Zero-Wire: A Deterministic and Low-Latency Wireless Bus Through Symbol-Synchronous Transmission of Optical Signals

Jonathan Oostvogels, Fan Yang, Sam Michiels, and Danny Hughes
imec-DistriNet, KU Leuven
3001 Leuven, Belgium
<firstname.lastname>@kuleuven.be

ABSTRACT

The performance dichotomy between wired and wireless networks for the Internet of Things primarily arises from the inherent complexity and inefficiency of networking abstractions such as routing, medium access control and store-and-forward packet switching. This paper aims to enable a new class of latency-sensitive applications by breaking all three of these abstractions to deliver a performance envelope that resembles that of a wired bus in terms of deterministic latency and throughput. The essence of this approach is a novel networking paradigm for optical wireless communication, referred to as a symbol-synchronous bus, wherein a mesh of nodes concurrently transmit LED-based signals. This paper realises the paradigm within a platform called Zero-Wire and evaluates it on a 25-node testbed under laboratory conditions. Key end-to-end performance measurements on this physical prototype include 19 kbps of contention-agnostic goodput, interface-level latency under 1 ms for two-byte frames across four hops, jitter on the order of 10s of µs, and a base reliability of 99%. These first results indicate a bright future for the under-explored area of optical wireless mesh networks in delivering ubiquitous connectivity through a simple and low-cost physical layer.

CCS CONCEPTS

• Networks → Link-layer protocols; **Routing protocols**; **Cross-layer protocols**; **Sensor networks**; **• Computer systems organization → Sensor networks**.

KEYWORDS

Internet of Things, wireless sensor networks, cyber-physical systems, visible light communication, optical wireless communication, concurrent transmission, synchronous transmission, cut-through forwarding, bus networks

ACM Reference Format:
Jonathan Oostvogels, Fan Yang, Sam Michiels, and Danny Hughes. 2020. Zero-Wire: A Deterministic and Low-Latency Wireless Bus Through Symbol-Synchronous Transmission of Optical Signals. In *The 18th ACM Conference on Embedded Networked Sensor Systems (SenSys '20), November 16–19, 2020, Virtual Event, Japan.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3384419.3430897

SenSys '20, November 16–19, 2020, Virtual Event, Japan
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-7590-0/20/11…$15.00
https://doi.org/10.1145/3384419.3430897

1 INTRODUCTION

Attaining wire-like performance has long been a goal of wireless networking research. Indeed, successful wireless technologies for the Internet of Things (IoT) are being marketed as achieving "wire-like reliability" [92], promising to replace costly and hard-to-maintain wired industrial networks with a drop-in wireless solution [28]. These successes build on years of research into link-layer, Medium Access Control (MAC) and routing strategies, resulting in a set of industry standards that use wireless mesh networks as a platform for embedded sensing and actuation (e.g. 6TiSCH [87], ISA100.11a [69], WirelessHART [78]).

Despite these advances, key challenges must be answered if wireless networks are to truly replace their wired counterparts. In particular, wireless sensor meshes' reliance on store-and-forward routing and Time Division Multiple Access (TDMA) makes end-to-end latency, i.e. the time it takes for a frame to be transmitted from a sending to a receiving interface across the network, far larger and much more variable than for wired industrial control networks such as CAN [47, 79]. Since reducing latency and jitter is a key consideration in the design and operation of the latter type of network [58, 88], the applicability of wireless meshes in latency-sensitive or event-driven scenarios, such as real-time control loops [64, 93], delay-sensitive actuation [22] and robotics [21], remains limited [47]. Merely eliminating routing or TDMA does not suffice to close the performance gap between wired and wireless: contention-based star topologies, also found in wide area networks such as LoRaWAN [9], may avoid some of the difficulties of time-synchronised meshing, but fall even further short of addressing those demanding application scenarios, since their large interference domains cause poor link utilisation and indeterministic performance [8, 48]. Aspiring to overcome these limitations, researchers have explored systems that break conventional MAC and routing abstraction layers through, for example, synchronous transmission [100]. Still, none of these approaches match the performance of even the simplest wired networks, where e.g. I^2C communicates short frames at < 1 ms of contention-agnostic latency [51].

If wireless networks are to one day support the variety of embedded applications that still depend on wires, innovative solutions will be needed. This paper proposes one such solution, the essence of which is to eliminate store-and-forward packet switching and to replace it with a novel networking paradigm, the *symbol-synchronous bus*, which makes a wireless mesh network behave like a single wire. In such buses, nodes do not wait for the complete reception of a link-layer data frame, or even a symbol, before re-broadcasting it to their neighbours. In pursuit of this vision, we introduce the Zero-Wire platform, which is a hardware and software prototype of a symbol-synchronous bus that uses Optical Wireless Communication (OWC), i.e. free-space signaling between commonly available light sources and photodetectors. OWC is a natural medium on which to implement the proposed approach due to the abundance of largely unregulated bandwidth in the optical part of the electromagnetic spectrum, its remarkable interference characteristics [70], and the simplicity with which OWC platforms expose the entire network stack for research [35]. While the physical and link-layer aspects of OWC have been studied extensively [65], research to enable end-to-end connectivity across dense multi-hop deployments is at a very early stage [17]. Still, such meshing may help mitigate considerable drawbacks of optical wireless [59], namely its line-of-sight requirements and range limitations [65]. This paper introduces a novel and empirically verified approach towards multi-hop OWC, which we believe to be particularly promising for fine-grained real-time instrumentation of hard-to-reach industrial environments (cf. [43, 54]), especially in application scenarios where radio links are undesirable, prohibited or impossible [17, 36].

Empirical results on a 25-node testbed confirm Zero-Wire's suitability for low-latency deterministic networking. Key end-to-end measurements include 19 kbps of flow-independent goodput, deterministic latency under 1 ms for two-byte frames, 99% base reliability, and priority-based contention resolution with zero delay. The remainder of this paper explains in more detail how symbol-synchronous buses and Zero-Wire achieve these results. Section 2 provides the necessary background on store-and-forward IoT networking, wired buses and OWC. Next, Section 3 discusses how some of the introduced concepts combine to explain the operation of a symbol-synchronous bus, and it identifies the key challenges in implementing the new paradigm. Section 4 discusses how Zero-Wire's design addresses these challenges, while Section 5 describes the physical implementation of a node prototype. Section 6 evaluates Zero-Wire's performance under laboratory conditions, and Section 7 reviews related work. Section 8 concludes the paper and elaborates on future research opportunities. To encourage such further efforts, we make the source materials for the Zero-Wire platform publicly available on https://github.com/jonathanoostvogels/zero-wire-public.

2 BACKGROUND

This section provides the background information that is required to understand the design of Zero-Wire. This includes the drawbacks of store-and-forward packet switching for the IoT, attractive features found in wired buses, and a review of the fundamental differences between optical and radio communication. These issues are discussed in Section 2.1 to 2.3 respectively.

2.1 Store-and-forward packet switching

Current IoT networking technologies struggle to support low-latency applications, particularly in dense network topologies. TDMA mesh networks over IEEE 802.15.4 radios, for example, introduce 100s to 1000s of milliseconds of latency and 10s of milliseconds of jitter in typical configurations [47], yet control loops for e.g. tactile feedback or robotics require deterministic end-to-end delays of under 1 ms [77]. Similarly, end-to-end and single-link throughput measures in duty-cycled sensor networks differ by roughly an order of magnitude, unless a specific multi-hop path is optimised for during a period of time corresponding to many packet transmissions (cf. [6, 30]). While the performance characteristics of certain traffic flows can be improved by favouring them over others in e.g. 6TiSCH's scheduling function [19], by manipulating timeslot duration [10], by setting up explicit end-to-end connections [72], or by adjusting node duty cycles [16], the resulting monopolisation of network-wide resources is hard to reconcile with applications involving bursty, dynamic, in-network, any-to-(m)any or event-driven traffic flows [29, 30, 40, 61]: configuration changes must happen before they are needed and cannot simultaneously favour all traffic patterns over all others. Low-latency wireless is often also subject to considerable protocol and hardware limitations: 15.4 timeslot duration at 2.4 GHz cannot drop below 0.96 ms, of which radio turnaround time — the 0.192 ms it takes to switch between transmit and receive mode — makes up a considerable fraction [10].

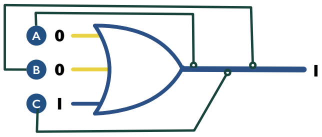

Figure 1: Wired buses can exploit dominant (1) and recessive (0) logic levels, as well as channel sensing, to non-destructively resolve contention.

Striving for IoT networks that better suit latency-sensitive in-network applications, researchers have investigated *Synchronous Transmission (ST)* [100]. ST protocols referred to as *primitives* [44] do not prevent nodes from interfering with each other's transmissions, but aim to correctly decode frames in spite of concurrent channel access by synchronising transmitters that send a series of link-layer anycasts. Primitives then realise an application-level goal without prescribing the hop-by-hop traffic flows that implement it. Glossy [33], a seminal example, enables an *initiator node* to broadcast a 15.4 frame through a mesh network. Nodes that receive the frame repeat it while synchronising their transmissions down to 0.5 μs. Such synchronisation allows nodes to correctly decode concurrently transmitted frames with high probability due to a variety of modulation-specific mechanisms that are commonly misinterpreted as constructive interference [55, 94]. The process repeats until it is assumed to have flooded the entire network.

ST enables end-to-end communication at a few milliseconds of latency [33] and a significantly higher sustained throughput than TDMA-based multi-hop networks [25–27]. However, ST remains a store-and-forward solution: nodes receive and decode all physical-layer symbols in a link-layer frame before transmitting themselves. Latency bounds are therefore heavily impacted by network diameter, frame length, and the repetition of preambles and headers (0.352 ms per repetition) [33]. When multiple initiators are present or multiple packets need to be sent, ST nodes should also prevent distinct traffic flows from interfering. Combined with ST primitives' lack of global knowledge, this consideration reintroduces the need for MAC and centrally coordinated scheduling (cf. [32]).

2.2 Bus networks

Wired buses are conceptually much simpler than wireless networks: the former connect all nodes through a single, shared set of wires. These wires provide fast and deterministic half-duplex links that

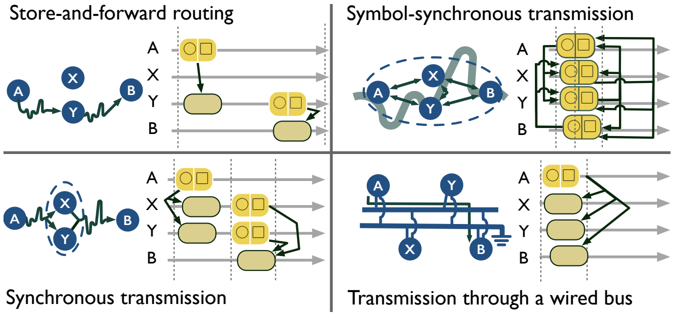
Store-and-forward routing
Symbol-synchronous transmission
Synchronous transmission
Transmission through a wired bus
A
X
Y
B

Figure 2: Symbol-synchronous transmissions mimic the time behaviour and global consistency of wired buses, while store-and-forward systems do not.

are isolated from their environment and offer all nodes a consistent view of the network: any signal written to the bus effectively reaches all nodes instantaneously (i.e. after a propagation delay that is small relative to the symbol duration [18]). As all nodes on the bus see all traffic, they can refrain from interfering with ongoing transmissions and may sense the channel to detect collisions while transmitting (i.e. apply CSMA/CD [84]). In some buses, the latter mechanism enables *priority-based arbitration* [18], which exploits the dominance of some logic levels over others to non-destructively resolve channel contention (cf. Figure 1): if multiple nodes transmit concurrently, nodes suppress low-priority traffic before it corrupts a high-priority frame [18, 51]. This results in *perfect* network utilisation for those buses: the network's performance is independent of the number of nodes, their density or the required traffic patterns.

The simplicity, performance and determinism of wired buses has made them a popular communication interface for both resource-constrained electronic peripherals (SPI, I^{2}C, 1-Wire [11, 51]) and real-time industrial control (*fieldbuses* such as Profibus and CAN [31, 90]). Considering performance and functionality, contemporary wireless buses do not fully live up to their name due to their reliance on an additional orthogonal channel per connected node, TDMA or wired control lines (cf. [32, 52, 62]). Moreover, compelling features offered by wired buses (arbitration, global consistency, CSMA/CD, flow-independent performance) are commonly believed to be impractical or infeasible in a wireless context [95].

2.3 Optical wireless communication

Although (visible) light has been used as a communication medium since antiquity [73], and lasers have found occasional use as point-to-point free-space links [39, 84], the recent surge in research interest towards optical wireless networking owes its existence to the proliferation of LED-based lighting [65, 83]: the intensity of cheap off-the-shelf LEDs can be modulated at tens of megahertz of bandwidth to wirelessly transmit data [65]. At the receiver side, photodetectors (i.e. phototransistors and -diodes) recover the data by converting the intensity level of ambient light into an electric current. Such *intensity modulation with direct detection (IM/DD)* [65] differs from conventional radio-based modulation, since it embeds information in the power envelope of a signal, rather than in the low-level properties of the underlying electromagnetic waves, such as their phase, amplitude, or frequency. This difference affects the way in which concurrent signals interfere. Intuitively, if two LEDs are modulated to transmit the same signal at the same time, the intensity of the light as seen by a detector is roughly twice that of one LED. Doing the same thing with two radios does not double the amplitude of the transmitted electromagnetic wave, since phenomena outside the control of any single transmitter obscure the resulting signal. Specifically, carrier frequency offsets between transmitters induce *beatings* in the resulting signal [55], while signal-to-noise ratios vary wildly with the receiving radio's position due to the location dependency of the signals' phase offset [94]. LED-based signals are far less susceptible to these phenomena, because LEDs are *incoherent* emitters: the *coherence time* of the waves that make up LED light, i.e. the period of time during which the phase of a wave remains consistent, is on the order of 10^{-13} s [23], orders of magnitude lower than the sampling interval of any reasonable intensity detector. This means that interference between IM/DD signals of incoherent emitters can be understood, mitigated and designed for at the (de)modulation level (cf. [70]): the resulting signal is the sum of the power envelopes of the interfering signals, and it is not meaningfully affected by out-of-reach low-level phenomena.

Optical wireless communication also differs from radio-based communication in its directionality. The intensity of the light emitted by an LED is a rapidly decreasing function of the angle relative to its central axis [65]. Similarly, the intensity levels perceived by a photodetector decrease with incoming signals' angle of incidence [65]. This directionality enables full-duplex communication on the same channel [91] as, unlike conventional radios, a node's emitter can be oriented away from its detector to avoid overpowering transmissions from more distant nodes; however, it complicates the design of mesh networks (cf. Section 7.4).

3 PARADIGM

3.1 A symbol-synchronous wireless bus

This paper introduces a new type of wireless mesh network, referred to as a *symbol-synchronous* bus, offering sub-millisecond end-to-end latency for small frames, microseconds of jitter and perfect priority-based network utilisation. As illustrated in Figure 2, attaining such bus-like behaviour in a mesh network requires parallellising retransmissions between source and destination nodes as much as possible, as well as avoiding MAC overhead between possible forwarders. More precisely, in a symbol-synchronous bus, *relay nodes* relay any symbol immediately after observing it, without coordinating with other nodes, and without waiting to make forwarding decisions based on the contents of the message. Peripherals, such as IoT sensors and actuators, are not required to relay themselves, and may instead connect to a mesh of full-duplex relays, referred to as the network's *canopy*.

Other approaches towards such network-level parallellism exist (cf. Section 7). The key innovation of the proposed type of network lies in the way relay nodes avoid feedback loops between their concurrent retransmissions. As illustrated in Figure 3, nodes relay incoming symbols with a relay offset time r such that the time offset q between a signal's first arrival and any relayed version perceptible to the same node is smaller than the symbol duration s. As shown in the figure, this constraint enables nodes to distinguish symbols that they still have to relay from symbols they have already forwarded and should not be relayed again by the same node. The point in time s after a symbol's first arrival thus becomes a demarcation point between old and new data, so nodes can avoid waiting for message identifiers or addressing information, and do

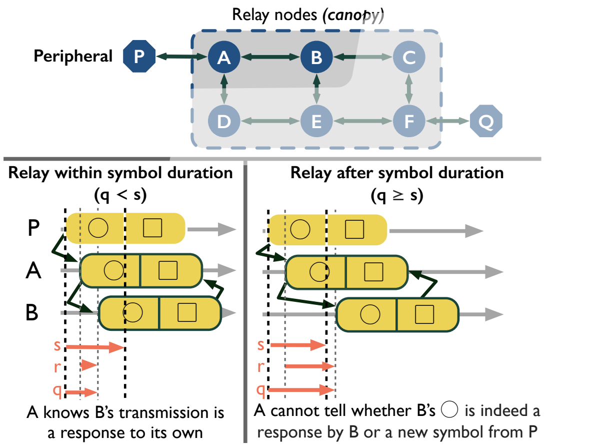
Relay nodes *(canopy)*
Peripheral
P
A
B
C
D
E
F
Q
Relay within symbol duration
(q < s)
Relay after symbol duration
(q ≥ s)
P
A
B
s
r
q
A knows B's transmission is
a response to its own
A cannot tell whether B's ○ is indeed a
response by B or a new symbol from P

Figure 3: Symbol-synchronicity prevents nodes from re-relaying signals they themselves first relayed.

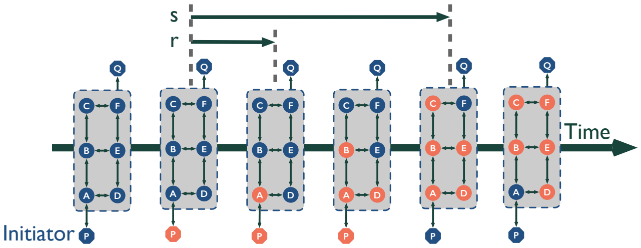

Figure 4: Changes between symbols propagate through the network in a wave-like fashion.

not require multiple transmission bands. Equivalently, a sending node maintains a symbol duration that is slightly longer than necessary for a receiver to decode the symbol: in the figure, P's value for s leaves time for the local effects of A's relay step to stabilise. This constraint guarantees at least one moment during the transmission of every symbol where nodes in a one-hop neighbourhood are *symbol-synchronous*: they all transmit and receive the same symbol at the same time.

As shown in Figure 2, the result of symbol-synchronicity is virtually deterministic timing behaviour similar to that of wired buses. For both types of network, latency is determined only by the transmission time of a single frame and a relatively small propagation delay. Specifically, for a symbol-synchronous bus, end-to-end latency l for an N-symbol frame is bounded by its transmission time $p = Ns$, the per-hop relay offset r, and the network diameter d: $Ns \leq l \leq Ns + rd$, where $r < s$ and typically $rd \ll Ns$, as will be demonstrated later in this paper[1]. As with store-and-forward solutions, latency increases linearly with the number of hops separating the initiator and receiver, yet every hop adds only $r < s$ microseconds instead of the transmission time of a full frame (i.e. Nsd). This approach subsumes routing and medium access control among forwarders: the entire canopy transmits concurrently, so the bus simultaneously connects source and destination(s) through any and all paths that exist.

As illustrated in Figure 4, in a symbol-synchronous bus, changes from one symbol to the next move away from the node that initiated the transmission in a wave-like fashion. Nodes propagate the wave by switching from the transmission of one symbol to to the next. The symbol-synchronicity time constraint essentially guarantees that no node observes two such wave fronts at the same time, thus disambiguating which symbol needs to be transmitted. Symbol-synchronicity hence is a local concept: it considers the signals transmitted and received by a node and its immediate neighbours. An initiator is free to start transmitting the next symbol as long as it can guarantee that doing so will not cause the new wave front to collide with old ones. This implies that, in some cases, a stricter time constraint is needed. Concretely, since the transmission of subsequent frames may be initiated by different nodes, the end of a frame should in general be indicated in a *globally* symbol-synchronous way. In other words, to prevent wave fronts belonging to different frames from colliding, all nodes must agree that a frame's transmission is over before the next one can be sent. Analogously, if we hope to implement a symbol-synchronous bus that provides non-destructive arbitration by having a symbol transmitted by one node dominate those sent by contending transmitters, all nodes must agree on which symbol is being transmitted before the next one is sent (i.e. $rd < s$). In both cases, propagation delays induce a maximum symbol rate $B = 1/s$, at which the network can operate reliably, as is the case for wired buses [50].

3.2 Synchronous inter-signal interference

To successfully propagate and decode information, nodes must be able to discern a single initiator's signal among a collectively much stronger set of slightly time-offset signals that are emitted by neighbouring relay nodes. More precisely, as shown in Figures 2 and 3, nodes that wish to relay the initiator's signal or decode the data that it conveys must deal with *Synchronous Inter-Signal Interference (SISI)*, i.e. they must recognise the initiator's next symbol while it is being interfered with by many more transmissions of the previous symbol that lag behind it. Since the network in no way constrains what forwarding paths will be used, nodes do not know how many concurrent transmissions there are, nor their exact time offsets or signal strengths. OWC, or intensity modulation of an incoherent transmitter, offers a solution: since interference patterns can be predicted reliably from the choice of symbols and their time offsets alone (cf. Section 2.3), a modulation scheme can be *designed* to help nodes infer what the initiator's signal looks like despite their imperfect understanding of many more interfering signals. Nodes in a symbol-synchronous bus therefore rely on a transceiver that is specifically designed to accomplish this task, rather than one that treats interference as a given for a well-established modulation technique (cf. Section 7.2) or tries to cancel it out (cf. Section 7.3). Symbol-synchronous relaying aids this process by constraining what symbols can be transmitted concurrently, and how far apart they can be in time.

3.3 Challenges

Although Sections 3.1 and 3.2 describe principles according to which a wireless mesh network can provide bus-like time behaviour, they do not detail how nodes should implement these principles. To build a concrete symbol-synchronous bus, nodes need to be equipped with a transceiver that is designed to respect the symbol-synchronicity constraint and exploit the interference characteristics

[1] These formulas do not describe scaling behaviour: as evidenced in Section 6.3, s may increase linearly with d in the worst case.

of OWC to overcome SISI. Solving these challenges is a modulation-specific exercise: inferring the initiator's signal requires knowledge of the specific symbols that are being used. Additionally, the symbol-synchronous paradigm only explains how to build wireless networks with the timing characteristics of a wired bus. Other convenient properties of wired networks do not carry over directly. Wireless buses cannot trivially rely on a shared clock line, data links free from ambient interference, or a common ground level relative to which symbols can be decoded (cf. [51]). Similarly, the proposed paradigm cannot exploit dominant versus recessive voltage levels to implement non-destructive arbitration, but will have to use a transceiver that is designed to reliably decode one specific symbol in the presence of contending transmissions.

4 DESIGN

4.1 Overview

The Zero-Wire platform introduced in this section solves the challenges of implementing a symbol-synchronous bus (cf. Section 3.3). We selected a simple modulation scheme based on On-Off Keying (OOK) of LEDs, which has an extremely low barrier to entry for resource-constrained devices. All that is required for a node to transmit data through the network is to blink an LED within reception range of the canopy at an arbitrary but fixed modulation speed. Many other IM/DD modulation schemes have been proposed [65], and exploring their performance trade-offs in symbol-synchronous networks is a target of our future work.

Figure 4 illustrates Zero-Wire's operation. When an initiator turns its LED on or off, that event quickly propagates through the network as nodes react by changing the state of their LEDs at slight time offsets. Nodes follow and decode the state of the initiator's LED, without being disturbed by concurrent transmissions by other nodes, and without any knowledge of network structure or node identity. The key difficulty with this method of operation is preventing false positive detection of symbols. Failing to do so causes spurious symbols to propagate unchecked across the network, since nodes cannot wait until they have verified a frame before retransmitting it, as would be done for store-and-forward networks. The following sections detail the design of a transceiver that solves this problem and implements wire-like features and performance by exploiting the principles outlined in the previous section.

4.2 Transceiver logic

Figure 5 provides a schematic overview of the design of Zero-Wire's transceiver. The transceiver implements a real-time control loop that repeatedly updates a set of interconnected state machines in sequence while interfacing with a photodetector, LED(s) and an external processor. The LEDs allow for the modulation of data, the photodetector enables its demodulation by sampling the intensity level of ambient light, and the state machines process the samples to recover data. The transceiver reads frames to be sent from — and writes frames that were successfully received to — a set of buffers that can also be accessed by the external processor. The control loop interval i is fixed: every i microseconds, a node samples the ambient light level exactly once and performs all processing steps associated with the new sample. If i is sufficiently small, this approach helps to guarantee that nodes recognise and start relaying a new symbol at small and predictable time offset r. To preserve this property when communicating with the external processor, the transceiver reads and writes at most one bit per control loop iteration to each of the data buffers.

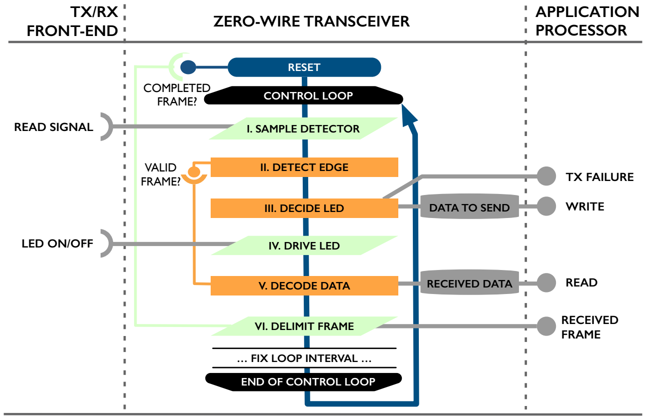

Figure 5: A Zero-Wire transceiver repeatedly updates a set of state machines in sequence to process intensity samples.

The transceiver's design parametrises s in terms of the number of samples per symbol and minimises r. For a sufficiently large s, the symbol-synchronicity time constraint is then satisfied by design. As will be demonstrated in Section 6, the maximum speed at which the network remains operational may therefore be obtained empirically for any given topology. An iteration of the control loop comprises the following six steps, the details of which will be examined by elaborating on Zero-Wire's edge detection logic (Section 4.3), frame structure (Section 4.4), and arbitration mechanism (Section 4.5).

I. The transceiver samples a photodetector, retrieving a number that represents the current intensity of ambient light.
II. A first state machine reasons on a recent history of samples to detect *edges*, i.e. one or more neighbouring nodes that change the state of their LED(s) in response to the initiator's signal. The state machine outputs what it believes to be the current state of the initiator's LED (`on` or `off`).
III. A second state machine decides whether the transceiver's own LED(s) should be on or off during the next i microseconds. That decision considers the output of the first state machine and the presence of data to be sent in the downstream buffer. If arbitration requires a node to suppress its ongoing transmission, this state machine informs the external processor of that event.
IV. The control loop drives the transceiver LED(s) on or off, based on the decision made in the previous step.
V. A third state machine keeps track of the time (i.e. the number of loop iterations) between recent edges and, leveraging Zero-Wire's frame structure, recovers transmitted data.
VI. The control loop decides whether the current loop iteration marks the end of a (valid) frame to be passed to the external processor. That decision also helps infer correct parameter setting for the edge detection logic described in Section 4.3.

4.3 Edge detection

Zero-Wire nodes must detect one or more of their neighbours toggling their LED(s). Relative to point-to-point OOK, symbol-synchronous buses face two additional challenges in doing so. First, the size of modulation-induced intensity changes is hard to predict.

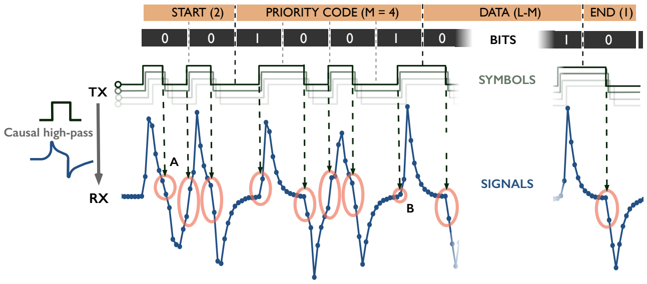

Figure 6: An initiator that toggles its LED(s) induces artefacts in the signal observed by neighbouring nodes.

Any number of neighbours may have preceded a node in toggling their LED(s) between two samples. The magnitude of a signal is then determined by all neighbours, whereas the detection of an initiator's next symbol depends on a signal that is only emitted by some. Secondly, the high-pass filters incorporated by OWC platforms [12, 35, 98] to remove low-frequency interference caused by external light sources are implemented in hardware to preserve a high dynamic range of reception circuitry [98]. As will be shown in Section 6, a symbol-synchronous bus's modulation rate is not easily fixed at design time, complicating design that targets a specific symbol duration. A Zero-Wire transceiver therefore accounts for the fact that symbols may be deformed by a filter's response, rather than depending on filters designed to minimise such deformations. A signal trace recorded on Zero-Wire's hardware platform (cf. Section 5) illustrates both phenomena: Figure 6 shows the effect of an initiator switching the state of its LED(s) on the signal observed in a clique of five relay nodes. The onset of the next symbol is obscured by the filter's response to the previous symbol; an initiator's edge does not result in consistent zero crossing of high-frequency signal components (cf. "A" in the figure), nor does it lead to fixed-size intensity changes ("B").

Figure 6 also reveals, however, that initiators that toggle their LED(s) consistently introduce an artefact in the response of the filter as the signal's shape switches from convex to concave or vice versa. If I_k indicates the intensity level k samples ago, and g the number of samples since the latest edge, Zero-Wire therefore detects edges by checking whether:

$$|I_1 - I_0| > |I_2 - I_1| + \Delta \ \wedge \ g > \frac{s}{i} - 2.$$

The first inequality is satisfied on convexity/concavity changes and examines the signal's second derivative; the second implements symbol-synchronicity by separating artefacts caused by subsequent relays from edges that identify a new symbol, while allowing for clock drift (-1) and imperfect agreement on the start time of a symbol (-1). If Δ is set sufficiently high to protect against noise, an IM/DD signal cannot "accidentally" satisfy the constraints: IM/DD SISI does not create false edges by inducing beatings or destructive interference between out-of-phase carrier waves. This approach does not depend on the initiator's exact signal intensity, nor on symbol duration, nor is it particularly sensitive to filter parameters: Zero-Wire only exploits the typical shape of the filter's step response (cf. [7]). It also imposes a minimum of six control loop iterations per symbol: nodes observe and relay an edge (1) before observing the response of the neighbours for which they relayed (2), which may be just ahead of them in their control loop (3). A node then needs two more samples (I_2, I_1) of the signal reverting towards its zero level (4, 5) before it can detect the next edge (6). As discussed in Section 6, this scheme thus makes control loop processing time and the resulting number of samples per unit of time the limiting factor for Zero-Wire's bus speed.

4.4 Frame structure

Figure 6 illustrates the idiosyncrasies of Zero-Wire's frame structure on a physical and logical level. In symbol-synchronous buses, nodes must begin decoding and relaying a signal before its first symbol completes. This implies that Zero-Wire nodes cannot wait for the completion of an extensive preamble used to synchronise receivers to incoming transmissions before transmitting themselves. The platform therefore relies on the self-clocking nature of Manchester encoding [84]: it encodes `0`- and `1`-bits as `on-off` and `off-on` respectively, forcing an edge in the middle of every bit. Its bus speed (data rate) b thus equals half its symbol rate B. To account for variation in propagation paths and delays, the decoding state machine interprets edges as if they arrived on the nearest multiple of s relative to the last edge and subsequently resynchronises.

To allow for short messages at low latency, Zero-Wire also eliminates other conventional fixed-size fields from its frame structure, i.e. those containing an error-detecting code and those indicating frame length. Instead, the absence of an edge for $2.5s/i$ samples marks the end of a transmission: Manchester encoding forces an edge at least every $2s/i$ samples; edges arriving later would be interpreted as if they had arrived at the wrong point in time anyway, corrupting the message. The platform starts every frame with a two-bit sequence (`00`) and terminates it with a single `0`: the former technique can be exploited to quickly suppress signals resulting from noise-induced, wrongly timed edges and prevents transient responses in the reception hardware from obscuring payload data; the latter guarantees that a transceiver's LED(s) are off at the end of every transmission. Upon the end of a frame, the transceiver is configured to not initiate transmissions for w microseconds. If d is the diameter of the network determined at deployment time, $w = id \approx rd$ is set such that all nodes agree that the transmission of a frame has been completed before the next one starts, thus enforcing inter-frame global symbol-synchronicity (cf. Section 3.1).

Zero-Wire detects frame corruption by constraining its payload size in bits to multiples of a small integer f: messages that do not satisfy the constraint are assumed to have been be corrupted by spurious or missed edges. Larger values for f detect more errors, but reduce network performance by wasting time transmitting padded messages. For the purposes of this paper, we set $f = 4$.

The proposed platform leverages its frame validation scheme to determine the edge detection threshold Δ introduced in Section 4.3. Frame corruption due to spurious or missed edges provides a node with no information concerning the correctness of its parameter setting: any other node might have set Δ inappropriately, thus corrupting the signal for the bus as a whole. Nodes therefore infer Δ by maintaining an exponentially weighted average of the values for $\epsilon = |I_1 - I_0| - |I_2 - I_1|$ they observe for edges in valid frames. We note that setting Δ to $1/8$ of this average value helps tolerate inherent variation in observed ϵ-values without requiring division hardware and without leading to falsely detected edges. Additionally, since Zero-Wire must prevent false symbols from propagating unchecked across the network, the platform is conservative about lowering Δ: the parameter value starts from and never becomes lower than Δ_m, the smallest value that eliminates unchecked symbol propagation induced by random noise. Δ_m is largely determined by the noisiness of the environment and hardware components, so it can be found at deployment time by increasing Δ_m until the network is stable when no messages are being sent.

4.5 Arbitration

Zero-Wire initiators suppress their transmission when they notice their signal is interfered with by a higher-priority frame seizing control of the bus. In particular, they revert to relaying perceived symbols when they observe edges that should not have occurred based on the transmission they initiated. Combined with Manchester encoding and symbol-synchronicity, such collision detection has two interesting properties. First, edges only occur at the boundary between two bits when they are identical. Secondly, when using Zero-Wire's edge detection strategy, the presence of an edge *dominates* its absence: initiators that observe an edge where they did not cause one detect the presence of a contending initiator without corrupting the bit transmitted by the latter. Zero-Wire transceivers exploit these two characteristics to implement priority-based arbitration. Nodes never interrupt ongoing messages and start transmitting available data exactly w/i control loop iterations after the end of the previous message, synchronising contending initiators down to roughly w microseconds. Assuming a sufficiently large symbol duration s as in Section 3.1, this approach enforces a complete ordering on the set of all possible frames, yet, as with e.g. CAN bus [31], it cannot detect the number of initiators that are concurrently transmitting the exact same frame. Zero-Wire leaves it to higher-level protocol designers to incorporate a unique node identifier in frames when messages need to be associated with a unique initiator. As illustrated in Figure 6, under this regime, an application processor prioritises a frame by deciding on and prefixing a priority code to the rest of the frame payload when it passes that payload to the transceiver. The code is generated by mapping a natural number $p < 2^M$, which indicates the priority level, to a string of M bits. Frames with a higher priority level are guaranteed to suppress those with a lower priority without being corrupted by them. For the purposes of this paper, we fix the length of a priority code to $M = 4$. The sequence of bits making up the code $(C[0], \ldots, C[M-1])$ can be computed as follows:

$$C[j](p) = \begin{cases} 0, & \text{if } j = -1 \text{ (recursive base case), else} \\ C[j-1](p), & \text{if } \left\lfloor \frac{p}{2^{M-j-1}} \right\rfloor \bmod 2 = 1, \text{ else} \\ 1 - C[j-1](p), & \text{if } \left\lfloor \frac{p}{2^{M-j-1}} \right\rfloor \bmod 2 = 0. \end{cases}$$

5 IMPLEMENTATION

Zero-Wire builds on DenseVLC's [4, 12] hardware platform, which provides transmitter (TX) and receiver (RX) front-ends. As shown in Figure 7, the TX circuit toggles a high-brightness white LED through a power transistor; the RX front-end samples a photodiode through an amplifier chain consisting of a photodiode, a transimpedance amplifier (TIA), an AC-coupled amplifier that filters incoming signals using an RC high-pass, and a seventh-order low-pass Butterworth filter to prevent aliasing when reading the signal through an analogue-to-digital converter (ADC) over SPI [12].

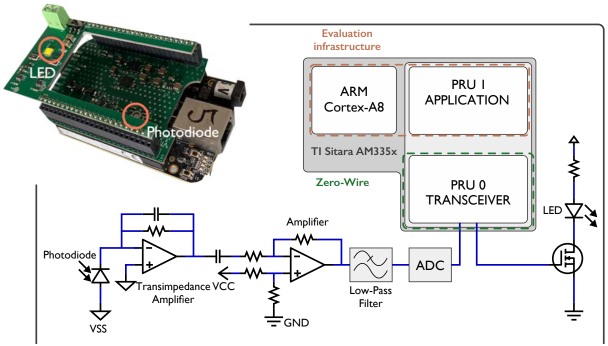

Figure 7: Overview of Zero-Wire's hardware platform.

We implement the Zero-Wire transceiver in software using ~500 lines of C code run on an STM32G474 microcontroller [80], which interfaces with the TX and RX circuitry through its GPIO pins. We have also ported this implementation to the BeagleBone Black board [1]. While both versions are interoperable and attain the same performance levels, the experiments conducted in this paper rely on the infrastructure provided by the BeagleBone, as it allows for easy monitoring and control of a network of nodes that have a convenient form factor (cf. Section 6). In particular, the board carries a single ARM Cortex-A8 core and two real-time coprocessors referred to as *PRUs*. For the purposes of this paper, PRUs are essentially low-complexity microcontrollers that offer single-cycle GPIO access at 200 MHz [86]. Functionally equivalent to the STM32, one of the PRUs (`PRU0`) therefore implements the transceiver logic. `PRU0` leverages its CYCLE register, which provides time measurements at single-cycle (i.e. 5 ns) granularity [85], to make the coprocessor wait for the completion of its control loop interval i. By toggling a GPIO pin on every iteration and determining the lowest setting for i that consistently allows for timely completion of the loop's processing steps with a logic analyser, we establish $i = 2.5\,\mu s$. The remaining PRU (`PRU1`) acts as an external processor that implements application logic. It interfaces with the transceiver through two three-entry, 5.25 kB cyclic buffers in shared RAM to pass data without compromising the transceiver's real-time behaviour, as described in Section 4. The ARM core in turn communicates with `PRU1` using TI's RPMSg framework [3], offering a convenient Linux environment for logging and monitoring services. Figure 7 shows a BeagleBone-based Zero-Wire relay node. LED and photodiode are oriented in the same direction, with the diode outside the field of view of the LED to prevent transmitted signals from saturating the RX amplifier chain. This positioning allows to deploy relay nodes such that they receive signals through their reflection off walls and floors. Relay nodes are mains-powered: as envisioned in Section 3.1, they form a backhaul network that relays traffic initiated by peripherals, which do not relay themselves.

6 EVALUATION

6.1 Testbed set-up

This section evaluates Zero-Wire in terms of end-to-end network performance. In contrast to IEEE 802.15.4 [42], large-scale OWC testbeds that could support our evaluation remain unavailable [17]. We therefore created our own testbed of 25 Zero-Wire relay nodes in a laboratory environment. As shown in Figure 8, the nodes are all positioned in the same plane on a 5-by-5 grid and oriented such that their LEDs point towards a grey painted surface, connecting them

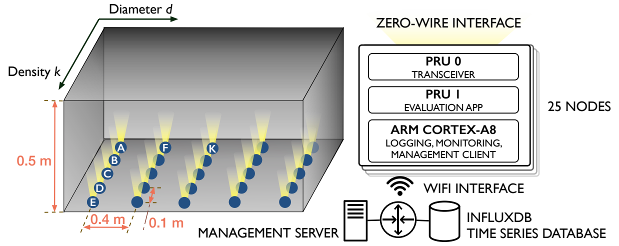

Figure 8: Testbed set-up. Dimensions are such that node A is in range of B through F, but not of K.

through their signals' diffuse reflection. Such a planar topology, which mimics a deployment of ceiling-mounted light bulbs that communicate via their signals' reflections off walls and floors, is both practical and common in contemporary OWC research [4, 12, 57, 96]. The dimensions of the testbed, also shown in the figure, and the relative positions of nodes within it, were chosen to enable an examination of how the network's performance characteristics scale with parameters of the network topology in a controlled setting (cf. Section 6.3). In particular, the setup makes sure the number of columns in the grid of nodes corresponds to the network diameter. Nodes in different columns are positioned one hop apart, so a signal from a node in one column reaches those in neighbouring columns, but not those farther away[2]. By contrast, the number of rows in the grid corresponds to the network's density k, since nodes in the same column form a clique, i.e. they correctly receive each other's signals without requiring intermediate relays. Higher values for k thus provide more links between nodes in neighbouring columns. We position the testbed such that, on one side, it is exposed to a variety of natural and artificial light sources present in our offices over the business days during which the experiments were performed. The three other sides are opaque and diffusely reflect optical signals.

As also shown in Figure 8, each node connects to a WiFi network that provides a management back-plane to applications that are implemented on PRU1 to collect evaluation data without disturbing the symbol-synchronous bus. Concretely, the transceiver logic embodied in PRU0 reports frame transmission and reception to the Beaglebone's other processors and associates a timestamp with each of these events at an $i = 2.5\,\mu$s granularity by inspecting its CYCLE register [85]. An InfluxDB [2] time series database logs these events, providing a complete history of what data was transmitted and received by which node at a specific point in time.

6.2 Performance characteristics

Scenario. We study Zero-Wire's performance characteristics by replicating traffic patterns resembling those used to interact with SPI or I^{2}C-based peripherals. Specifically, we instruct pairs of nodes to exchange frames in a request-reply pattern. One node sends a frame carrying L bits of data through a Zero-Wire bus configured to run at speed b; the receiving node replies with another frame of length L immediately after it detects the end of the request frame. As illustrated in Figure 9, this process then repeats, allowing for the study of end-to-end latency, reliability and goodput. For the purposes of this section, latency is the delay between the time at which the request node completes its request by receiving its own message from the bus and the time at which it completes receiving the corresponding reply at the level of the transceiver. The time it takes for PRU1 to receive the request and pass the corresponding reply to the transceiver is on the order of a few microseconds, and hence has negligible impact on the latency measurements reported below. Reliability then is the proportion of reply frames correctly received by the request node within all reply frames that were sent. Goodput is the number of bits in valid received frames divided by the timespan of the repeated request-reply process from the request node's perspective. We arbitrarily set the number of requests in such an exchange to 10 and subsequently restart the process with a new random pair of nodes, thus eventually collecting information on the performance of every end-to-end path within the network. Such variation of traffic flows does not involve network-wide reconfiguration: no node is aware when another node will transmit; any node can decide to use the full capacity of the bus at any time without prior or consistent resource allocation. Unless indicated otherwise, metrics reported in this evaluation that are based on this procedure average over all end-to-end traffic flows. All data points for $b < 25$ kbps consider > 500 frames.

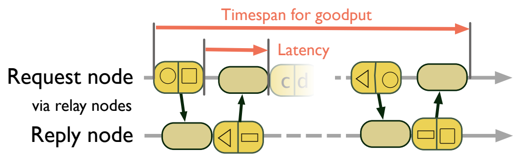

Figure 9: Traffic pattern used for evaluation.

Latency. Figure 10 shows end-to-end latency for Zero-Wire as a function of frame length. Latency is shown to be proportional to the speed of the bus and the amount of data to be transmitted. As shown in the inset, Zero-Wire thus enables sub-millisecond latency as long as the amount of data to be transmitted is sufficiently small relative to the speed of the bus, which in this topology allows to communicate a 2-byte payload across a four-hop network in one millisecond. We discuss the implications of using such small frame sizes in Section 6.4. As shown in the figure and explained in section 4.2, only certain discrete settings for the bus speed b are possible, since symbol duration $s = 1/(2b)$ can only be configured in terms of the number of samples per symbol[3].

In a Zero-Wire network, the key lower bound on latency is the fixed time cost associated with transmitting the three start-of-frame and end-of-frame bits (cf. Section 4.4), as evidenced by the non-zero y-axis intercept of the lines in the figure. The rest of the latency profile is driven by payload size only. The obtained latency numbers are virtually deterministic: for any given bus speed and frame length, they vary by no more than tens of microseconds across traffic endpoints or experimental replicates. Variance in latency due to hop-by-hop relaying across paths of different lengths is negligible, since the main sources of such variance are propagation delay between neighbouring nodes ($r \approx i = 2.5\,\mu$s per hop) and differences in the positions of nodes within their control loops.

Reliability. Figure 11 illustrates Zero-Wire's reliability profile. For messages no longer than a few bytes and up to a bus speed of 20 kbps, the end-to-end frame delivery ratio is practically constant at 99%, after which reliability collapses as the symbol-synchronicity time constraint can no longer be met consistently (cf. Section 3.1).

[2] This assumes a standard configuration of the network, i.e. 16-bit frames, frame delivery ratio across one hop in isolation $> 95\%$; across a distance corresponding to two hops without intermediate relay $< 0.5\%$; $s/i = 9$.

[3] E.g. for $s/i = 9$, the bus speed is $1/(9 * 2.5 * 10^{-3} * 2) \approx 22.2$ kbps.

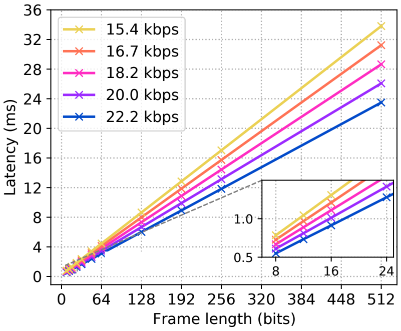

Figure 10: End-to-end latency.

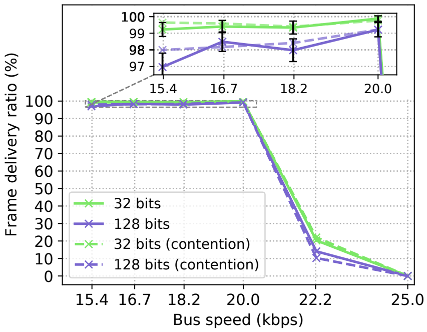

Figure 11: End-to-end reliability.

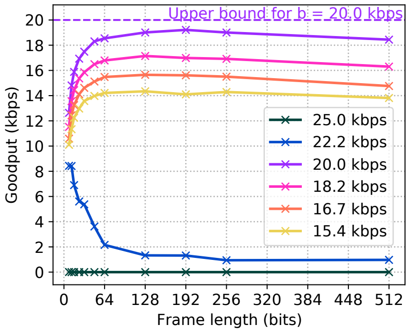

Figure 12: End-to-end goodput.

Reliability primarily depends on frame length and favours short frames: delivery ratios decrease by roughly one percentage point when using 128-bit instead of 32-bit frames. Contention, on the other hand, has no significant effect. When configuring two nodes to reply to a request using a different priority code, the high-priority frame is received as reliably as frames that are not contended with.

The maximum speed of 20 kbps corresponds to a transceiver configuration with 10 samples per symbol. When increasing speed and hence reducing the number of samples per symbol, we note that Zero-Wire networks deteriorate before the limit of 6 samples per symbol anticipated in Section 4.3. We believe this effect occurs because interference domains are larger than communication domains: short-term variance in observed intensity values for a given symbol (i.e. ϵ in Section 4.3) results in sporadic interference from nodes that are more than one hop away.

Variance in perceived intensity values is also the main cause of frame loss. We observe that frames that get corrupted are typically received as several fragments because ϵ-values are occasionally too small to lead to successful edge detection by being indistinguishable from random noise. Due to the absence of a timely edge, nodes then prematurely terminate an ongoing frame. Other failure modes, such as spurious symbols corrupting a transmission, are rare, since Δ is set conservatively (cf. Section 4.4). While future research may improve Zero-Wire's reliability with more advanced signal processing, these observation suggest that the relationship between frame length and reliability is inherently stronger for a symbol-synchronous bus than for conventional mesh networks: nodes must decide how to interpret a symbol while that symbol is ongoing, and cannot wait for the completion of a frame to mask transmission errors before relaying it.

Goodput. Figure 12 details end-to-end goodput for a Zero-Wire network. Goodput increases with bus speed up to $b = 20$ kbps, i.e up to the point at which the symbol-synchronicity constraint breaks and the network stops operating reliably. When considering frame length, goodput is subject to a trade-off: longer frames allow to amortise the three bits of per-frame overhead[4] over a larger amount of payload data, but are also considerably more likely to be corrupted. Goodput-optimal frame lengths are situated between 128 and 256 bits bits and result in a goodput of 19 kbps at a bus speed setting of 20 kbps. In that setting, Zero-Wire thus utilises 95% of its theoretical throughput capacity for application data. Shorter frames enable low-latency applications, but incur a considerable goodput penalty: when using 8-bit frames, the network attains only 63% of its throughput capacity. Larger-than-optimal frames cause slow goodput loss, so they may be better sent as fragments.

6.3 Scaling behaviour

Scenario. We evaluate scaling behaviour by removing rows and columns of nodes from the deployment in Figure 8. Removing a column reduces the number of nodes in the network by five and decreases its diameter d. Due to the parameters of our testbed, such *diameter scaling* can continue until only a five-clique of nodes remains. *Density scaling*, i.e. removing a row of five nodes, leaves the network's diameter unchanged, but decreases its density k until only a five-node, four-hop network remains. For all ten topologies obtained this way, we repeat the analysis from the previous section.

Results. For a given bus speed, end-to-end latency is practically unaffected by the network's diameter or density. As discussed in Section 6.2, such latency is, by design, virtually deterministic and effectively independent of traffic endpoints. Topology-induced differences in latency are hence on the same order as the granularity with which our testbed infrastructure can measure latency (i.e. 2.5 µs), rendering further analysis moot.

By contrast, reliability, and therefore goodput, are substantially affected by parameters of the network topology. Figure 13 provides an overview of goodput[5] for each of the network topologies under consideration. When scaling the testbed's diameter, goodput tends to decrease as the number of nodes increases, falling from 25.4 kbps in a five-clique to the 19 kbps observed in the previous section for a 25-node network. When scaling network density, however, goodput increases from 16.5 kbps to 19 kbps when going from a minimally to a maximally dense configuration. In neither scaling scenario does the presence of contention have a considerable impact.

Scaling network diameter. Figure 14 reveals why scaling network diameter decreases goodput by plotting frame delivery ratios in function of bus speed for varying diameter d and 128-bit frames. When increasing the number of hops in a network, the reliability curve established in the previous section shifts to the left. The highest bus speed setting at which the network operates reliably thus decreases. The goodput-optimal bus speed setting hence goes from 28.6 kbps to 20 kbps as d goes from 1 to 5, explaining most of the goodput loss. The inset demonstrates that adding nodes to the network also has a measurable effect on reliability that is independent of this left shifting, introducing roughly a percentage point of additional frame loss in our scenario.

[4] To be precise: the time it takes to transmit the three bits, plus $1.5s$ to indicate the end of a frame, plus w to enforce global symbol-synchronicity (cf. Section 4.4).

[5] Optimal goodput, as in Figure 12.

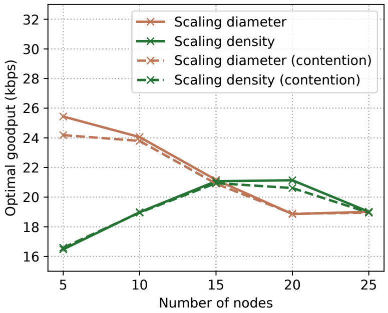

Figure 13: Effect of network topology size on goodput.

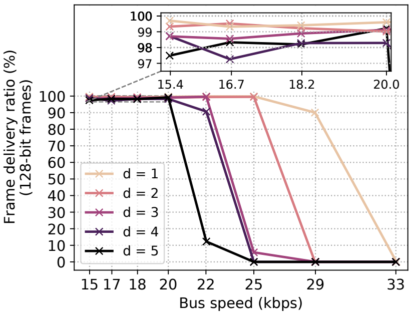

Figure 14: Effect of network diameter on reliability.

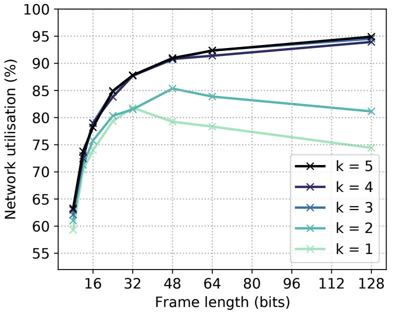

Figure 15: Effect of network density on network utilisation.

The root cause for the shifting of the reliability curve is introduced in Section 3.1. When adding hops to the network, a Zero-Wire node becomes exposed to interference by other nodes whose signals are further offset in time. To maintain symbol-synchronicity, nodes therefore need to rely on a longer symbol duration and hence a lower bus speed setting. In idealised circumstances, the time offset between signals emitted by neighbouring nodes corresponds to one control loop interval $i = 2.5\,\mu s$, making the curve shift to the left with one discrete speed setting (i.e. the minimal value for s/i for a reliable network increases by one). As shown in the figure, this means throughput capacity (i.e. optimal b, the point of network collapse in terms of bus speed) decreases sublinearly with increasing d, since b is inversely proportional to the number of samples per symbol (i.e. s/i)[6]. In a contention-free scenario, the need to decrease bus speed due to diameter scaling should eventually halt, since symbol-synchronicity is a local constraint (cf. Section 3.1). We do not yet observe this phenomenon in our four-hop testbed.

Scaling network density. Figure 15 illustrates why increasing network density improves goodput. For varying density k, the figure plots frame length against network utilisation, i.e. the ratio between observed goodput and the theoretical optimal bus speed b. For low density, Zero-Wire's reliability decreases more rapidly with increasing frame length. The net result is that, for $k = 1$ and $k = 2$, the tails of goodput curves are tilted downward relative to more dense settings. Maximum goodput is then achieved for frame lengths of 32 and 48 bits, far smaller than the 128-256 bits observed for sufficiently dense topologies. Network utilisation therefore tops out at around 80% instead of 95%. The optimal bus speed setting, used to obtain these results, varies between $b = 20$ kbps and $b = 22.2$ kbps without a clear trend. The apparent decrease in goodput shown in Figure 13 for density scaling from 20 to 25 nodes hence does not allow for generalised conclusions on network performance beyond a 25-node scenario: it is the result of a variation in bus speed ($b = 22.2$ to $b = 20$ kbps), which cannot be masked by the small marginal increase in network utilisation from $k = 4$ to $k = 5$.

Mutiple phenomena may explain the effect of network density on the relation between frame length and reliability. For example, synchronous intensity modulation by multiple nodes results in larger intensity variations than would be caused by any single node (i.e. signals "interfere constructively"). Additionally, networks with higher density exhibit a larger number of redundant paths: if a node fails to detect an edge in a signal, one of its neighbouring nodes may still detect and subsequently relay it to the former node, which masks the initial failure since nodes tolerate some slack on the arrival time of edges (cf. Section 4.4). Further analysis would be required to assess the relative importance of each of these phenomena. Such analysis should also examine to what extent the scaling trends observed in this section continue beyond a 25-node network.

6.4 Discussion

Performance envelope. While Zero-Wire's end-to-end latency is virtually deterministic and has a lower lower bound than conventional wireless mesh networks, the 99% reliability offered by the current prototype is also considerably lower than the five nines claimed for e.g. TSCH-based networks [28]. From a goodput perspective, Zero-Wire's performance is roughly on par with radio-based low-power wireless mesh networks that are not optimised for a specific traffic pattern [30], while the platform, at the cost of inhibiting co-existent traffic flows, provides a mechanism for contention resolution with no latency cost. To realise these novel possibilities, Zero-Wire frames may have to be considerably smaller than the already small frames found in current IoT networks.

The above comparisons are flawed, however. In mature networking technologies, techniques such as link-layer retransmission, fragmentation or error correction actively trade off certain performance metrics against others. A future protocol on top of Zero-Wire's low-level network architecture must explore these trade-offs, but is beyond the scope of this paper. Analogously, some applications may depend on more complex notions of latency and reliability than evaluated here (e.g. they consider multiple frames [40] or application-level queuing delay [88]). To determine the true benefits of symbol-synchronous transmission, an assessment of its performance advantages relative to a well-researched Glossy-like [33] OWC primitive would also have been instructive. Unfortunately, such a primitive does not exist to the best of our knowledge. Still, we believe our results point towards a platform that enables wireless interaction with remote devices that is more reminiscent of I^{2}C-like buses than of conventional wireless networks.

Range. Zero-Wire, like other OWC platforms [65], is a short-range technology: initial measurements indicate that the reliability of the current Zero-Wire prototype breaks down rapidly beyond 2 metres of (line-of-sight) range for a single pair of nodes, and that low bus speeds are more robust over longer distances. Quantifying range as a single number is somewhat problematic because of its relationship with transmission power, topology density (cf. Section 6.3) and directionality of the optical front-end. Moreover, the impact of a network's diameter — and thus that of a node's range

[6] Assuming the number of nodes scales as $O(d^2)$, such harmonic scaling of per-node throughput is consistent with well-known theoretical results for conventional wireless mesh networks [38], at the cost of prohibiting parallel traffic flows.

— on end-to-end performance (e.g. latency) may be considerably smaller than for conventional architectures. A complete investigation of Zero-Wire's transmission range therefore requires the design of low-power peripheral hardware and deployment scenarios fine-tuned to the type of network it introduces. In this paper, we have instead chosen to focus on exploring the merits of the platform's network architecture using existing hardware.

Transceiver implementation. The experiments in this section reveal the limitations of a software-defined transceiver on a microcontroller. Many of the phenomena that drive Zero-Wire's reliability and scaling behaviour are contingent on the number of intensity samples per symbol and the time it takes to process such a sample. With a control loop interval requiring roughly 500 single-cycle instructions at 200 MHz, that processing time currently cannot be reduced below 2.5 µs. Moreover, imperfections in the current implementation, such as processor stalls of the PRU implementing the transceiver, occasionally cause control loop iterations to finish late and may thus artificially lower reliability. In our vision, future versions of the Zero-Wire transceiver should therefore be implemented as a dedicated hardware module instead of a software-defined platform. Such an implementation would also allow for a fairer comparison with existing radio transceivers and enable more complex (de)modulation schemes to increase performance, since Zero-Wire's analogue front-end supports modulation speeds at least an order of magnitude higher than can now be exploited [12].

Visibility. In contrast to many contemporary OWC research efforts, this paper does not devote particular attention to issues pertaining to the *visibility* of light, such as flickering [14, 65], but focuses on architectural aspects of optical networks. If relevant, Zero-Wire's white LEDs and matching photodiodes provided by the DenseVLC platform can be replaced by ones that emit/detect light outside of the visible spectrum without further changes to the platform. Alternatively, a Zero-Wire deployment that configures one node to inject a steady stream of short low-priority frames avoids flickering issues altogether: at settings for s examined in this section, Manchester encoding guarantees $> 10^4$ pulses per second, whereas flickering becomes inconsequential to humans at a few hundred Hz [14].

7 RELATED WORK

This paper cross-cuts several active research topics. This section reviews its relation to deterministic networking, synchronous transmission, cut-through forwarding, and multi-hop OWC.

7.1 Deterministic networks

Protocol stacks such as 6TiSCH [87] and WirelessHART [69, 78] support deterministic behaviour in low-power wireless mesh networks by time-multiplexing the wireless medium and centrally scheduling traffic flow requirements onto contention-free time slots. These slots are grouped into *tracks*, which establish end-to-end paths with latency and reliability guarantees by incorporating link and network-layer redundancy [47]. Zero-Wire nodes explore a diametrically opposed approach: they save time by using networks as a whole instead of individual links, they resolve contention instead of avoiding it, and they actively seize the bus instead of using pre-established schedules. As such, they introduce additional guarantees (latency and jitter are practically constant across traffic flows), subsume traditional design considerations (e.g. routing buffer management [34]) and enable new performance settings (i.e. latency < 1 ms for small frames at 10s of microseconds of jitter).

Zero-Wire leverages several techniques that have long been of interest to the research community. BitMAC [74], for example, offers contention resolution by interpreting synchronised on-off keyed signals as a bitwise `or` among contending neighbours. Likewise, WiDom [66] implements priority-based arbitration. Due to their reliance on radio, however, both systems suffer from interference between concurrently transmitted signals, necessitating store-and-forward packet switching, routing and channel allocation [66–68, 74]. Like Zero-Wire, Zippy [81] propagates messages through a mesh without waiting for a complete frame to be received to provide near-deterministic and low latency. The latter system depends on an off-the-shelf OOK radio, using carrier frequency randomisation to mitigate inter-signal interference. Whereas Zero-Wire transceivers reason on a raw signal and relay symbols, Zippy nodes therefore adapt the sequence of symbols they transmit to relay bits, limiting throughput to 1.4 kbps. Optical buses have also been studied for more than two decades already: fibre-optic CAN serves as a resilient control network in electromagnetically disturbed settings, for example, near high-voltage equipment [24, 37]. Free-space optical networks may similarly function as a communication medium in harsh industrial conditions, where e.g. hot reflective surfaces may render radio links prohibitively unreliable [15].

7.2 Synchronous transmission

Synchronous Transmission (cf. Section 2.1) works in spite of interference between concurrently transmitted signals. Such interference can be dealt with in several ways: Chaos [49] leverages the FM capture effect; Glossy [33] benefits from 15.4's direct sequence spread spectrum technique [55]; the system in [56] profits from energy spreading of interfering LoRa symbols by inserting small time offsets. Zero-Wire introduces a new approach: it exploits the interference characteristics of intensity modulation by incoherent transmitters to eliminate store-and-forward packet switching, the implications of which are best observed in contrast with *Low-Power Wireless Bus (LWB)* [32]. LWB maintains schedules that construct a one-to-one mapping between a network's traffic flows and Glossy floods. LWB therefore sits halfway between Zero-Wire and deterministic 6TiSCH: it lowers latency relative to routing solutions, but does not allow for sub-millisecond end-to-end communication of very short frames with ad-hoc contention resolution.

In contrast to Zero-Wire, ST allows to exchange multiple frames in parallel. Though direct comparisons are flawed due to different deployment scenarios and underlying physical layers, several ST primitives report considerably higher goodput than Zero-Wire, either by exchanging information aggregated from multiple frames (Mixer [40], Codecast [61], Chaos [49]), or by optimising for a sequence of frames originating from the same initiator (Splash [25], P^3 [26], Pando [27]). These high-throughput primitives enable the quick delivery of many frames, yet the latency between any single frame's initial transmission and its delivery across multiple hops is in the order of 100s of milliseconds. These raw performance measurements do not paint a complete picture either: many ST systems incorporate aspects that may yet be built on top of Zero-Wire's low-level architecture. Examples include low-power event-driven wake-up [82] and explicit notions of network membership [32].

7.3 Cut-through forwarding

Recent advances in radio-based self-interference cancellation have enabled full-duplex radios that concurrently receive and relay on the same band [13]. On the networking side, these radios allow for *wireless in-band cut-through forwarding*, which reduces end-to-end latency by pipelining transmission and reception along forwarders in a routing path using amplify-and-forward or decode-and-forward schemes [20], thus breaking store-and-forward packet switching and avoiding MAC overhead between forwarders. The work presented in this paper differs by designing (de)modulation logic to tolerate interference by concurrent transmitters, rather than using an existing physical layer while trying to model and subtract such interference from the signal to be received. A key advantage of this paper's approach is reduced implementation complexity and improved scalability in terms of the number of concurrent forwarders: state-of-the-art cut-through systems rely on software-defined radios and limit the number of concurrent forwarders to five, without redundant links [20]. This limitation entails that these systems still route and perform medium access control across different *virtual* forwarders [20], i.e. groups of physical nodes, instead of implementing a bus-like network. A disadvantage is that Zero-Wire's approach does not trivially extend to arbitrary (radio-based) modulation schemes and only allows for network-wide broadcasts. Alternatively, work concurrent to that presented in this paper brings cut-through forwarding to the IoT by equipping nodes with two 15.4 radios that transmit and receive on *different* bands, effectively parallellising forwarding steps in a Glossy flood [41]. Compared to this paper, such an approach allows for larger ranges and higher raw throughput, but suffers from reduced reliability (65-85% vs. ~99%) due to inter-band interference and incurs a higher per-hop latency cost (~500 vs. ~2.5 µs), thus inhibiting bus-like operation.

7.4 Multi-hop optical wireless communication

The directionality of OWC aggravates the hidden node problem [45]: relative to omnidirectional radios, senders are less likely to correctly determine whether they should refrain from interfering with an ongoing transmission. In demanding scenarios, OWC standards (IEEE 802.15.7 [71]) may hence suffer from poor link-layer performance, even though their architecture closely resembles that of well-established radio-based technologies ([53], cf. IEEE 802.15.4 [5]). Directional signals are also prone to *blockage* [45]: standing in front of an LED completely stops its signal. Analogously, deployments of multi-hop networks must avoid *deafness* [45]: nodes that form a link should be oriented towards each other to maintain line of sight, though reflections may somewhat relax this constraint [39, 96]. VL-MAC [45] and VL-ROUTE [46] mitigate these effects through direction-dependent route reliability scores, link-layer handshakes, and *busy tones* [89], i.e. signals that reveal the presence of a hidden node in a link-layer neighbourhood while its transmission is ongoing. Zero-Wire takes the idea of a busy tone a step further by propagating it through a mesh network while embedding the information from the original transmission within it. The disadvantages of directionality hence become less relevant: a mesh network turns into a single, virtual, multi-directional transmitter.

Other systems, like Zero-Wire, consider directionality to be an advantage rather than a disadvantage because it trivially allows for full-duplex in-band communication. The cooperative relaying models proposed in [63, 96, 97, 99] relay an incoming frame before the end of its transmission to extend OWC range using amplify-and-forward or decode-and-forward schemes. Unlike Zero-Wire, however, these approaches do not generalise to arbitrary multi-hop networks, but remain limited to single-relay [63, 99], linear [96] and triangular topologies [97]. Concurrent relaying has been shown to enable CSMA/CD in [57], but this system's approach is restricted to star topologies and uses orthogonal channels to separate original from relayed signals. Other MAC protocols have also exploited the unique characteristics of IM/DD through symbol-level synchronisation, Manchester encoding and OOK [75, 76, 91], but with very different goals to Zero-Wire: these systems consider `off` symbols as intra-frame transmission slots to allow for link-layer full-duplex communication, or to use a single LED as both emitter and detector.

8 CONCLUSION AND OUTLOOK

This paper introduced the *symbol-synchronous bus*, a paradigm for free-space optical communication that provides end-to-end connectivity across a wireless mesh network and breaks conventional abstractions concerning store-and-forward packet switching, medium access control and routing. The Zero-Wire platform implements this paradigm; empirical results in a laboratory setting demonstrate how a Zero-Wire mesh attains performance characteristics and interaction patterns that, until now, have been considered impossible to deliver over wireless mesh networks. In particular, a first Zero-Wire prototype has been shown to provide deterministic sub-millisecond end-to-end latency for two-byte frames, priority-based contention resolution without collisions, 19 kbps of goodput, and 99% reliability. Nodes can access the network's resources at a very low barrier to entry: transmitting data through the network is as simple as blinking an LED at regular intervals.

Several research tracks identified in this paper may enable the improvement of our initial evaluation results. As Zero-Wire currently relies on an unoptimised hardware platform, our future work will naturally investigate performance improvements that might be obtained with tailor-made hardware for both relay nodes and peripherals. We also aim to characterise these platforms in terms of evaluation criteria that are not (fully) addressed in this paper, such as transmission range and power consumption. Future protocol design efforts should also reconcile Zero-Wire with duty-cycled low-power operation of peripherals, e.g. through light-based active wake-up [60], as well as explore how to best exploit the performance envelope unlocked by it to enable a novel class of wireless embedded applications. In addition, these efforts should examine what security guarantees can be offered on symbol-synchronous buses, since their lack of store-and-forward mechanisms renders frame-based authentication schemes ineffective.

ACKNOWLEDGMENTS

This research is partially funded by the Research Fund KU Leuven, Jonathan Oostvogels' PhD fellowship (11H7921N) of the Research Foundation – Flanders (FWO), and the FWO D3-CPS project. The authors would like to thank Jona Beysens and Qing Wang for their support regarding the DenseVLC hardware platform, as well as the anonymous shepherd and reviewers for their detailed feedback.

REFERENCES

[1] [n.d.]. BeagleBone Black. https://beagleboard.org/black. Accessed: 2020-04-01.

[2] [n.d.]. InfluxDB. https://www.influxdata.com. Accessed: 2020-05-26.

[3] [n.d.]. RPMsg Quick Start Guide. https://processors.wiki.ti.com/index.php/RPMsg_Quick_Start_Guide. Accessed: 2020-03-06.

[4] [n.d.]. Visible Light Communication Testbed. https://www.esat.kuleuven.be/telemic/research/NetworkedSystems/infrastructure/IoT-Lab/vlc-lab/visible-light-communication-testbed. Accessed: 2020-07-10.

[5] 2016. IEEE Standard for Low-Rate Wireless Networks. *IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011)* (April 2016), 1–709. https://doi.org/10.1109/IEEESTD.2016.7460875

[6] 2016. *IEEE Std 802.15.4-2015 (revision of IEEE Std 802.15.4-2011): IEEE standard for low-rate wireless personal area networks (WPANs).* IEEE, 3 Park Avenue, New York, NY 10016-5997, USA.

[7] David J Acunzo, Graham MacKenzie, and Mark CW van Rossum. 2012. Systematic biases in early ERP and ERF components as a result of high-pass filtering. *Journal of neuroscience methods* 209, 1 (2012), 212–218.

[8] Ferran Adelantado, Xavier Vilajosana, Pere Tuset-Peiro, Borja Martinez, Joan Melia-Segui, and Thomas Watteyne. 2017. Understanding the limits of LoRaWAN. *IEEE Communications magazine* 55, 9 (2017), 34–40.

[9] LoRa Alliance. 2017. *LoRaWAN specification v1.1.* Technical Report. https://lora-alliance.org/sites/default/files/2018-04/lorawantm_specification_-v1.1.pdf. Accessed: 2020-14-01.

[10] A. N. Alvi, S. S. Naqvi, S. H. Bouk, N. Javaid, U. Qasim, and Z. A. Khan. 2012. Evaluation of Slotted CSMA/CA of IEEE 802.15.4. In *2012 Seventh International Conference on Broadband, Wireless Computing, Communication and Applications.* 391–396. https://doi.org/10.1109/BWCCA.2012.69

[11] Dan Awtrey and Dallas Semiconductor. 1997. Transmitting data and power over a one-wire bus. *Sensors-The Journal of Applied Sensing Technology* 14, 2 (1997), 48–51.

[12] Jona Beysens, Ander Galisteo, Qing Wang, Diego Juara, Domenico Giustiniano, and Sofie Pollin. 2018. DenseVLC: A Cell-Free Massive MIMO System with Distributed LEDs. In *Proceedings of the 14th International Conference on Emerging Networking EXperiments and Technologies (CoNEXT '18).* Association for Computing Machinery, New York, NY, USA, 320–332. https://doi.org/10.1145/3281411.3281423

[13] Dinesh Bharadia and Sachin Katti. 2014. FastForward: Fast and constructive full duplex relays. *ACM SIGCOMM Computer Communication Review* 44, 4 (2014), 199–210.

[14] Rens Bloom, Marco Zúñiga Zamalloa, and Chaitra Pai. 2019. LuxLink: Creating a Wireless Link from Ambient Light. In *Proceedings of the 17th Conference on Embedded Networked Sensor Systems (SenSys '19).* Association for Computing Machinery, New York, NY, USA, 166–178. https://doi.org/10.1145/3356250.3360021

[15] Abhishek Borkar and Prabhat Ranjan. 2011. Optical wireless sensor network design for a conducting chamber. In *2011 IEEE 36th Conference on Local Computer Networks.* IEEE, 990–993.

[16] Marco Cattani, Andreas Loukas, Marco Zimmerling, Marco Zuniga, and Koen Langendoen. 2016. Staffetta: Smart duty-cycling for opportunistic data collection. In *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM.* 56–69.

[17] Nan Cen, Jithin Jagannath, Simone Moretti, Zhangyu Guan, and Tommaso Melodia. 2019. LANET: Visible-light ad hoc networks. *Ad Hoc Networks* 84 (2019), 107–123.

[18] Gianluca Cena and Adriano Valenzano. 2002. A multistage hierarchical distributed arbitration technique for priority-based real-time communication systems. *IEEE Transactions on Industrial Electronics* 49, 6 (2002), 1227–1239.

[19] Tengfei Chang, Thomas Watteyne, Qin Wang, and Xavier Vilajosana. 2016. LLSF: Low latency scheduling function for 6TiSCH networks. In *2016 International Conference on Distributed Computing in Sensor Systems (DCOSS).* IEEE, 93–95.

[20] Bo Chen, Yue Qiao, Ouyang Zhang, and Kannan Srinivasan. 2015. AirExpress: Enabling seamless in-band wireless multi-hop transmission. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking.* 566–577.

[21] Nikolaus Correll, Prabal Dutta, Richard Han, and Kristofer Pister. 2017. Wireless Robotic Materials. In *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems (SenSys '17).* Association for Computing Machinery, New York, NY, USA, Article Article 24, 6 pages. https://doi.org/10.1145/3131672.3131702

[22] Conrad Dandelski, Bernd-Ludwig Wenning, Daniel Viramontes Perez, Dirk Pesch, and Jean-Paul MG Linnartz. 2015. Scalability of dense wireless lighting control networks. *IEEE Communications Magazine* 53, 1 (2015), 157–165.

[23] Yuanbo Deng and Daping Chu. 2017. Coherence properties of different light sources and their effect on the image sharpness and speckle of holographic displays. *Scientific Reports* 7 (12 2017). https://doi.org/10.1038/s41598-017-06215-x

[24] Marco Di Natale, Haibo Zeng, Paolo Giusto, and Arkadeb Ghosal. 2012. *Understanding and using the controller area network communication protocol: theory and practice.* Springer Science & Business Media.

[25] Manjunath Doddavenkatappa, Mun Choon Chan, and Ben Leong. 2013. Splash: Fast data dissemination with constructive interference in wireless sensor networks. In *Presented as part of the 10th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 13).* 269–282.

[26] Manjunath Doddavenkatappa and Mun Choon. 2014. P^3: a practical packet pipeline using synchronous transmissions for wireless sensor networks. In *IPSN-14 Proceedings of the 13th International Symposium on Information Processing in Sensor Networks.* IEEE, 203–214.

[27] Wan Du, Jansen Christian Liando, Huanle Zhang, and Mo Li. 2015. When pipelines meet fountain: Fast data dissemination in wireless sensor networks. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems.* 365–378.

[28] D. Dujovne, T. Watteyne, X. Vilajosana, and P. Thubert. 2014. 6TiSCH: deterministic IP-enabled industrial Internet (of Things). *IEEE Communications Magazine* 52, 12 (December 2014), 36–41. https://doi.org/10.1109/MCOM.2014.6979984

[29] Simon Duquennoy, Olaf Landsiedel, and Thiemo Voigt. 2013. Let the tree bloom: Scalable opportunistic routing with ORPL. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems.* ACM, 2.

[30] Simon Duquennoy, Fredrik Österlind, and Adam Dunkels. 2011. Lossy links, low power, high throughput. In *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems.* 12–25.

[31] Mohammad Farsi, Karl Ratcliff, and Manuel Barbosa. 1999. An overview of controller area network. *Computing & Control Engineering Journal* 10, 3 (1999), 113–120.

[32] Federico Ferrari, Marco Zimmerling, Luca Mottola, and Lothar Thiele. 2012. Low-power wireless bus. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems.* ACM, 1–14.

[33] F. Ferrari, M. Zimmerling, L. Thiele, and O. Saukh. 2011. Efficient network flooding and time synchronization with Glossy. In *Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks.* 73–84.

[34] Norman Finn. 2018. Introduction to time-sensitive networking. *IEEE Communications Standards Magazine* 2, 2 (2018), 22–28.

[35] Ander Galisteo, Diego Juara, and Domenico Giustiniano. 2019. Research in visible light communication systems with OpenVLC1. 3. In *2019 IEEE 5th World Forum on Internet of Things (WF-IoT).* IEEE, 539–544.

[36] Zabih Ghassemlooy, Shlomi Arnon, Murat Uysal, Zhengyuan Xu, and Julian Cheng. 2015. Emerging optical wireless communications-advances and challenges. *IEEE journal on selected areas in communications* 33, 9 (2015), 1738–1749.

[37] G Gräwer and W Heinze. 1997. *Using a fiber optic CAN bus for the proton source control of the CERN PS-Linac.* Technical Report.

[38] Piyush Gupta and Panganmala R Kumar. 2000. The capacity of wireless networks. *IEEE Transactions on information theory* 46, 2 (2000), 388–404.

[39] Navid Hamedazimi, Zafar Qazi, Himanshu Gupta, Vyas Sekar, Samir R Das, Jon P Longtin, Himanshu Shah, and Ashish Tanwer. 2014. FireFly: A reconfigurable wireless data center fabric using free-space optics. In *ACM SIGCOMM Computer Communication Review*, Vol. 44. ACM, 319–330.

[40] Carsten Herrmann, Fabian Mager, and Marco Zimmerling. 2018. Mixer: efficient many-to-all broadcast in dynamic wireless mesh networks. In *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems.* ACM, 145–158.

[41] Nicolas Himmelmann, Dingwen Yuan, Lars Almon, and Matthias Hollick. 2020. Concurrent Wireless Cut-Through Forwarding: Ultra-Low Latency Multi-Hop Communication for the Internet of Things. In *2020 International Conference on Distributed Computing in Sensor Systems (DCOSS).* IEEE.

[42] Jens Horneber and Anton Hergenröder. 2014. A survey on testbeds and experimentation environments for wireless sensor networks. *IEEE Communications Surveys & Tutorials* 16, 4 (2014), 1820–1838.

[43] Qingqing Huang, Baoping Tang, and Lei Deng. 2015. Development of high synchronous acquisition accuracy wireless sensor network for machine vibration monitoring. *Measurement* 66 (2015), 35–44.

[44] Romain Jacob, Jonas Baechli, Reto Da Forno, and Lothar Thiele. 2019. Synchronous Transmissions made easy: Design your network stack with Baloo. In *16th International Conference on Embedded Wireless Systems and Networks (EWSN 2019).*

[45] Jithin Jagannath and Tommaso Melodia. 2018. An opportunistic medium access control protocol for visible light ad hoc networks. In *2018 International Conference on Computing, Networking and Communications (ICNC).* IEEE, 609–614.

[46] Jithin Jagannath and Tommaso Melodia. 2019. VL-ROUTE: A cross-layer routing protocol for visible light ad hoc network. *CoRR* abs/1904.05177 (2019). arXiv:1904.05177 http://arxiv.org/abs/1904.05177

[47] Abdulkadir Karaagac, Jetmir Haxhibeqiri, Ingrid Moerman, and Jeroen Hoebeke. 2018. Time-critical communication in 6TiSCH networks. In *2018 IEEE Wireless Communications and Networking Conference Workshops (WCNCW).* IEEE, 161–166.

[48] V. Kawadia and P. R. Kumar. 2005. Principles and protocols for power control in wireless ad hoc networks. *IEEE Journal on Selected Areas in Communications* 23, 1 (Jan 2005), 76–88. https://doi.org/10.1109/JSAC.2004.837354

[49] Olaf Landsiedel, Federico Ferrari, and Marco Zimmerling. 2013. 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*. ACM, 1.

[50] Alleyne Leach. 1994. Profibus: the German fieldbus standard. *Assembly automation* 14, 1 (1994), 8–12.

[51] F. Leens. 2009. An introduction to I2C and SPI protocols. *IEEE Instrumentation Measurement Magazine* 12, 1 (February 2009), 8–13. https://doi.org/10.1109/MIM.2009.4762946

[52] Nickolaus E. Leggett. 2004. Wireless bus. US Patent 6,771,9353.

[53] Carlos Ley-Bosch, Itziar Alonso-González, David Sánchez-Rodríguez, and Carlos Ramírez-Casañas. 2016. Evaluation of the effects of hidden node problems in IEEE 802.15.7 uplink performance. *Sensors* 16, 2 (2016), 216.

[54] Qijie Liang, Xiaoqin Yan, Xinqin Liao, Shiyao Cao, Shengnan Lu, Xin Zheng, and Yue Zhang. 2015. Integrated active sensor system for real time vibration monitoring. *Scientific reports* 5 (2015), 16063.

[55] Chun-Hao Liao, Yuki Katsumata, Makoto Suzuki, and Hiroyuki Morikawa. 2016. Revisiting the so-called constructive interference in concurrent transmission. In *2016 IEEE 41st Conference on Local Computer Networks (LCN)*. IEEE, 280–288.

[56] Chun-Hao Liao, Guibing Zhu, Daiki Kuwabara, Makoto Suzuki, and Hiroyuki Morikawa. 2017. Multi-hop LoRa networks enabled by concurrent transmission. *IEEE Access* 5 (2017), 21430–21446.

[57] KIX Lin and K Hirohashi. 2009. High-speed full-duplex multiaccess system for LED-based wireless communications using visible light. In *Proc of the International Symposium on Optical Engineering and Photonic Technology (OEPT)*. 1–5.

[58] Fabian Mager, Dominik Baumann, Romain Jacob, Lothar Thiele, Sebastian Trimpe, and Marco Zimmerling. 2019. Feedback control goes wireless: Guaranteed stability over low-power multi-hop networks. In *Proceedings of the 10th ACM/IEEE International Conference on Cyber-Physical Systems*. 97–108.

[59] Luiz M Matheus, Alex B Vieira, Marcos AM Vieira, and Luiz FM Vieira. 2019. DYRP-VLC: A dynamic routing protocol for wireless ad-hoc visible light communication networks. *Ad Hoc Networks* 94 (2019), 101941.

[60] Maxim Integrated 2010. *Wake up and hear the IR*. Maxim Integrated. Application Note 4467. https://www.maximintegrated.com/en/design/technical-documents/app-notes/4/4467.html. Accessed: 2020-04-01.

[61] Mobashir Mohammad and Mun Choon Chan. 2018. Codecast: supporting data driven in-network processing for low-power wireless sensor networks. In *2018 17th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 72–83.

[62] Masanobu Morishita. 1989. Radio bus system. US Patent 4,866,733.

[63] Omer Narmanlioglu, Refik Caglar Kizilirmak, Farshad Miramirkhani, and Murat Uysal. 2017. Cooperative visible light communications with full-duplex relaying. *IEEE Photonics Journal* 9, 3 (2017), 1–11.

[64] Zhibo Pang, Michele Luvisotto, and Dacfey Dzung. 2017. Wireless high-performance communications: The challenges and opportunities of a new target. *IEEE Industrial Electronics Magazine* 11, 3 (2017), 20–25.

[65] Parth H Pathak, Xiaotao Feng, Pengfei Hu, and Prasant Mohapatra. 2015. Visible light communication, networking, and sensing: a survey, potential and challenges. *IEEE communications surveys & tutorials* 17, 4 (2015), 2047–2077.

[66] Nuno Pereira, Bjrn Andersson, and Eduardo Tovar. 2007. WiDom: A dominance protocol for wireless medium access. *IEEE Transactions on Industrial Informatics* 3, 2 (2007), 120–130.

[67] Nuno Pereira, Bjorn Andersson, Eduardo Tovar, and Anthony Rowe. 2007. Static-priority scheduling over wireless networks with multiple broadcast domains. In *28th IEEE International Real-Time Systems Symposium (RTSS 2007)*. IEEE, 447–458.

[68] Nuno Pereira, Ricardo Gomes, Björn Andersson, and Eduardo Tovar. 2009. Efficient aggregate computations in large-scale dense WSN. In *2009 15th IEEE Real-Time and Embedded Technology and Applications Symposium*. IEEE, 317–326.

[69] Stig Petersen and Simon Carlsen. 2011. WirelessHART vs. ISA100. 11a: The format war hits the factory floor. (2011).

[70] Michael Rahaim and Thomas DC Little. 2017. Interference in IM/DD optical wireless communication networks. *IEEE/OSA Journal of Optical Communications and Networking* 9, 9 (2017), D51–D63.

[71] S. Rajagopal, R. D. Roberts, and S. Lim. 2012. IEEE 802.15.7 visible light communication: modulation schemes and dimming support. *IEEE Communications Magazine* 50, 3 (March 2012), 72–82. https://doi.org/10.1109/MCOM.2012.6163585

[72] Bhaskaran Raman, Kameswari Chebrolu, Sagar Bijwe, and Vijay Gabale. 2010. PIP: A connection-oriented, multi-hop, multi-channel TDMA-based MAC for high throughput bulk transfer. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*. 15–28.

[73] JA Richmond. 1998. Spies in ancient Greece. *Greece & Rome* 45, 1 (1998), 1–18.

[74] Matthias Ringwald and Kay Römer. 2005. BitMAC: a deterministic, collision-free, and robust MAC protocol for sensor networks.. In *EWSN*. 57–69.

[75] Stefan Schmid, Giorgio Corbellini, Stefan Mangold, and Thomas R Gross. 2013. LED-to-LED visible light communication networks. In *Proceedings of the fourteenth ACM international symposium on Mobile ad hoc networking and computing*. ACM, 1–10.

[76] S. Schmid, G. Corbellini, S. Mangold, and T. R. Gross. 2014. Continuous synchronization for LED-to-LED visible light communication networks. In *2014 3rd International Workshop in Optical Wireless Communications (IWOW)*. 45–49. https://doi.org/10.1109/IWOW.2014.6950774

[77] Meryem Simsek, Adnan Aijaz, Mischa Dohler, Joachim Sachs, and Gerhard Fettweis. 2016. 5G-enabled tactile internet. *IEEE Journal on Selected Areas in Communications* 34, 3 (2016), 460–473.

[78] Jianping Song, Song Han, Al Mok, Deji Chen, Mike Lucas, Mark Nixon, and Wally Pratt. 2008. WirelessHART: Applying wireless technology in real-time industrial process control. In *2008 IEEE Real-Time and Embedded Technology and Applications Symposium*. IEEE, 377–386.

[79] ISO Standard. 1993. Iso 11898, 1993. *Road vehicles–interchange of digital information–Controller Area Network (CAN) for high-speed communication* (1993).

[80] STMicroelectronics. [n.d.]. STM32G474xB STM32G474xC STM32G474xE. https://www.st.com/resource/en/datasheet/stm32g474cb.pdf. Accessed: 2020-04-01.

[81] Felix Sutton, Bernhard Buchli, Jan Beutel, and Lothar Thiele. 2015. Zippy: On-demand network flooding. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*. 45–58.

[82] Felix Sutton, Reto Da Forno, Jan Beutel, and Lothar Thiele. 2017. Blitz: A network architecture for low latency and energy-efficient event-triggered wireless communication. In *Proceedings of the 4th ACM Workshop on Hot Topics in Wireless*. 55–59.

[83] Yuichi Tanaka, Shinichiro Haruyama, and Masao Nakagawa. 2000. Wireless optical transmissions with white colored LED for wireless home links. In *11th IEEE International Symposium on Personal Indoor and Mobile Radio Communications. PIMRC 2000. Proceedings (Cat. No. 00TH8525)*, Vol. 2. IEEE, 1325–1329.

[84] Andrew S Tanenbaum et al. 2003. *Computer networks, 4th edition*. Prentice Hall.

[85] Texas Instruments 2013. *AM335x PRU-ICSS Reference Guide*. Texas Instruments. https://elinux.org/images/d/da/Am335xPruReferenceGuide.pdf. Accessed: 2020-01-08.

[86] Texas Instruments 2019. *PRU-ICSS / PRU_ICSSG Getting Starting Guide on Linux*. Texas Instruments. Application report. http://www.ti.com/lit/an/sprace9a/sprace9a.pdf. Accessed: 2020-01-08.

[87] Pascal Thubert. 2019. *An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4*. Internet-Draft draft-ietf-6tisch-architecture-28. IETF Secretariat. http://www.ietf.org/internet-drafts/draft-ietf-6tisch-architecture-28.txt Accessed: 2020-06-04.

[88] Ken Tindell, Alan Burns, and Andy J Wellings. 1995. Calculating controller area network (CAN) message response times. *Control Engineering Practice* 3, 8 (1995), 1163–1169.

[89] Fouad Tobagi and Leonard Kleinrock. 1975. Packet switching in radio channels: Part II-The hidden terminal problem in carrier sense multiple-access and the busy-tone solution. *IEEE Transactions on communications* 23, 12 (1975), 1417–1433.

[90] E. Tovar and F. Vasques. 1999. Real-time fieldbus communications using Profibus networks. *IEEE Transactions on Industrial Electronics* 46, 6 (Dec 1999), 1241–1251. https://doi.org/10.1109/41.808018

[91] Qing Wang and Domenico Giustiniano. 2014. Communication networks of visible light emitting diodes with intra-frame bidirectional transmission. In *Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies*. ACM, 21–28.

[92] Thomas Watteyne, Joy Weiss, Lance Doherty, and Jonathan Simon. 2015. Industrial IEEE802. 15.4e networks: Performance and trade-offs. In *2015 IEEE International Conference on Communications (ICC)*. IEEE, 604–609.

[93] Matthew Weiner, Milos Jorgovanovic, Anant Sahai, and Borivoje Nikolié. 2014. Design of a low-latency, high-reliability wireless communication system for control applications. In *2014 IEEE International conference on communications (ICC)*. IEEE, 3829–3835.

[94] Matthias Wilhelm, Vincent Lenders, and Jens B Schmitt. 2014. On the reception of concurrent transmissions in wireless sensor networks. *IEEE Transactions on Wireless Communications* 13, 12 (2014), 6756–6767.

[95] Andreas Willig, Kirsten Matheus, and Adam Wolisz. 2005. Wireless technology in industrial networks. *Proc. IEEE* 93, 6 (2005), 1130–1151.

[96] Hongming Yang and Ashish Pandharipande. 2013. Full-duplex relay VLC in LED lighting linear system topology. In *IECON 2013-39th Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 6075–6080.

[97] Hongming Yang and Ashish Pandharipande. 2014. Full-duplex relay VLC in LED lighting triangular system topology. In *2014 6th international symposium on communications, control and signal processing (ISCCSP)*. IEEE, 85–88.

[98] S. Yin and O. Gnawali. 2016. Towards embedded visible light communication robust to dynamic ambient light. In *2016 IEEE Global Communications Conference*

(GLOBECOM). 1–6. https://doi.org/10.1109/GLOCOM.2016.7842344

[99] Chao Zhang, Jia Ye, Gaofeng Pan, and Zhiguo Ding. 2018. Cooperative hybrid VLC-RF systems with spatially random terminals. *IEEE Transactions on Communications* 66, 12 (2018), 6396–6408.

[100] Marco Zimmerling, Luca Mottola, and Silvia Santini. 2020. Synchronous transmissions in low-power wireless: A survey of communication protocols and network services. *arXiv preprint arXiv:2001.08557* (2020).