarantees the faithfulness of the network interface (*Reliability*), supports *Composability*, and leverages the recent trend toward ultra low-power multi-processor architectures, which can be chosen individually to match the needs of the application and the networking protocol respectively (*Efficiency*).

---

[2] The framework provides the "bare necessities" for the design and implementation of ST-based network stacks; so we called it *Baloo*.

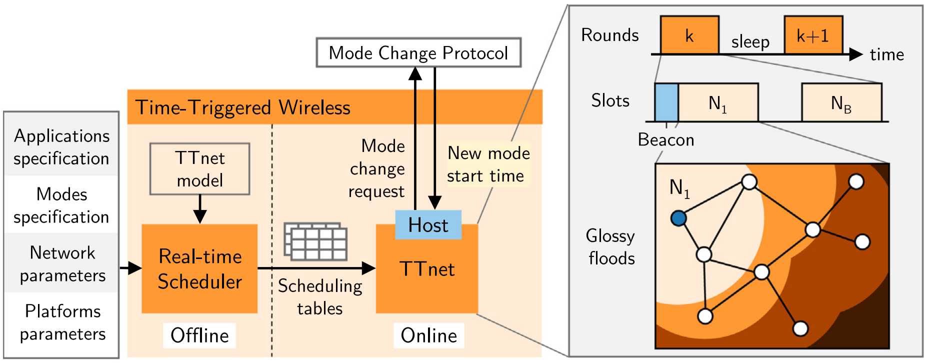


Fig. 9. High-level overview of the Time-Triggered Wireless (*TTW*) architecture.

Real-time scheduler We design the Distributed Real-time Protocol (*DRP*), a scheduler that provably guarantees that all messages received at the application interfaces meet their end-to-end deadlines (*Timeliness*) and that message buffers along the data transfers do not overflow (*Reliability*).

To accomplish this while being adaptive to unpredictable changes (*Adaptability*), *DRP* dynamically establishes at runtime a set of contracts based on the current traffic demands in the system. A contract determines the mutual obligations in terms of (i) minimum service provided, and (ii) maximum demand generated between the networking protocol and an application. *DRP* contracts define time bounds that can be analyzed to ensure that end-to-end deadlines are met, while preserving flexibility in the timing of distributed task executions (**Question 4**).

In summary, [59] presents the following contributions.

- We designed *DRP*, a wireless CPS system that provably provides end-to-end real-time guarantees between distributed applications. *DRP* does so by harnessing the benefits of synchronous transmissions (Section 1.3) and building upon the Blink real-time scheduler [102] and the Dual-Processor Platform architecture (Section 1.4).
- We simulated *DRP* execution to demonstrate that the provided bounds are both safe and tight, we implemented the protocol on embedded hardware, and showcased that it works as expected.
- We made our implementation of *DRP* publicly available [103], which includes the Blink scheduler for LWB [30].

Key idea. The key concept of *DRP* is to (i) physically decouple the communication protocol from the application tasks (each running on dedicated communication and application processors), and (ii) guarantee the timeliness of message transmissions throughout the system using minimally restrictive contracts between the different entities.

Take away. Our proof-of-concept implementation of *DRP* on embedded hardware confirmed that *DRP* appears to be a promising solution for low-rate applications, such as smart homes, where coexists multiple context-specific "applications" (*e.g.*, fridge, air-conditioning, lightning) which would particularly benefit from being scheduled independently of each other while being able to communicate in real-time.

Limitations. By design, enabling timing asynchrony between the connected applications leads to long end-to-end delays. Therefore, *DRP* is ill-suited for latency-sensitive applications, for which we designed a different system, presented in the next section.

5. *TTW*: Low-latency real-time guarantees

We revisited the challenge addressed in the previous section: Providing end-to-end real-time guarantees in wireless cyber–physical systems (CPS). With the design of *DRP* (Section 4), we demonstrated that, by leveraging synchronous transmissions (ST), it is possible to meet end-to-end deadlines between distributed tasks communicating through a multi-hop wireless network. *DRP* keeps all tasks as independent as possible; *i.e.*, constraining their schedule as little as necessary to provide end-to-end guarantees.

Because of that maximal-flexibility principle, the guarantees that can be provided by *DRP* are rather "slow": the minimal end-to-end deadline supported by the protocol is more than two times as large as a communication round. Furthermore, there is large jitter between successive task executions and message transmissions. This does not comply well with the requirements of industrial CPS applications, which often require short delays (the order of ms) and benefit from negligible jitter.

Thus, we considered another design objective: Instead of focusing on flexibility, we aim for minimizing latency and jitter in the system execution.

Question 3 Can we provide end-to-end real-time guarantees between distributed applications in wireless CPS?

Question 5 How can we minimize latency and jitter in the application execution while retaining some level of runtime adaptability?

The problem. To understand the challenges of wireless CPS, it is helpful to highlight the fundamental difference between a field bus and a wireless network. In a field bus, whenever a node is not transmitting, it can idly listen for incoming messages. Upon request from a central host, each node can wake up and react quickly. For a low-power wireless node, the major part of the energy is consumed by its radio. Therefore, energy efficiency requires to turn the radio off whenever possible to support long autonomous operation without an external power source. Since nodes are unreachable until they wake up, they require overlapping wake-up time intervals to communicate.

This observation often results in wireless system designs that minimize energy consumption by using communication rounds, *i.e.*, time intervals where all nodes wake-up, exchange messages, then turn off their radio [30,58,97,99]. Scheduling policies define when the rounds take place (*i.e.*, when to wake up) and which nodes are allowed to send messages during the round. Moreover, CPS do not only exchange messages, they also execute tasks (*e.g.*, sensing or actuation). Typically, the system requirements are specified end-to-end, *i.e.*, between distributed tasks exchanging messages. One option to meet such end-to-end requirements (**Question 3**) is to co-schedule the execution of tasks and the transmission of messages, as proposed in the literature for wired architectures [104–106]. However, these schedules result from complex optimization problems which are difficult to solve online, even more so in a low-power setting. Thus, schedules are often pre-computed offline, which restricts the runtime adaptability of the resulting system (**Question 5**).

The challenge. To support wireless CPS applications in an industrial context, a solution to this problem should fulfill the following requirements.

Timeliness All distributed applications meet their end-to-end deadlines.

Reliability A large ratio of messages is successfully transmitted over wireless and conflict-free communication is guaranteed between the system's nodes.

Adaptability The system adapts to runtime changes.

Mobility The system supports mobile devices.

Efficiency The system supports short end-to-end latency (ms), scales to medium-to-large system sizes, and optimizes its energy consumption and bandwidth utilization.

Our solution. In [60], we proposed *TTW*, a solution to the industrial wireless CPS problem that fulfill these requirements. We do so by combining co-scheduling techniques, inspired from the wired literature, with a ST-based wireless system design using communication rounds.

ST provides highly reliable wireless communication (*Reliability*) and inherent support for *Mobility*. A round-based design allows to minimize the energy consumed for communication, which is a large part of the total energy budget of a low-power system (*Efficiency*). The co-scheduling approach results in highly optimized schedules (*Efficiency*) which guarantee to meet the application deadlines (*Timeliness*). *TTW* provides some runtime *Adaptability* by switching between multiple pre-computed operation modes, a well-known concept in the wired literature [107].

The main challenge in realizing such a system is to integrate the allocation of messages to communication rounds (which is similar to a bin-packing problem [108]) with a co-scheduling approach (which typically solves a MILP [109] or an SMT [110–112] formulation). In summary, [60] presents the following contributions.

- We presented Time-Triggered Wireless (*TTW*, illustrated in Fig. 9), a low-power wireless CPS that meets the common requirements of industrial applications.
- We formulated a joint optimization problem for co-scheduling distributed tasks, messages, and communication rounds that guarantees to meet application deadlines, minimize the energy consumed for wireless communication, and ensures safety in terms of conflict-free communication, even under packet loss.
- We provided a methodology that efficiently solves this optimization problem, known to be NP-hard [113].
- Using time and energy models, we quantified the benefits of rounds to minimize energy, and we derive the minimum end-to-end latency achievable.
- We implemented *TTW* on embedded hardware and demonstrate that the system is suited for fast feedback control applications.

Key idea. The main challenge in the *TTW* design is that, with wireless communication, it is highly beneficial in terms of energy to send messages in rounds. Thus, the assignment of messages to round (similar to a bin-packing problem) must be combined to the traditional co-scheduling approaches, which is non-trivial. We solved this problem and implemented a multi-mode scheduler that allows critical applications to seamlessly switch between modes while minimizing the energy consumption spent for wireless communication. We further implemented a predictable network stack, called *TTnet*. Together, these two pieces form *TTW*, a publicly available [114] real-time wireless CPS design.

Limitations. Efficient static scheduling requires a precise model of the worst-case execution time of the tasks, which may be challenging to get in practice. Moreover, the synthesis of scheduling tables is computationally expensive, and does not scale well to large applications (hundreds of tasks and messages per application's period).

Take-away. *TTW* achieves near-optimal end-to-end latency by a tight coupling in the timing of task executions and message exchanges. Compared to *DRP* (Section 4), *TTW*'s static schedules allow meeting shorter end-to-end deadlines (*Efficiency*) at the cost of a lesser *Adaptability*; indeed, *TTW*'s runtime adaptability is limited to switching between predefined operation modes.

6. Open issues and future work

Benchmarking Wireless Protocols. Our work on *TriScale* stemmed from discussions in the low-power wireless community regarding the need for a benchmark to compare networking protocols [56]. As we were reflecting on how to design such a benchmark, the need for a more rigorous experimental methodology became obvious; a need that *TriScale* tries to fill. We can now return to our initial objectives and attempt to realize the vision of IoTBench: a benchmark to thoroughly and confidently compare the performance of wireless networking protocols [115].

Going further with ST. With the design of *Baloo*, we attempted to make ST more accessible; an attempt that appears to be successful. Less than a year after the initial paper, the first independent studies using *Baloo* have been published [116]. There are many opportunities for future developments of the framework; the most natural being the port of *Baloo* to other platforms. It has been shown that the principle of ST also work on other physical layers that IEEE802.15.4 (*e.g.*, Bluetooth [25] and LoRa [26]). To investigate this further, a port to the LoRa-compatible SemTech SX1262 chip [117] is currently under development. A port to the Bluetooth-compatible nRF52840 Dongle [118] has just been recently released [119]. These would allow to experiment with ST-based networking on different physical layers and, by leveraging (hopefully upcoming) wireless protocol benchmarks, we would be able to objectively compare the performance trade-offs of these different technologies in a wide range of scenarios and applications.

On the application side, researchers are starting to harness the benefits of ST for wireless CPS, following the tracks opened by *TTW*—see *e.g.*, [92,120].

Dependable networking. One important limitation of the system designs presented in this dissertation is the reliance on a central authority, which we call *host*, in order to coordinate communication within the network. This creates a single point of failure: if the host should fail (or be jammed), the entire network would stop its operation. For any safety-critical applications, this is not acceptable. It is therefore important to work on system designs that would "distribute the responsibility" of the host. Recent contributions provide consensus primitives in low-power networks [25,116,121], an important piece for fault-tolerance in distributed systems. However, these works rely on a central authority for key network functions, such as time synchronization. More efforts are required to designed truly dependable wireless networks.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] NSF, Cyber Physical Systems – Nsf10515, NSF, 2010, URL https://www.nsf.gov/pubs/2010/nsf10515/nsf10515.htm.

[2] J.A. Stankovic, When sensor and actuator networks cover the world, ETRI J. 30 (5) (2008) 627–633, http://dx.doi.org/10.4218/etrij.08.1308.0099, URL https://onlinelibrary.wiley.com/doi/abs/10.4218/etrij.08.1308.0099.

[3] E.A. Lee, Cyber physical systems: design challenges, in: 2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing, ISORC, 2008, pp. 363–369, http://dx.doi.org/10.1109/ISORC.2008.25.

[4] R. Rajkumar, I. Lee, L. Sha, J. Stankovic, Cyber-physical systems: the next computing revolution, in: Design Automation Conference, 2010, pp. 731–736, http://dx.doi.org/10.1145/1837274.1837461.

[5] M. Luvisotto, Z. Pang, D. Dzung, Ultra high performance wireless control for critical applications: challenges and directions, IEEE Trans. Ind. Inf. 13 (3) (2017) 1448–1459, http://dx.doi.org/10.1109/TII.2016.2617459.

[6] M. Haegele, Logistics drives 39% increase in professional service robot sales, IFR Int. Fed. Robot. (2018) URL https://ifr.org/post/logistics-drives-39-increase-in-professional-service-robot-sales.

[7] L. Mottola, M. Moretta, K. Whitehouse, C. Ghezzi, Team-level programming of drone sensor networks, in: Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems, in: SenSys '14, ACM, New York, NY, USA, 2014, pp. 177–190, http://dx.doi.org/10.1145/2668332.2668353, URL http://doi.acm.org/10.1145/2668332.2668353.

[8] J. Åkerberg, M. Gidlund, M. Björkman, Future research challenges in wireless sensor and actuator networks targeting industrial automation, in: 2011 9th IEEE International Conference on Industrial Informatics, 2011, pp. 410–415, http://dx.doi.org/10.1109/INDIN.2011.6034912.

[9] G.C. Buttazzo, Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications, Springer Science & Business Media, 2011.

[10] K. Geissdoerfer, B. Kusy, R. Jurdak, M. Zimmerling, Getting more out of energy-harvesting systems: energy management under time-varying utility with preact, in: 2019 18th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN), 2019, pp. 109–120, http://dx.doi.org/10.1145/3302506.3310393.

[11] S. Weber, J. Beutel, R.D. Forno, A. Geiger, S. Gruber, T. Gsell, A. Hasler, M. Keller, R. Lim, P. Limpach, M. Meyer, I. Talzi, L. Thiele, C. Tschudin, A. Vieli, D. Vonder Mühll, M. Yücel, A decade of detailed observations (2008–2018) in steep bedrock permafrost at the matterhorn hörnligrat (zermatt, CH), Earth Syst. Sci. Data (2019) http://dx.doi.org/10.5194/essd-11-1203-2019, URL https://www.earth-syst-sci-data.net/11/1203/2019/.

[12] K. Chintalapudi, T. Fu, J. Paek, N. Kothari, S. Rangwala, J. Caffrey, R. Govindan, E. Johnson, S. Masri, Monitoring civil structures with a wireless sensor network, IEEE Internet Comput. 10 (2) (2006) 26–34, http://dx.doi.org/10.1109/MIC.2006.38.

[13] B. Cassens, M. Hartmann, T. Nowak, N. Duda, J. Thielecke, A. Kölpin, R. Kapitza, Bursting: increasing energy efficiency of erasure-coded data in animal-Borne sensor networks, in: Proceedings of the 2019 International Conference on Embedded Wireless Systems and Networks, in: EWSN '19, Junction Publishing, USA, 2019, pp. 59–70, URL http://dl.acm.org/citation.cfm?id=3324320.3324328.

[14] P. Zhang, C.M. Sadler, S.A. Lyon, M. Martonosi, Hardware design experiences in ZebraNet, in: Proceedings of the 2Nd International Conference on Embedded Networked Sensor Systems, in: SenSys '04, ACM, New York, NY, USA, 2004, pp. 227–238, http://dx.doi.org/10.1145/1031495.1031522, URL http://doi.acm.org/10.1145/1031495.1031522.

[15] Wikipedia, Chinese whispers, 2019, Wikipedia URL https://en.wikipedia.org/w/index.php?title=Chinese_whispers&oldid=920368233.

[16] T. Watteyne, V. Handziski, X. Vilajosana, S. Duquennoy, O. Hahm, E. Baccelli, A. Wolisz, Industrial wireless IP-based cyber-physical systems, Proc. IEEE 104 (5) (2016) 1025–1038, http://dx.doi.org/10.1109/JPROC.2015.2509186.

[17] H.-S. Kim, J. Ko, D.E. Culler, J. Paek, Challenging the IPv6 routing protocol for low-power and lossy networks (RPL): A survey, IEEE Commun. Surv. Tutor. 19 (4) (Fourthquarter 2017) 2502–2525, http://dx.doi.org/10.1109/COMST.2017.2751617.

[18] L. Mottola, G.P. Picco, MUSTER: adaptive energy-aware multisink routing in wireless sensor networks, IEEE Trans. Mob. Comput. 10 (12) (2011) 1694–1709, http://dx.doi.org/10.1109/TMC.2010.250.

[19] J. Stankovic, I. Lee, A. Mok, R. Rajkumar, Opportunities and Obligations for Physical Computing Systems, 2005, URL https://repository.upenn.edu/cis_papers/222.

[20] D. Yuan, M. Hollick, Let's talk together: understanding concurrent transmission in wireless sensor networks, in: 38th Annual IEEE Conference on Local Computer Networks, 2013, pp. 219–227, http://dx.doi.org/10.1109/LCN.2013.6761237.

[21] A. Escobar-Molero, Improving reliability and latency of wireless sensor networks using concurrent transmissions, At - Automatisierungstechnik 67 (1) (2019) 42–50, http://dx.doi.org/10.1515/auto-2018-0064, URL https://www.degruyter.com/view/j/auto.2019.67.issue-1/auto-2018-0064/auto-2018-0064.xml.

[22] M. Wilhelm, V. Lenders, J.B. Schmitt, On the reception of concurrent transmissions in wireless sensor networks, IEEE Trans. Wireless Commun. 13 (12) (2014) 6756–6767, http://dx.doi.org/10.1109/TWC.2014.2349896.

[23] F. Ferrari, M. Zimmerling, L. Thiele, O. Saukh, Efficient network flooding and time synchronization with glossy, in: Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks, 2011, pp. 73–84, URL https://ieeexplore.ieee.org/document/5779066.

[24] D. Yuan, M. Riecker, M. Hollick, Making 'glossy' networks sparkle: exploiting concurrent transmissions for energy efficient, reliable, ultra-low latency communication in wireless control networks, in: Proceedings of the 11th European Conference on Wireless Sensor Networks - Vol. 8354, in: EWSN 2014, Springer-Verlag New York, Inc., New York, NY, USA, 2014, pp. 133–149, http://dx.doi.org/10.1007/978-3-319-04651-8_9.

[25] B. Al Nahas, S. Duquennoy, O. Landsiedel, Concurrent transmissions for multi-hop bluetooth 5, in: Proceedings of the 2019 International Conference on Embedded Wireless Systems and Networks, in: EWSN '19, Junction Publishing, USA, 2019, pp. 130–141, URL http://dl.acm.org/citation.cfm?id=3324320.3324336.

[26] M. Wegmann, Reliable 3rd Generation Data Collection, ETH Zurich, 2018, URL https://pub.tik.ee.ethz.ch/students/2018-FS/MA-2018-08.pdf.

[27] R. Lim, R. Da Forno, F. Sutton, L. Thiele, Competition: robust flooding using back-to-back synchronous transmissions with channel-hopping, in: Proceedings of the 2017 International Conference on Embedded Wireless Systems and Networks, in: EWSN '17, Junction Publishing, USA, 2017, pp. 270–271, URL http://dl.acm.org/citation.cfm?id=3108009.3108076.

[28] M. Baddeley, U. Raza, A. Stanoev, G. Oikonomou, R. Nejabati, M. Sooriyabandara, D. Simeonidou, Atomic-SDN: is synchronous flooding the solution to software-defined networking in IoT? IEEE Access 7 (2019) 96019–96034, http://dx.doi.org/10.1109/ACCESS.2019.2920100.

[29] X. Ma, P. Zhang, Y. Liu, X. Li, W. Tang, P. Tian, J. Wei, L. Shu, O. Theel, Competition: using decot+ to collect data under interference, in: Proceedings of the 2019 International Conference on Embedded Wireless Systems and Networks, 2019, p. 2, URL http://dl.acm.org/citation.cfm?id=3324320.3324385.

[30] F. Ferrari, M. Zimmerling, L. Mottola, L. Thiele, Low-power wireless bus, in: Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems, in: SenSys '12, ACM, New York, NY, USA, 2012, pp. 1–14, http://dx.doi.org/10.1145/2426656.2426658, URL http://doi.acm.org/10.1145/2426656.2426658.

[31] M. Zimmerling, L. Mottola, S. Santini, Synchronous transmissions in low-power wireless: a survey of communication protocols and network services, ACM Comput. Surv. 53 (6) (2020) 121:1–121:39, http://dx.doi.org/10.1145/3410159.

[32] O. Landsiedel, F. Ferrari, M. Zimmerling, Chaos: versatile and efficient all-to-all data sharing and in-network processing at scale, in: Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems, in: SenSys '13, ACM, New York, NY, USA, 2013, pp. 1:1–1:14, http://dx.doi.org/10.1145/2517351.2517358, URL http://doi.acm.org/10.1145/2517351.2517358.

[33] C. Herrmann, F. Mager, M. Zimmerling, Mixer: efficient many-to-all broadcast in dynamic wireless mesh networks, in: Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems, in: SenSys '18, ACM, New York, NY, USA, 2018, pp. 145–158, http://dx.doi.org/10.1145/3274783.3274849, URL http://doi.acm.org/10.1145/3274783.3274849.

[34] M. Schuß, C.A. Boano, M. Weber, K. Römer, A competition to push the dependability of low-power wireless protocols to the edge, in: Proceedings of the 14th International Conference on Embedded Wireless Systems and Networks, in: EWSN'17, Junction Publishing, USA, 2017, pp. 54–65, http://dx.doi.org/10.5555/3108009.3108018.

[35] A. Escobar, F. Moreno, A.J. Cabrera, J. Garcia-Jimenez, F.J. Cruz, U. Ruiz, J. Klaue, A. Corona, D. Tati, T. Meyerhoff, Competition: BigBangBus, in: Proceedings of the 2018 International Conference on Embedded Wireless Systems and Networks, in: EWSN '18, Junction Publishing, USA, 2018, pp. 213–214, URL http://dl.acm.org/citation.cfm?id=3234847.3234894.

[36] P. Sommer, Y.-A. Pignolet, Competition: dependable network flooding using glossy with channel-hopping, in: Proceedings of the 2016 International Conference on Embedded Wireless Systems and Networks, 2016, URL http://dl.acm.org/citation.cfm?id=2893711.2893785.

[37] A. Escobar-Molero, J. Garcia-Jimenez, J. Klaue, F. Moreno-Cruz, B. Saez, F.J. Cruz, U. Ruiz, A. Corona, Competition: RedNodeBus, stretching out the preamble, in: Proceedings of the 2019 International Conference on Embedded Wireless Systems and Networks, in: EWSN '19, Junction Publishing, USA, 2019, pp. 304–305, URL http://dl.acm.org/citation.cfm?id=3324320.3324391.

[38] NXP, LPC541XX, URL https://www.nxp.com/products/processors-and-microcontrollers/arm-microcontrollers/general-purpose-mcus/lpc54000-cortex-m4-/low-power-microcontrollers-mcus-based-on-arm-cortex-m4-cores-with-optional-cortex-m0-plus-co-processor:LPC541XX.

[39] NXP, VF3xxR URL https://www.nxp.com/products/processors-and-microcontrollers/legacy-mcu-mpus/vfxxx-controller/r-series/32-bit-devices-for-advanced-connected-radio-entry-level-infotainment-and-digital-instrument-cluster-applications.:VF3xxR.

[40] J. Beutel, R. Trüb, R.D. Forno, M. Wegmann, T. Gsell, R. Jacob, M. Keller, F. Sutton, L. Thiele, The dual processor platform architecture: demo abstract, in: Proceedings of the 18th International Conference on Information Processing in Sensor Networks, in: IPSN '19, ACM, New York, NY, USA, 2019, pp. 335–336, http://dx.doi.org/10.1145/3302506.3312481, URL http://doi.acm.org/10.1145/3302506.3312481.

[41] F. Sutton, M. Zimmerling, R. Da Forno, R. Lim, T. Gsell, G. Giannopoulou, F. Ferrari, J. Beutel, L. Thiele, Bolt: a stateful processor interconnect, in: Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems, in: SenSys '15, ACM, New York, NY, USA, 2015, pp. 267–280, http://dx.doi.org/10.1145/2809695.2809706, URL http://doi.acm.org/10.1145/2809695.2809706.

[42] Texas Instruments, MSP-EXP432P401R SimpleLink™, URL http://www.ti.com/tool/MSP-EXP432P401R.

[43] Texas Instruments, CC430F6137 16-bit ultra-low-power MCU, 2019, [Online] URL http://www.ti.com/product/CC430F6137. (Accessed 11 January 2019).

[44] S. Duquennoy, B. Al Nahas, O. Landsiedel, T. Watteyne, Orchestra: robust mesh networks through autonomously scheduled TSCH, in: Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems, in: SenSys '15, ACM, New York, NY, USA, 2015, pp. 337–350, http://dx.doi.org/10.1145/2809695.2809714, URL http://doi.acm.org/10.1145/2809695.2809714.

[45] S. Duquennoy, J. Eriksson, T. Voigt, Five-nines reliable downward routing in RPL, 2017, Cs arXiv:1710.02324.

[46] T. Istomin, M. Trobinger, A.L. Murphy, G.P. Picco, Interference-resilient ultra-low power aperiodic data collection, in: Proceedings of the 17th ACM/IEEE International Conference on Information Processing in Sensor Networks, in: IPSN '18, IEEE Press, Piscataway, NJ, USA, 2018, pp. 84–95, http://dx.doi.org/10.1109/IPSN.2018.00015.

[47] M. Matos, Towards reproducible evaluation of large-scale distributed systems, in: Proceedings of the International Workshop on Advanced Tools, Programming Languages, and Platforms for Implementing and Evaluating Algorithms for Distributed Systems, ApPLIED, ACM, Egham, United Kingdom, 2018, pp. 5–7, http://dx.doi.org/10.1145/3231104.3231113, URL http://doi.acm.org/10.1145/3231104.3231113.

[48] R. Burchfield, E. Nourbakhsh, J. Dix, K. Sahu, S. Venkatesan, R. Prakash, RF in the jungle: effect of environment assumptions on wireless experiment repeatability, in: Proceedings of the International Conference on Communications, (ICC), IEEE, 2009, pp. 1–6, URL https://ehsaan.net/wp-content/uploads/publications/rfij.pdf.

[49] A. Maricq, D. Duplyakin, I. Jimenez, C. Maltzahn, R. Stutsman, R. Ricci, Taming performance variability, in: Proceedings of the 13th International USENIX Symposium on Operating Systems Design and Implementation, OSDI, USENIX Association, Carlsbad, CA, USA, 2018, pp. 409–425, URL https://www.usenix.org/system/files/osdi18-maricq.pdf.

[50] S.M. Blackburn, A. Diwan, M. Hauswirth, P.F. Sweeney, J.N. Amaral, T. Brecht, L. Bulej, C. Click, L. Eeckhout, S. Fischmeister, D. Frampton, L.J. Hendren, M. Hind, A.L. Hosking, R.E. Jones, T. Kalibera, N. Keynes, N. Nystrom, A. Zeller, The truth, the whole truth, and nothing but the truth: a pragmatic guide to assessing empirical evaluations, ACM Trans. Program. Lang. Syst. 38 (4) (2016) 15:1–15:20, http://dx.doi.org/10.1145/2983574.

[51] V. Bajpai, O. Bonaventure, K. Claffy, D. Karrenberg, Encouraging reproducibility in scientific research of the internet (dagstuhl seminar 18412), in: V. Bajpai, O. Bonaventure, K. Claffy, D. Karrenberg (Eds.), Dagstuhl Rep. 8 (10) (2019) 41–62, http://dx.doi.org/10.4230/DagRep.8.10.41, URL http://drops.dagstuhl.de/opus/volltexte/2019/10347.

[52] M. Baker, Is there a reproducibility crisis? Nat. News 533 (7604) (2016) 452–454, URL https://www.nature.com/news/polopoly_fs/1.19970!/menu/main/topColumns/topLeftColumn/pdf/533452a.pdf.

[53] C. Collberg, T. Proebsting, A.M. Warren, Repeatability and Benefaction in Computer Systems Research, Technical Report TR 14–04, University of Arizona, 2015, URL http://reproducibility.cs.arizona.edu/v2/RepeatabilityTR.pdf.

[54] D. Saucez, L. Iannone, Thoughts and recommendations from the ACM SIGCOMM 2017 reproducibility workshop, SIGCOMM Comput. Commun. Rev. 48 (1) (2018) 70–74, http://dx.doi.org/10.1145/3211852.3211863, URL https://ccronline.sigcomm.org/wp-content/uploads/2017/11/sigcomm-ccr-paper149.pdf.

[55] V. Bajpai, A. Brunstrom, A. Feldmann, W. Kellerer, A. Pras, H. Schulzrinne, G. Smaragdakis, M. Wählisch, K. Wehrle, The dagstuhl beginners guide to reproducibility for experimental networking research, SIGCOMM Comput. Commun. Rev. 49 (1) (2019) 24–30, URL https://dl.acm.org/citation.cfm?id=3314217.

[56] C.A. Boano, S. Duquennoy, A. Förster, O. Gnawali, R. Jacob, H.-S. Kim, O. Landsiedel, R. Marfievici, L. Mottola, G.P. Picco, X. Vilajosana, T. Watteyne, M. Zimmerling, IoTBench: towards a benchmark for low-power wireless networking, in: Proceedings of the 1st Workshop on Benchmarking Cyber-Physical Networks and Systems (CPSBench 2018), 2018, http://dx.doi.org/10.3929/ethz-b-000256517, URL https://www.research-collection.ethz.ch/handle/20.500.11850/256517.

[57] R. Jacob, M. Zimmerling, C.A. Boano, L. Vanbever, L. Thiele, Designing replicable networking experiments with triscale, J. Syst. Res. 1 (1) (2021) http://dx.doi.org/10.5070/SR31155408, URL https://escholarship.org/uc/item/63n4s9w2.

[58] R. Jacob, J. Bächli, R. Da Forno, L. Thiele, Synchronous transmissions made easy: design your network stack with baloo, in: Proceedings of the 2019 International Conference on Embedded Wireless Systems and Networks, 2019, http://dx.doi.org/10.3929/ethz-b-000324254.

[59] R. Jacob, M. Zimmerling, P. Huang, J. Beutel, L. Thiele, End-to-end real-time guarantees in wireless cyber-physical systems, in: 2016 IEEE Real-Time Systems Symposium, RTSS, 2016, pp. 167–178, http://dx.doi.org/10.1109/RTSS.2016.025, URL https://www.research-collection.ethz.ch/handle/20.500.11850/221542.

[60] R. Jacob, L. Zhang, M. Zimmerling, J. Beutel, S. Chakraborty, L. Thiele, The time-triggered wireless architecture, in: M. Völp (Ed.), 32nd Euromicro Conference on Real-Time Systems, ECRTS 2020, in: Leibniz International Proceedings in Informatics (LIPIcs), 165, Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 2020, pp. 19:1–19:25, http://dx.doi.org/10.4230/LIPIcs.ECRTS.2020.19.

[61] H.E. Plesser, Reproducibility vs. replicability: a brief history of a confused terminology, Front. Neuroinform. 11 (76) (2018) 1–4.

[62] L.A. Barba, Terminologies for reproducible research, 2018, arXiv:1802.03311 [Cs] arXiv:1802.03311.

[63] ACM, Artifact review and badging, 2020, URL https://www.acm.org/publications/policies/artifact-review-badging.

[64] V. Paxson, Strategies for sound internet measurement, in: Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement, in: IMC '04, Association for Computing Machinery, Taormina, Sicily, Italy, 2004, pp. 263–271, http://dx.doi.org/10.1145/1028788.1028824.

[65] J. Vitek, T. Kalibera, Repeatability, reproducibility and rigor in systems research, in: Proceedings of the 9th International Conference on Embedded Software (EMSOFT), ACM, 2011, pp. 33–38, URL https://www.cs.kent.ac.uk/pubs/2011/3174/content.pdf.

[66] K. Kritsis, G.Z. Papadopoulos, A. Gallais, P. Chatzimisios, F. Théoleyre, A tutorial on performance evaluation and validation methodology for low-power and lossy networks, IEEE Commun. Surv. Tutor. (2018) 1, http://dx.doi.org/10.1109/COMST.2018.2820810, URL https://icube-publis.unistra.fr/docs/13111/IEEECOMST-Perf-LLN.pdf.

[67] M. Flittner, M.N. Mahfoudi, D. Saucez, M. Wählisch, L. Iannone, V. Bajpai, A. Afanasyev, A survey on artifacts from CoNEXT, ICN, IMC, and SIGCOMM conferences in 2017, SIGCOMM Comput. Commun. Rev. 48 (1) (2018) 75–80, http://dx.doi.org/10.1145/3211852.3211864, URL http://doi.acm.org/10.1145/3211852.3211864.

[68] L. Nussbaum, Testbeds support for reproducible research, in: Proceedings of the International ACM SIGCOMM Reproducibility Workshop, in: Reproducibility'17, ACM, Los Angeles, CA, USA, 2017, pp. 24–26, http://dx.doi.org/10.1145/3097766.3097773, URL https://hal.inria.fr/hal-01577849/document.

[69] F.Y. Yan, J. Ma, G.D. Hill, D. Raghavan, R.S. Wahby, P. Levis, K. Winstein, Pantheon: the training ground for internet congestion-control research, in: Proceedings of the International USENIX Annual Technical Conference, ATC, USENIX Association, Boston, MA, USA, 2018, pp. 731–743, URL https://www.usenix.org/conference/atc18/presentation/yan-francis.

[70] F.Y. Yan, H. Ayers, C. Zhu, S. Fouladi, J. Hong, K. Zhang, P. Levis, K. Winstein, Learning in situ: A randomized experiment in video streaming, in: 17th USENIX Symposium on Networked Systems Design and Implementation, NSDI 20, 2020, pp. 495–511, URL https://www.usenix.org/conference/nsdi20/presentation/yan.

[71] A. Uta, A. Custura, D. Duplyakin, I. Jimenez, J. Rellermeyer, C. Maltzahn, R. Ricci, A. Iosup, Is big data performance reproducible in modern cloud networks? in: Proceedings of the 17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20), 2020, pp. 513–527, URL https://www.usenix.org/conference/nsdi20/presentation/uta.

[72] M. Focus, Seven Ways to Fail, Technical Report Brochure on Application Development, Test, and Delivery, Micro Focus, 2018, URL https://www.microfocus.com/media/brochure/seven_ways_to_fail_brochure.pdf.

[73] R. Jacob, C.A. Boano, U. Raza, M. Zimmerling, L. Thiele, Towards a methodology for experimental evaluation in low-power wireless networking, in: Proceedings of the 2nd Workshop on Benchmarking Cyber-Physical Systems and Internet of Things, CPS-IoTBench'19, 2019, http://dx.doi.org/10.3929/ethz-b-000325096, URL https://www.research-collection.ethz.ch/handle/20.500.11850/325096.

[74] R. Jacob, TriScale, 2021, URL https://github.com/romain-jacob/triscale.

[75] R. Jacob, TriScale Demo, 2021, URL http://triscale.ethz.ch.

[76] A. Escobar, F.J. Cruz, J. Garcia-Jimenez, J. Klaue, A. Corona, RedFixHop with channel hopping: reliable ultra-low-latency network flooding, in: 2016 Conference on Design of Circuits and Integrated Systems, (DCIS), 2016, pp. 1–4, http://dx.doi.org/10.1109/DCIS.2016.7845367.

[77] B. Al Nahas, S. Duquennoy, O. Landsiedel, Network-wide consensus utilizing the capture effect in low-power wireless networks, in: Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems, in: SenSys '17, ACM, New York, NY, USA, 2017, pp. 1:1–1:14, http://dx.doi.org/10.1145/3131672.3131685, URL http://doi.acm.org/10.1145/3131672.3131685.

[78] M. Mohammad, M.C. Chan, Codecast: supporting data driven in-network processing for low-power wireless sensor networks, in: Proceedings of the 17th ACM/IEEE International Conference on Information Processing in Sensor Networks, in: IPSN '18, IEEE Press, Piscataway, NJ, USA, 2018, pp. 72–83, http://dx.doi.org/10.1109/IPSN.2018.00014.

[79] W. Du, J.C. Liando, H. Zhang, M. Li, When pipelines meet fountain: fast data dissemination in wireless sensor networks, in: Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems, in: SenSys '15, ACM, New York, NY, USA, 2015, pp. 365–378, http://dx.doi.org/10.1145/2809695.2809721, URL http://doi.acm.org/10.1145/2809695.2809721.

[80] M. Mohammad, M. Doddavenkatappa, M.C. Chan, Improving performance of synchronous transmission-based protocols using capture effect over multichannels, ACM Trans. Sen. Netw. (2017) http://dx.doi.org/10.1145/3043790, URL http://doi.acm.org/10.1145/3043790.

[81] P. Zhang, A.Y. Gao, O. Theel, Less is more: learning more with concurrent transmissions for energy-efficient flooding, in: Proceedings of the 14th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, in: MobiQuitous 2017, ACM, New York, NY, USA, 2017, pp. 323–332, http://dx.doi.org/10.1145/3144457.3144482.

[82] F. Sutton, R. Da Forno, D. Gschwend, T. Gsell, R. Lim, J. Beutel, L. Thiele, The design of a responsive and energy-efficient event-triggered wireless sensing system, in: Proceedings of the 2017 International Conference on Embedded Wireless Systems and Networks, in: EWSN '17, Junction Publishing, USA, 2017, pp. 144–155, URL http://dl.acm.org/citation.cfm?id=3108009.3108028.

[83] C. Sarkar, R.V. Prasad, R.T. Rajan, K. Langendoen, Sleeping beauty: efficient communication for node scheduling, in: 2016 IEEE 13th International Conference on Mobile Ad Hoc and Sensor Systems (MASS), 2016, pp. 56–64, http://dx.doi.org/10.1109/MASS.2016.018.

[84] L. Mottola, G.P. Picco, Middleware for wireless sensor networks: An outlook, J. Internet Serv. Appl. 3 (1) (2012) 31–39, http://dx.doi.org/10.1007/s13174-011-0046-7, URL https://link.springer.com/article/10.1007/s13174-011-0046-7.

[85] Advanticsys, MTM-CM5000-MSP 802.15.4 telosb mote module, 2019, [Online] URL https://www.advanticsys.com/shop/mtmcm5000msp-p-14.html. (Accessed 11 January 2019).

[86] V. Poirot, O. Landsiedel, Dimmer: self-adaptive network-wide flooding with reinforcement learning, in: 2021 IEEE 41st International Conference on Distributed Computing Systems, ICDCS, 2021, pp. 293–303, http://dx.doi.org/10.1109/ICDCS51616.2021.00036.

[87] M. Baddeley, A. Aijaz, U. Raza, A. Stanoev, Y. Jin, M. Schuß, C.A. Boano, G. Oikonomou, 6TiSCH++ with bluetooth 5 and concurrent transmissions, 2020, Cs arXiv:2010.09529.

[88] B.-M. Cho, S. Kim, K.-D. Kim, K.-J. Park, A controller switching mechanism for resilient wireless sensor–actuator networks, Appl. Sci. 12 (4) (2022) 1841, http://dx.doi.org/10.3390/app12041841, URL https://www.mdpi.com/2076-3417/12/4/1841.

[89] J. Oostvogels, F. Yang, S. Michiels, D. Hughes, Zero-wire: A deterministic and low-latency wireless bus through symbol-synchronous transmission of optical signals, in: Proceedings of the 18th Conference on Embedded Networked Sensor Systems, Association for Computing Machinery, New York, NY, USA, 2020, pp. 164–178.

[90] X. Ma, P. Zhang, O. Theel, J. Wei, Gathering data with packet-in-packet in wireless sensor networks, Comput. Netw. 170 (2020) 107124, http://dx.doi.org/10.1016/j.comnet.2020.107124, URL https://www.sciencedirect.com/science/article/pii/S1389128619306255.

[91] C. Sarkar, R.V. Prasad, K. Langendoen, FLEET: when time-bounded communication meets high energy-efficiency, IEEE Access 7 (2019) 77555–77568, http://dx.doi.org/10.1109/ACCESS.2019.2920937.

[92] F. Mager, D. Baumann, C. Herrmann, S. Trimpe, M. Zimmerling, Scaling beyond bandwidth limitations: wireless control with stability guarantees under overload, ACM Trans. Cyber-Phys. Syst. (2021) http://dx.doi.org/10.1145/3502299.

[93] R. Jacob, Baloo, 2018, [Online] URL http://www.romainjacob.net/baloo/ (Accessed 11 January 2019).

[94] C. Lu, B. Blum, T. Abdelzaher, J. Stankovic, T. He, RAP: A real-time communication architecture for large-scale wireless sensor networks, in: Proceedings. Eighth IEEE Real-Time and Embedded Technology and Applications Symposium, 2002, pp. 55–66, http://dx.doi.org/10.1109/RTTAS.2002.1137381.

[95] J.A. Stankovic, T.F. Abdelzaher, L. Chenyang, S. Lui, J.C. Hou, Real-time communication and coordination in embedded sensor networks, Proc. IEEE 91 (7) (2003) 1002–1022, http://dx.doi.org/10.1109/JPROC.2003.814620.

[96] T. He, J.A. Stankovic, C. Lu, T. Abdelzaher, SPEED: A stateless protocol for real-time communication in sensor networks, in: 23rd International Conference on Distributed Computing Systems, 2003. Proceedings, 2003, pp. 46–55, http://dx.doi.org/10.1109/ICDCS.2003.1203451.

[97] International Electrotechnical Commission (IEC), Industrial networks - wireless communication network and communication profiles - wirelesshart, 2020, URL https://webstore.iec.ch/publication/24433. (Accessed 14 April 2020).

[98] ISA100, Wireless compliance institute, 2009, URL http://www.isa100wci.org/.

[99] W. Watteyne, P. Tuset-Peiro, X. Vilajosana, S. Pollin, B. Krishnamachari, Teaching communication technologies and standards for the industrial IoT? use 6tisch!, IEEE Commun. Mag. 55 (5) (2017) 132–137, http://dx.doi.org/10.1109/MCOM.2017.1700013.

[100] A. Saifullah, Y. Xu, C. Lu, Y. Chen, Real-time scheduling for wirelesshart networks, in: 2010 31st IEEE Real-Time Systems Symposium, 2010, pp. 150–159, http://dx.doi.org/10.1109/RTSS.2010.41.

[101] A. Saifullah, Y. Xu, C. Lu, Y. Chen, End-to-end communication delay analysis in industrial wireless networks, IEEE Trans. Comput. 64 (5) (2015) 1361–1374, http://dx.doi.org/10.1109/TC.2014.2322609.

[102] M. Zimmerling, L. Mottola, P. Kumar, F. Ferrari, L. Thiele, Adaptive real-time communication for wireless cyber-physical systems, ACM Trans. Cyber-Phys. Syst. 1 (2) (2017) 8:1–8:29, http://dx.doi.org/10.1145/3012005.

[103] R. Jacob, Distributed real-time protocol – DRP, 2019, URL https://github.com/romain-jacob/drp.

[104] T. Abdelzaher, K. Shin, Combined task and message scheduling in distributed real-time systems, IEEE Trans. Parallel Distrib. Syst. 10 (11) (1999) 1179–1191, http://dx.doi.org/10.1109/71.809575.

[105] S.S. Craciunas, R.S. Oliver, Combined task- and network-level scheduling for distributed time-triggered systems, Real-Time Syst. 52 (2) (2016) 161–200, http://dx.doi.org/10.1007/s11241-015-9244-x, URL http://dx.doi.org/10.1007/s11241-015-9244-x.

[106] M. Ashjaei, N. Khalilzad, S. Mubeen, M. Behnam, I. Sander, L. Almeida, T. Nolte, Designing end-to-end resource reservations in predictable distributed embedded systems, Real-Time Syst. 53 (6) (2017) 916–956, http://dx.doi.org/10.1007/s11241-017-9283-6.

[107] G. Fohler, Changing operational modes in the context of pre run-time scheduling, IEICE Trans. Inf. Syst. 76 (11) (1993) 1333–1340, URL https://pdfs.semanticscholar.org/272b/615266e763369e903dcb0b966e22077f127c.pdf.

[108] Wikipedia, Bin packing problem, 2019, Wikipedia URL https://en.wikipedia.org/w/index.php?title=Bin_packing_problem&oldid=921684516.

[109] A. Azim, Scheduling of Overload-Tolerant Computation and Multi-Mode Communication in Real-Time Systems (Ph.D. thesis), University of Waterloo, 2014, URL http://hdl.handle.net/10012/8973.

[110] W. Steiner, An evaluation of SMT-based schedule synthesis for time-triggered multi-hop networks, in: 2010 31st IEEE Real-Time Systems Symposium, 2010, pp. 375–384, http://dx.doi.org/10.1109/RTSS.2010.25.

[111] S.S. Craciunas, R.S. Oliver, SMT-based task- and network-level static schedule generation for time-triggered networked systems, in: Proceedings of the 22Nd International Conference on Real-Time Networks and Systems, in: RTNS '14, ACM, New York, NY, USA, 2014, pp. 45:45–45:54, http://dx.doi.org/10.1145/2659787.2659812, URL http://doi.acm.org/10.1145/2659787.2659812.

[112] J. Huang, J.O. Blech, A. Raabe, C. Buckl, A. Knoll, Static scheduling of a time-triggered network-on-chip based on SMT solving, in: 2012 Design, Automation Test in Europe Conference Exhibition (DATE), 2012, pp. 509–514, http://dx.doi.org/10.1109/DATE.2012.6176522.

[113] K. Jeffay, D.F. Stanat, C.U. Martel, On non-preemptive scheduling of period and sporadic tasks, in: Proceedings Twelfth Real-Time Systems Symposium, 1991, pp. 129–139, http://dx.doi.org/10.1109/REAL.1991.160366.

[114] R. Jacob, The Time-Triggered Wireless Architecture URL https://ttw.ethz.ch/.

[115] IoT Benchmarks Initiative, IoTBench URL https://www.iotbench.ethz.ch/.

[116] A. Spina, M. Breza, N. Dulay, J. McCann, XPC: fast and reliable synchronous transmission protocols for 2-phase commit and 3-phase commit, 2019, Cs arXiv:1910.09941.

[117] Semtech, SX1262. Long Range Low Power LoRa URL https://www.semtech.com/products/wireless-rf/lora-transceivers/sx1262.

[118] Nordic Semiconductors, Nrf52840, 2018, [Online] URL https://www.nordicsemi.com/Products/Low-power-short-range-wireless/nRF52840. (Accessed 16 May 2020).

[119] L. Tuchtenhagen, More Easy Synchronous Transmissions: Expanding Baloo to the nRF52840 (Ph.D. thesis), 2022, URL https://www.ds.informatik.uni-kiel.de/en/teaching/bachelor-and-master-theses/completed-master-and-bachelor-theses/2021_Bachelor-Thesis-Lars-Tuchtenhagen.pdf.

[120] D. Baumann, F. Mager, U. Wetzker, L. Thiele, M. Zimmerling, S. Trimpe, Wireless control for smart manufacturing: recent approaches and open challenges, Proc. IEEE 109 (4) (2021) 441–467, http://dx.doi.org/10.1109/JPROC.2020.3032633.

[121] V. Poirot, B. Al Nahas, O. Landsiedel, Paxos made wireless: consensus in the air, in: Proceedings of the 2019 International Conference on Embedded Wireless Systems and Networks, in: EWSN '19, Junction Publishing, USA, 2019, pp. 1–12, URL http://dl.acm.org/citation.cfm?id=3324320.3324322.