Computer Networks 97 (2016) 74–87

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Firefly synchronization with phase rate equalization and its experimental analysis in wireless systems

G. Brandner[b,*], U. Schilcher[a], C. Bettstetter[a,b]

[a] University of Klagenfurt, Lakeside B02, Klagenfurt 9020, Austria
[b] Lakeside Labs GmbH, Lakeside B04, Klagenfurt 9020, Austria

ARTICLE INFO

Article history:
Received 24 June 2015
Revised 30 December 2015
Accepted 4 January 2016
Available online 12 January 2016

Keywords:
Synchronization
Pulse-coupled oscillators
Firefly synchronization
Self-organization
Phase rate equalization
Programmable radio

ABSTRACT

The convergence and precision of synchronization algorithms based on the theory of pulse-coupled oscillators is evaluated on programmable radios. Measurements in different wireless topologies show that such algorithms reach precisions in the low microsecond range. Based on the observation that phase rate deviation among radios is a limiting factor for the achievable precision, we propose a distributed algorithm for automatic phase rate equalization and show by experiments that an improved precision below one microsecond is possible in the given setups. It is also experimentally demonstrated that the stochastic nature of coupling is a key ingredient for convergence to synchrony. The proposed scheme can be applied in wireless systems for distributed synchronization of transmission slots, or sleep cycles.

1. Introduction

There is a broad spectrum of work on pulse-coupled oscillators (PCO) to model synchronization phenomena in biology, physics, and other sciences (see [1–7] and references therein). A prominent example is swarms of fireflies that synchronize their blinking behavior [8]. The beauty of these synchronization phenomena lies in the fact that system-wide synchrony emerges among the participating entities in a completely distributed, self-organizing manner without any need for central entities. Furthermore, PCO synchronization — sometimes called firefly synchronization — is scalable with respect to the number of entities and robust against full failure of individual entities or appearance of new entities.

Many communication protocols and scheduling techniques, as well as novel approaches such as interference alignment in wireless networks, require synchronization, e.g., [9–12]. Thus, the telecommunications engineering community has been interested to transfer the concepts behind these natural synchronization phenomena to design algorithms for the synchronization of nodes in wireless networks [13]. A one-to-one transfer is, however, infeasible due to the differences between biological and wireless communication systems. Several extensions and modifications are required with respect to delays, noise, and multihop communications, to mention a few (see [13–20]).

Despite the conceptual and theoretical advances in the design of PCO synchronization for wireless systems, real-world performance studies and proofs of concepts are largely missing. There only exist a few implementations on low-cost sensor platforms (see [16,21,22]), whose results are of interest, but whose synchronization precision is limited by restricted hardware capabilities.

* Corresponding author. Tel.: +43 6766835035.
E-mail addresses: guenther.brandner@me.com (G. Brandner), udo.schilcher@aau.at (U. Schilcher), christian.bettstetter@aau.at (C. Bettstetter).

http://dx.doi.org/10.1016/j.comnet.2016.01.001

To the best of our knowledge, there is no comprehensive performance analysis of PCO synchronization on programmable radio boards. The paper at hand addresses this issue. We analyze three recently proposed PCO synchronization algorithms by implementing them using field programmable gate-array (FPGA)-based radios and study their performance with respect to the achieved synchronization precision. Besides this experimental contribution, our main conceptual contribution comes from the *lessons learned* during our measurements: we propose an automatic phase rate equalization algorithm, integrate it into PCO synchronization, and show by experiments that this new feature significantly improves the synchronization precision compared to existing PCO algorithms.

The contributions can be summarized as follows:

- Providing a proof-of-concept for three PCO synchronization algorithms on FPGA-based radios.
- Analyzing and comparing the synchronization precision of these algorithms by means of real-world measurements in different network topologies, namely fully connected, star, line, and ring topology with five nodes.
- Showing that the synchronization precisions are in the low μs range in these topologies, and that the key factor preventing better precisions are phase rate deviations among the radios.
- Proposing a new distributed algorithm to additionally synchronize phase rates and showing by measurements that this algorithm used with PCO synchronization achieves precisions below one μs.
- Investigating unreliable or intentionally stochastic communication of synchronization words.

Our work is the most comprehensive experimental performance study of PCO algorithms on programmable radios over real wireless channels. It enables us to state that the following building blocks are essential for PCO synchronization in wireless networks and should be considered by protocol designers: a combination of positive (excitatory) and negative (inhibitory) coupling, unreliable or intentionally stochastic communication of synchronization words, and automatic phase rate equalization.

The remainder of this paper is as follows: Section 2 discusses related work. Section 3 reviews and explains three PCO algorithms for wireless systems. Section 4 addresses the implementation of these algorithms on programmable radios. Section 5 presents an experimental analysis of the algorithms in terms of their convergence and precision. It demonstrates the importance of the stochastic nature in the exchange of synchronization words for convergence and shows that precisions in the order of some ten microseconds are possible. Section 6 proposes and analyzes a distributed phase rate equalization algorithm to further improve the precision. Section 7 presents a performance analysis of synchronization with phase rate equalization and shows that precisions below one microsecond are possible.

In our prior work [23], we have applied phase rate equalization manually by measuring the phase rate of each device and programming static phase rate equalization factors. The paper at hand presents a completely distributed algorithm, which achieves phase rate synchronization among the devices during operation. This is particularly important since phase rates may vary when environmental factors, e.g., temperature, change. Our new algorithm adapts to these changes and hence enables its practical application. Furthermore, the experimental analysis of this paper is much more comprehensive than that of Brandner et al. [23].

2. Related work

Research on PCO synchronization in wireless systems can be divided into conceptual and analytical work (algorithms are proposed and theoretically analyzed) and experimental work (algorithms are implemented in testbeds to evaluate and verify their performance).

2.1. Concepts and analytical work

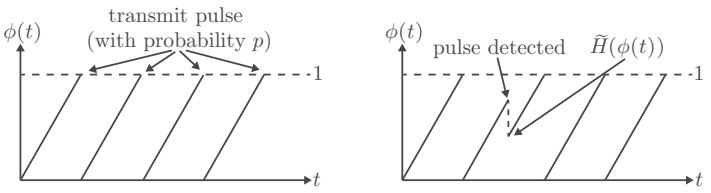
The mathematical modeling of pulse-coupled biological oscillators, as proposed, e.g., in [1] inspired by Peskin [8], offers a fully distributed and scalable approach for time synchronization with a broad set of applications (see, e.g., [2–7,24–26] and references therein).

The fact that technical limitations hinder a one-to-one transfer of these models to wireless systems led to papers investigating necessary changes for applying these models: Mathar and Mattfeldt [13] present extensions to apply PCO synchronization in TDMA systems. They show, for two oscillators, that synchronization is reached even in the presence of delays. Hong and Scaglione [14] propose a distributed PCO synchronization protocol for wireless networks considering pulse detection and refractory periods. Lucarelli and Wang [15] present synchronization protocols for dense, large-scale sensor networks and show that convergence to a synchronized state is reached even when the communication topology is time varying. Klinglmayr et al. [7] present a PCO synchronization algorithm with inhibitory and excitatory coupling and stochastic pulse emission. They prove that arbitrary networks of pulse-coupled oscillators converge almost surely, i.e., with probability one.

Further manuscripts dealing with the applicability of PCO synchronization in wireless networks are [17,22,27–30].

2.2. Experimental work

Comprehensive experimental performance studies and proofs of concepts of PCO synchronization are largely missing. There only exist a few implementations on low-cost wireless sensor platforms: Werner-Allen et al. [16] implement a PCO synchronization algorithm on TinyOS-based motes. They reach synchronization precisions of about 100 μs. Leidenfrost and Elmenreich [21] implement a PCO synchronization algorithm on ZigBee nodes. The evaluation is performed in terms of time to synchronization and synchronization precision. The 50% quantile of the synchronization precision is at about 700 μs. Pagliari and Scaglione [22] propose and implement a PCO synchronization algorithm on MicaZ nodes. The reported synchronization precision is in the range of a few hundred microseconds.



(a) no pulse is detected (b) pulse is detected

Fig. 1. Pulse-coupled oscillator synchronization.

The authors of An et al. [31] propose and analyze a PCO model for oscillators with non-identical frequencies. They perform simulations and experiments on a wireless sensor network testbed, consisting of 15 MICA2-compatible nodes. The reported synchronization precision achieved in the testbed is at around 100 µs.

All of the aforementioned implementations are based on low-cost devices. In contrast to this, we investigate and compare various PCO synchronization algorithms on FPGA-based radios and show that the synchronization precisions can be improved dramatically when implementing the algorithms directly in the physical layer.

3. Algorithms

Let us review three PCO synchronization algorithms; their performance will be evaluated later. The algorithms are: (i) synchronization by Pagliari and Scaglione (PS) [22,32], (ii) synchronization by Wang and Doyle III (WD) [33], and (iii) synchronization with inhibitory and excitatory coupling with stochastic pulse emission (IES) [7].

The basic procedure common to all three PCO algorithms is shown in Fig. 1: the oscillator's phase ϕ linearly increases from zero to one. When ϕ reaches one, it is reset, and a synchronization pulse is emitted (Fig. 1(a)). This pulse is emitted either always or with probability $p < 1$ depending on the algorithm. When receiving a pulse from another oscillator, the oscillator adjusts its own phase by jumping to the phase value determined by the update function $\widetilde{H}(\phi)$ (Fig. 1(b)).

The following notation is used: the absolute time is called t. The period τ_{ij} denotes the delay in seconds between oscillator i and oscillator j, i.e., the time it takes from the start of a pulse at i until it is processed at j. Let $\tau_{\min}$, $\tau_{\max}$, and $\overline{\tau}$ denote the minimum, maximum, and mean values of all delays, respectively. Furthermore, $\phi(t) \in [0, 1]$ is an oscillator's phase at time t, and $\phi(t^+)$ its phase infinitely short after t. The term δ denotes the cycle duration, i.e., the time it takes for an oscillator to increase its phase from zero to one. The function $f_\phi(\Delta t)$ maps a time Δt to the corresponding phase ϕ of an oscillator. We have $f_\phi(\Delta t) = \Delta t/\delta$ for $0 \le \Delta t \le \delta$.

Algorithm 1 specifies the three algorithms. For PS and WD the algorithm is the same, except for the update function. As infinitely short pulses are inappropriate for wireless systems, we send entire data packets with embedded synchronization words. This also enables synchronization concurrently with data transmission [17]. Details on packet length and structure are given in Section 4. Delays between sender and receiver are taken into account using an *auxiliary function*

$$H_{\mathcal{X}}(\phi) = \widetilde{H}_{\mathcal{X}}(\phi - f_\phi(\tau_{\min}) \bmod 1) + f_\phi(\tau_{\min}) \bmod 1, \quad (1)$$

where $\mathcal{X} \in \{\text{PS}, \text{WD}, \text{IES}\}$.

A requirement imposed on all update functions $\widetilde{H}_{\mathcal{X}}$ is that they are bounded in [0, 1]:

$$\phi \in [0, 1] \Rightarrow 0 \le \widetilde{H}_{\mathcal{X}}(\phi) \le 1. \quad (2)$$

Furthermore, to avoid echoing effects due to delays, a refractory interval $[0, \phi_{\text{ref}}]$ is used [34]. If an oscillator detects a synchronization word within this refractory interval, it will not adjust its phase. We employ $\phi_{\text{ref}} = f_\phi(2\tau_{\max} - \tau_{\min})$ in all algorithms, conceptually as in [35].

The three functions $\widetilde{H}_{\mathcal{X}}(\phi)$ are specified as follows:

- For PS we use

$$\widetilde{H}_{\text{PS}}(\phi) = \begin{cases} \phi & \text{if } \phi \le \phi_{\text{ref}}, \\ \min[1, a_1\phi + a_0] & \text{else,} \end{cases}$$

  where $a_1 = \exp(b\varepsilon)$ and $a_0 = \frac{\exp(b\varepsilon)-1}{\exp(b)-1}$ with curvature parameter b and coupling strength ε [22]. We use $b = 1$ and $\varepsilon = 0.1$ as suggested in [22].
- For WD, we use an update function proposed in [33] (see (24) in that paper). Making changes to fit the value range of our variables and introducing a

Algorithm 1 Synchronization algorithms

1. An oscillator increases its phase $\phi(t)$ from 0 to 1.
2. Whenever $\phi(t) = 1$, the oscillator sends a packet.
3. Upon detection of synchronization word at time t, the oscillator adjusts its phase according to:

$$\phi(t^+) = \begin{cases} H_{\text{PS}}(\phi(t)) & \text{for PS and} \\ H_{\text{WD}}(\phi(t)) & \text{for WD.} \end{cases}$$

(a) PS / PS

1. An oscillator increases its phase $\phi(t)$ from 0 to 1.
2. Whenever $\phi(t) = 1$, the oscillator sends a packet with probability $p < 1$.
3. Upon detection of synchronization word at time t, the oscillator adjusts its phase according to:

$$\phi(t^+) = H_{\text{IES}}(\phi(t)).$$

(b)IES

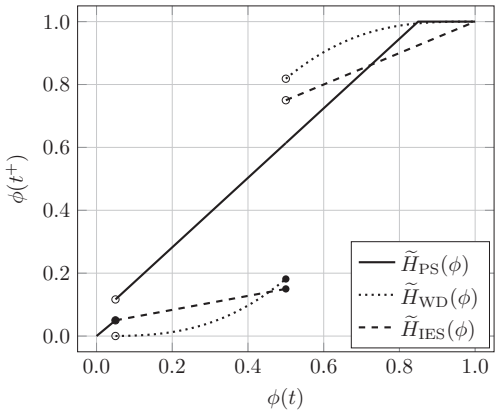


Fig. 2. Update functions.

refractory interval yields

$$\widetilde{H}_{\mathrm{WD}}(\phi) = \begin{cases} \phi & \text{if } \phi \le \phi_{\mathrm{ref}}, \\ \phi - F(\phi) & \text{if } \phi_{\mathrm{ref}} < \phi \le \frac{1}{2}, \\ \phi + F(\phi) & \text{if } \frac{1}{2} < \phi \le 1, \end{cases}$$

where $F(\phi) = f_\pi^{-1}\left(\sqrt{C/\pi}\sin\left(f_\pi(\phi)/2\right)\right)$, $f_\pi(\phi) = 2\pi\phi$, $f_\pi^{-1}(x) = x/(2\pi)$ and C is a scaling constant. The constant C is given in [33] as $2\pi^3/3$. This value, however, does not satisfy (2). The following property, however, states the interval of values C for which (2) is satisfied:

Property 1. For $0 \le C \le 4\pi$, we have $0 \le \widetilde{H}_{\mathrm{WD}}(\phi) \le 1$ for all $0 \le \phi \le 1$.

Proof. Let $C' = \sqrt{C/\pi}$. We have $0 \le f_\pi(\phi) - C'\sin\left(f_\pi(\phi)/2\right) \le 2\pi \Rightarrow -\pi \le C' \le 2$, and $0 \le f_\pi(\phi) + C'\sin\left(f_\pi(\phi)/2\right) \le 2\pi \Rightarrow -2 \le C' \le \pi$. Combining the above inequalities we get $-2 \le C' \le 2$. Thus, we have $0 \le C \le 4\pi$. □

In our experiments we apply $C = 4\pi$, as it yields the fastest synchronization of all $C \in [0, 4\pi]$ values. Note that in [33] the coupling weight is assumed to be much smaller than one. For comparability between the algorithms, however, we use unit coupling. Our experimental evaluations have shown no disadvantages in terms of, e.g., synchronization convergence when using unit coupling.

- IES uses an update function of the form [7]

$$\widetilde{H}_{\mathrm{IES}}(\phi) = \begin{cases} \phi & \text{if } \phi \le \phi_{\mathrm{ref}}, \\ h_1(\phi) & \text{if } \phi_{\mathrm{ref}} < \phi \le \frac{1}{2}, \\ h_2(\phi) & \text{if } \frac{1}{2} < \phi \le 1, \end{cases}$$

where h_1 and h_2 are continuous functions that satisfy the requirements stated in [7].
Throughout this paper we use $h_1(\phi) = \alpha \cdot [\phi - f_\phi(\tau_{\max})] + f_\phi(\tau_{\max})$ and $h_2(\phi) = \beta \cdot [\phi - 1] + 1$ with $\alpha = \frac{\frac{1}{4} - 2f_\phi(\tau_{\max}) - f_\phi(\tau_{\min})}{\frac{1}{2} - f_\phi(\tau_{\max})}$ and $\beta = \frac{1}{2} + 2f_\phi(\tau_{\min}) - 2f_\phi(\tau_{\max})$. These functions fulfill all requirements.

Fig. 2 plots the three update functions for $\phi_{\mathrm{ref}} = f_\phi(\tau_{\min}) = f_\phi(\tau_{\max}) = 0.05$. WD and IES use inhibitory coupling (the phase is decreased) in the interval $(\phi_{\mathrm{ref}}, 0.5]$ and excitatory coupling (the phase is increased) in (0.5, 1.0). PS is excitatory over $(\phi_{\mathrm{ref}}, 1)$.

4. Radio implementation

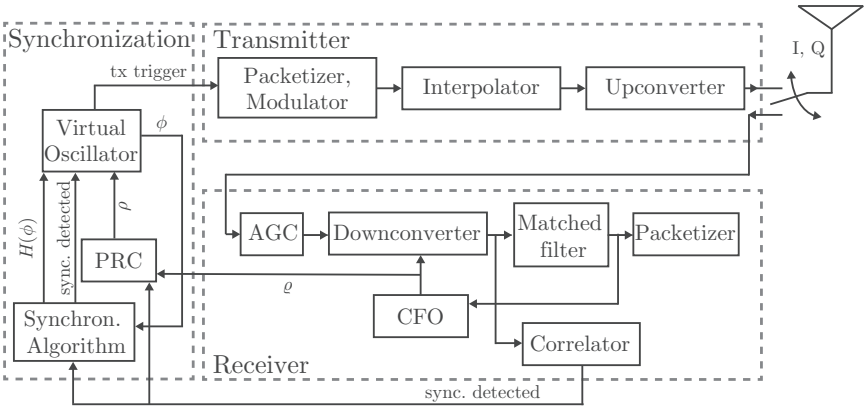
We implement all synchronization algorithms on the FPGA-based radio boards of the Wireless Open-Access Research Platform (WARP) [36]. A custom single-carrier physical layer is used with 5 MHz bandwidth and quadrature phase-shift keying (QPSK) as modulation scheme. Boards operate at 2.4 GHz and use a peak transmit power of 20 dBm. The overall structure of the transceiver is shown in Fig. 3(a). The architecture comprises three main components: transmitter, receiver, and synchronization. All components are implemented directly on the FPGA.

On the transmitter side, the packetizer and modulator build a packet after a trigger signal is received from the synchronization component. Two cyclic redundancy checksums (CRC) of 32 bit length each are added for header and payload. The modulated packet is then fed into an interpolator and upconverter, and finally transmitted over the air. We send packets of 1060 bytes, consisting of an 8 byte preamble used for automatic gain control (AGC) and to combat carrier frequency offsets (CFO). A 4 byte synchronization word is used for the synchronization algorithms. The header has 24 bytes. The remaining 1024 bytes are the payload.

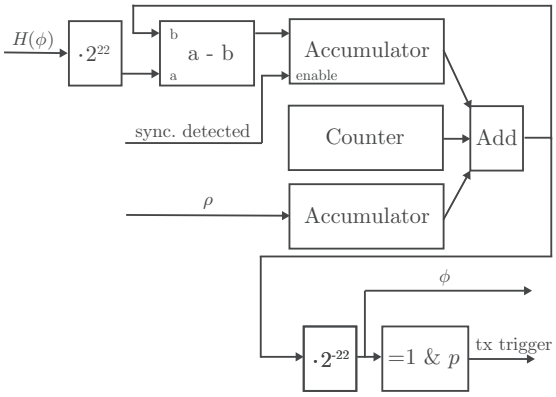
On the receiver side, the inphase (I) and quadrature (Q) components of the signal are used to estimate and set the amplifier gains of the boards (AGC). The downconverter brings signals to the baseband. A non-data aided algorithm is implemented for carrier frequency offset (CFO) correction (cf. [37], pp. 727–738; [38]) as illustrated in Fig. 3(c). The carrier frequency offset is estimated by evaluating $|I| - |Q|$. A non-zero value indicates a mismatch between the carrier frequency of the received signal and the current downconverter frequency of the receiver. The resulting value ϱ indicates the phase increment which is fed back into the downconverter to adjust its downconversion frequency. The output of the downconverter is fed into two modules:

(i) The first module is a correlator, implemented as a finite impulse response (FIR) filter, to detect the synchronization word inserted on the transmitter side. Whenever such a synchronization word is detected, a trigger signal is forwarded to the synchronization module and to the phase rate equalization (PRE) module.
(ii) The second module is the matched filter module, which downsamples the signals and forwards them to the packetizer, where header and payload of the packet are reconstructed and, concurrently, the CRCs from the transmitter are verified.

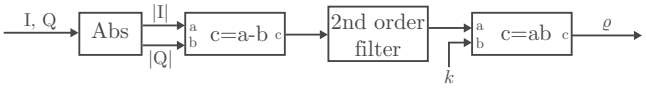
The synchronization component on the FPGA consists of three modules: virtual oscillator, synchronization algorithm, and phase rate equalization (PRE) algorithm. The synchronization algorithm module contains implementations of the algorithms discussed in Section 3. The execution of the algorithms is triggered whenever



(a) Overall FPGA design



(b) Virtual oscillator



(c) CFO correction

Fig. 3. FPGA design.

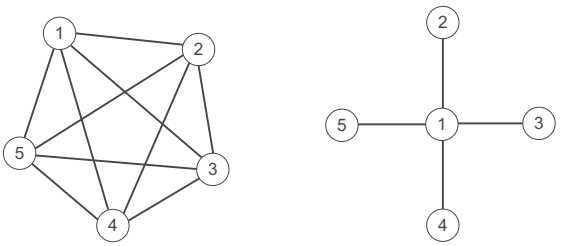
a synchronization word is detected by the correlator. This module then evaluates the new phase $H(\phi)$ based on the current phase ϕ according to the specification of the chosen synchronization algorithm. The virtual oscillator block (Fig. 3(b)) replicates the oscillator on the WARP boards. The main part of the module is a 22 bit wrap-around counter running at a clock frequency of 40 MHz. This counter expresses, after reinterpreting the output as a fractional number by multiplying it with 2^{-22}, the phase of the virtual oscillator. Thus, the cycle duration of the oscillator is $\delta = \frac{2^{22}}{40 \cdot 10^6}$ s ≈ 105 ms. There are two accumulators: the first is used to adjust the phase ϕ to $H(\phi)$, and the second is used to adjust the phase rate of the oscillator by adding a correction term ρ. This correction term is determined by the PRE module explained in Section 6. The value $\rho = 0$ is used if PRE is not applied.

5. Synchronization performance

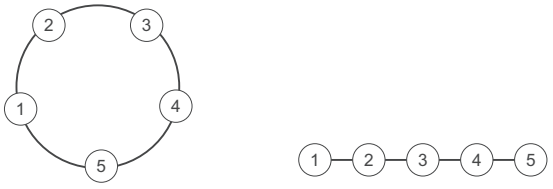
Let us now evaluate the performance of the synchronization algorithms by measurements over a real channel. Three terms are used as performance indicators: (i) synchronization precision, (ii) synchronization convergence, and (iii) synchronization time. Before giving a formal definition of these terms in Section 5.2, we discuss the experimental setup.

5.1. Experimental setup

Five WARP radio boards (denoted as radios in the following) are used in four network topologies shown in Fig. 4: a fully connected topology, a star topology, a ring topology, and a line topology. The topologies are created



(a) fully connected (b) star



(c) ring (d) line

Fig. 4. Network topologies.

Table 1
System parameters.

Parameter	Value
Radio carrier frequency f	2.4 GHz
Transmission power P_{Tx}	20 dBm
Cycle duration δ	≈105 ms
Minimum delay $\tau_{\min}$	75.61 µs
Mean delay $\overline{\tau}$	75.88 µs
Maximum delay $\tau_{\max}$	76.12 µs
Coupling strength ε (PS)	0.1
Curvature b (PS)	1
Scaling constant C (WD)	4π
Sending probability p (IES)	0.5

in the lab by packet filtering: a receiving node evaluates the sender identification (id) in the packet header and discards packets with a "wrong" id. The distances between all radios are a few meters. For each algorithm and each topology, we observe 100 measurement runs.

Measurements are conducted as follows: we use an Agilent 33220A waveform generator [39] connected to each of the five radios via the "debug header connector".This connector is directly connected to the FPGA allowing for extremely low delay input/output. The waveform generator generates a trigger signal in each cycle. Upon reception of such a signal, each radio writes its current phase over the Ethernet to a dedicated computer where it is logged. Based on this data, we evaluate the synchronization algorithms. The measurement results are accurate to a precision of ± 25 ns.

For each algorithm and topology, we perform at least 100 measurement runs, where at the beginning of each run ($t = 0$) the phases of all radios are randomly initialized following a uniform distribution. In particular, the phase of each radio is independently chosen from all other radios.

Table 1 shows the system parameters used for measurements.

The propagation and processing delays τ are important to all synchronization algorithms. Measurements show that, for the used implementation, the minimum delay $\tau_{\min}$ is 75.61 µs, the mean delay $\overline{\tau}$ is 75.88 µs, and the maximum delay $\tau_{\max}$ is 76.12 µs.

5.2. Definition of precision, convergence, and synchronization time

The synchronization *precision* $\Gamma(t)$ in seconds at time instant t is defined as (cf. (6) in [7])

$$\Gamma(t) = \delta \cdot \max_{i,j} \{\min[|\phi_i(t) - \phi_j(t)|, 1 - |\phi_i(t) - \phi_j(t)|]\}\,.$$

Note that we multiply the phase deviation, which is a value between 0 and 1, by the cycle duration δ to get the precision in seconds.

We say that an algorithm *converges in a certain measurement run* if there is a cycle c^* such that for all $c^* \le c \le 1000$ we have

$$\Gamma(\delta \cdot c) < \zeta\,, \tag{3}$$

where $\zeta > 0$ is the *precision threshold*. Furthermore, we say that an algorithm *converges* if there is a c^* such that (3) holds for all 100 measurement runs.

For an algorithm which converges, we define the *synchronization time* as the minimum cycle c^*, for which $\Gamma(\delta \cdot c) < \zeta$ for all $c \ge c^*$.

5.3. Results on performance

5.3.1. State-of-the-art algorithms

Figs. 5 and 6 show the precisions of PS, WD, and IES over time. We show its mean along with the 5% and 95% quantiles over the measurement runs.

The major results are as follows: For $\zeta = 100\,\mu s$, WD and IES converge to synchrony in a fully connected network topology; the eventually achieved precision differs from algorithm to algorithm. The IES algorithm (Figs. 5 and 6, bottom row) achieves the best average precision of about 3 µs. The precision of WD (Figs. 5 and 6, middle row) is worse, which is mainly due to the fact that the radios cannot hear each other if their transmissions overlap. In contrast, IES uses stochastic pulse emission, which means that radios are only sending with probability $p = 0.5$, such that non-sending radios can detect packets of sending radios and therefore adapt their phases. In all other studied topologies, only IES but neither PS nor WD converge. It is important to note that we use unit coupling for WD, but even if we use small coupling strengths as demanded in [33], it does not converge for these topologies. The mean precision of IES is 10 µs for star and ring and 18 µs for the line topology. In summary, IES converges for all studied topologies, and its synchronization precision degrades for star, ring, and line compared to the fully connected topology. The key factor for this behavior is non-homogeneous phase rates among the radios, which deteriorate the synchronization precision. This non-homogeneity is due to tolerances of the FPGAs; their clock speeds vary slightly, leading to non-homogeneous phase rates. Finally, the mean synchronization times (in cycles) are shown in Table 2. Again we use $\zeta = 100\,\mu s$.

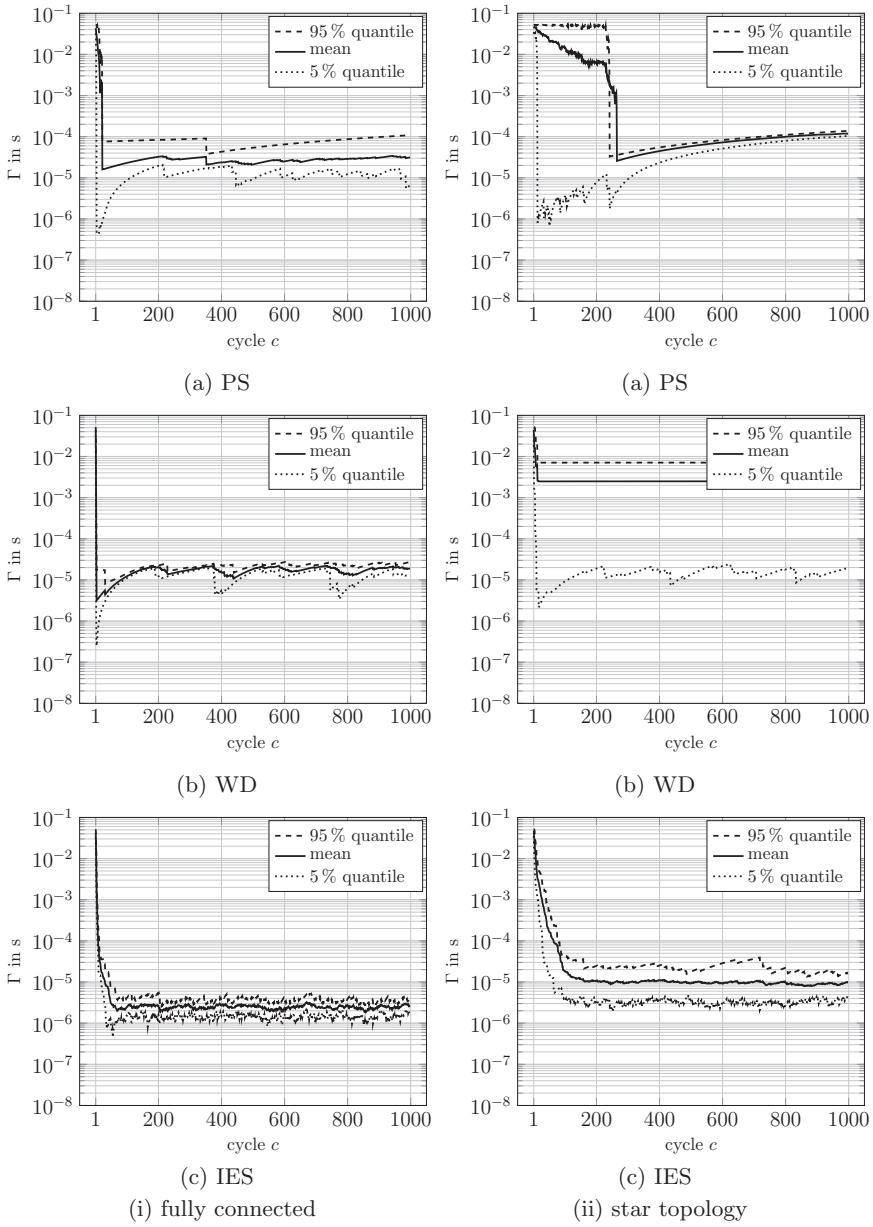


Fig. 5. Precision of pulse-coupled oscillator synchronization algorithms (fully connected and star topology).

Table 2
Mean synchronization times (in cycles) for $\zeta = 100\,\mu s$; "–" indicates that the algorithm does not converge in the given topology and for the specified threshold ζ.

Topology	PS	WD	IES
Fully connected	35.2	2.9	9.3
Star	–	–	54.1
Ring	–	–	44.6
Line	–	–	129.9

Table 3
Mean synchronization precision Γ for WD synchronization with stochastic pulse emission ($p = 0.5$).

Topology	Mean Γ
Fully connected	$2\,\mu s$
Star	$8\,\mu s$
Ring	$5\,\mu s$
Line	$10\,\mu s$

5.3.2. WD with stochastic pulse emission

The reason that WD does not converge in some of the studied topologies is due to its non-stochastic pulse emission. If we introduce stochastic pulse emission to this algorithm, experiments show that it converges for all above topologies with the mean synchronization precisions given in Table 3.

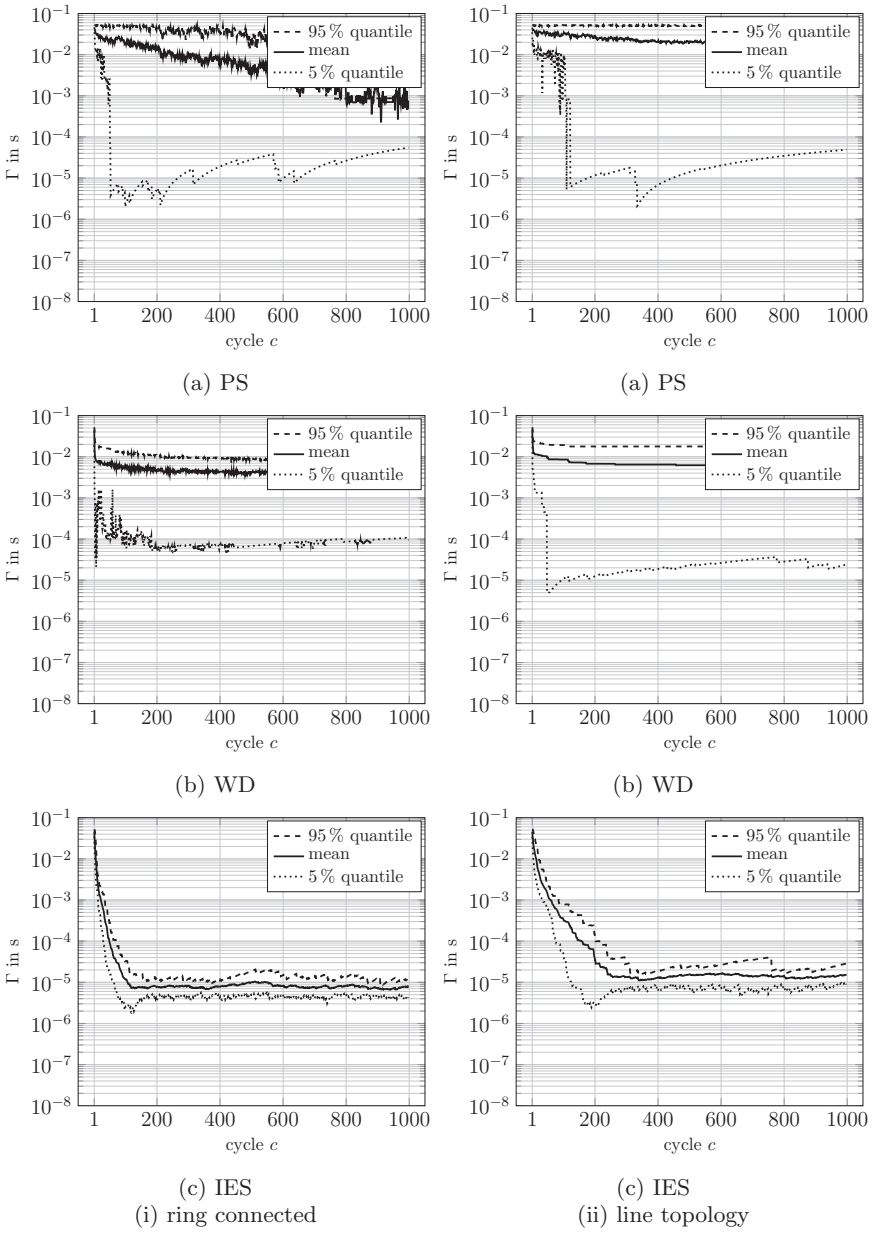


Fig. 6. Precision of pulse-coupled oscillator synchronization algorithms (ring and line topology).

These precision values are even slightly better than those of IES, and the synchronization times are considerably faster; the synchronization times (in cycles) are 3.1 (fully connected), 26.9 (star), 31.6 (ring), and 43.4 (line). The reason for these improvements seems to be due to the fact that WD adjusts the phase stronger, i.e., the phase jumps closer to 0 or 1, than IES (compare Fig. 2). The implementation effort for WD, in turn, is more complex than that of IES, as WD requires the computation of a sine function (cf. (3)) on the FPGA; IES requires only multiplications.

6. Phase rate equalization

The performance analysis of the previous section has shown that phase rate deviations between radios limit the achievable synchronization precision. By adding correction terms ρ in the virtual oscillator (Fig. 3(b)) we adapt its phase rate. The aim is to adapt the phase rates of all radios in a network to harmonize them. Since phase rates of radios depend on environmental factors that may change over time, e.g., temperature, an adaptive algorithm choosing the terms ρ_i for each radio i in a network is desired.

Furthermore, the algorithm should be distributed and scalable. We present such an equalization algorithm in this section. It can, in principle, be employed to any PCO-based synchronization algorithm that exchanges packets rather than pulses.

6.1. Algorithm for phase rate equalization

To measure the difference between sender j's and receiver i's phase rate, we use the CFO correction algorithm shown in Fig. 3(c). The phase increment ϱ_i of receiver i, determined by this algorithm, is proportional to the phase rate deviation ν_{ji} between sender j and receiver i. This phase increment expresses how much faster ($\varrho_i > 0$) or slower ($\varrho_i < 0$) the phase rate of j is compared to that of i. The proportionality constant is given by $\gamma = f/f'$, where f is the radio frequency and $f' = 40$ MHz is the intermediate frequency applied in the physical layer. This proportionality constant stems from fact that the upconverter brings the signal from baseband to f'. The upconversion from f' to f is done by the radio frequency (RF) interface [40,41]. At the receiver side, the RF interface brings the signal from f to f', and the downconverter on the FPGA brings it to the baseband. Thus, any phase rate deviations between two radios is multiplied by γ and an estimate for ν_{ji} is thus given by

$$\widehat{\nu}_{ji} = \frac{\varrho_i}{\gamma}. \tag{4}$$

The phase rate equalization (PRE) is specified in Algorithm 2. As a prerequisite, each radio j always writes,

Algorithm 2 Phase rate equalization (PRE) algorithm

1: **procedure** PRE header$_j$, ω
2: $\omega \in \mathbb{N} \setminus \{0\}$, global variable k
3: **if** CRC(header$_j$)==valid
4: estimate ν_{ji} with (4) resulting in $\widehat{\nu}_{ji}$
5: retrieve correction factor ρ_j from header$_j$
6: $\theta_k = \widehat{\nu}_{ji} + \rho_j$
7: **if** $\theta_k < 0$
8: $\rho_i = 0$
9: **else**
10: $\rho_i = \frac{1}{\min(\omega,k)} \sum_{l=\max(k-\omega+1,1)}^{k} \theta_l$
11: **end if**
12: $k = k + 1$
13: **end if**
14: **end procedure**

prior to packet transmission, its current correction term ρ_j into the packet's header. Whenever the synchronization word is detected by the correlator shown in Fig. 3(a) this algorithm is executed. The algorithm given in its current form is from the viewpoint where radio i detects the synchronization word in some packet sent by radio j. First of all, the algorithm checks whether or not the cyclic redundancy check (CRC) of the packet's header, received from j, is valid. If this is the case, the current correction factor ρ_j of j is retrieved from the received header and ν_{ji} is estimated by (4). The if-statement distinguishes between two cases: (i) if $\theta_k < 0$, radio i becomes the new leader with the (currently) fastest phase rate; (ii) if $\theta_k \geq 0$, radio i sets its own correction factor ρ_i to the value obtained by averaging over the last ω values θ_k.

Note that k is a global variable which is initialized to one at startup, and then incremented by one for each received packet with valid header.

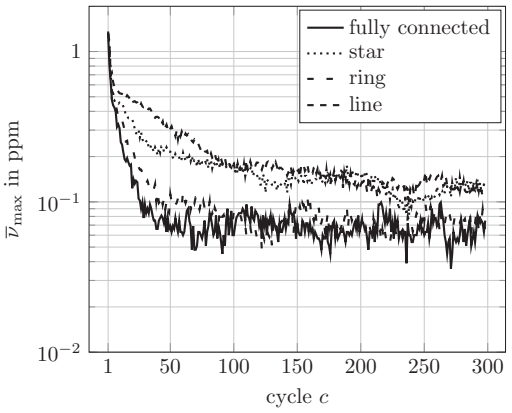


Fig. 7. Performance of PRE ($\omega = 10$).

Property 2. Under the assumption that $\widehat{\nu}_{ji} = \nu_{ji}$, the phase rates within a network converge to the fastest phase rate, as long as the nodes are connected among each other and each node sends with probability $p > 0$.

Proof. Let $\mathcal{S}$ be the set of all connected nodes of a network. Since each node sends with probability $p > 0$, radio i, with the fastest phase rate in $\mathcal{S}$, eventually sends a message to a subset $\mathcal{A} \subset \mathcal{S}$, where $|\mathcal{A}| \geq 1$. Each node $j \in \mathcal{A}$ adjusts its rate according to $\rho_j = \frac{1}{\min(\omega,k)} \sum_{l=\max(k-\omega+1,1)}^{k} \theta_l$. Thus, the rates of all nodes in $\mathcal{A}$ come closer to radio i's rate. Eventually, the fastest rate is propagated throughout $\mathcal{S}$ and all rates converge to this rate. □

6.2. Performance of phase rate equalization

Fig. 7 illustrates the performance of PRE with $\omega = 10$. The results are based on 100 measurement runs; the PRE algorithm and the phase rates are reset prior to each run. Each node transmits a packet with probability $p = 0.5$ when its phase reaches one. The phases are randomly initialized prior to each run and no synchronization of the phases is performed. We evaluate the maximum pairwise phase rate deviation at time t, defined as $\nu_{\max}(t) = \max_{i,j} |\nu_{ij}(t)|$, and plot its mean value $\overline{\nu}_{\max}(t)$ over all runs.

The plot illustrates the following: The rates of the nodes are initially uncorrected — we have $\overline{\nu}_{\max}(0) > 1$ parts per million (ppm). As intended, the average rate deviation decreases over time, until it reaches a certain saturation that shows a significantly improved rate deviation in all topologies. The fully connected topology reaches 0.09 ppm after about 50 cycles. Convergence takes longer for the other

topologies. After 300 cycles, the average deviation is between 0.09 and 0.12 ppm.

7. Performance with phase rate equalization

7.1. Synchronization algorithms

As mentioned, phase rate equalization (PRE) can be integrated into any PCO-based algorithm that exchanges packets rather than pulses. Whenever a packet is received, a radio not only adjusts its phase but also executes the PRE algorithm.

Some further modifications are required: (i) the function (1) is modified to $H_{\mathcal{X}}(\phi) = \widetilde{H}_{\mathcal{X}}(\phi - f_\phi(\overline{\tau}) \bmod 1) + f_\phi(\overline{\tau}) \bmod 1$, i.e., we now consider delay by subtracting the mean delay $\overline{\tau}$ instead of $\tau_{\min}$; (ii) to account for the possibility of jumping to a phase shortly before one, we introduce a condition where an oscillator only sends a packet (with probability p) when $\phi = 1$, and no packet was received within the last $\overline{\tau} - \tau_{\min}$ seconds.

We analyze PRE in both IES and WD synchronization, which yields IES *synchronization with automatic phase rate equalization* (IES_{PRE}) and WD *synchronization with automatic phase rate equalization* (WD_{PRE}). Furthermore, in [33,42] the optimal phase response function in case of homogeneous phase rates is given as (Eq. (9) in [33]):

$$F(\phi) = \begin{cases} -\phi & \text{if } 0 \le \phi \le \pi, \\ 2\pi - \phi & \text{if } \pi < \phi \le 2\pi. \end{cases}$$

This equation means that, when using unit coupling, upon reception of a pulse an oscillator immediately set its phase to 0. Thus, when additionally considering the refractory interval and the mean delay $\overline{\tau}$, we yield the following update function:

$$\widetilde{H}_{\text{WD}^*}(\phi) = \begin{cases} \phi & \text{if } \phi \le \phi_{\text{ref}}, \\ f_\phi(\overline{\tau}) & \text{else.} \end{cases}$$

Algorithm 3 specifies the three algorithms as implemented and applied for the experiments.

7.2. Results on performance

Fig. 8 shows the synchronization precisions of IES_{PRE}, WD_{PRE} and WD^*_{PRE}. Again we employ $p = 0.5$ and $\zeta = 100\,\mu\text{s}$. The algorithms converge for all network topologies. Due to space limitations, we only show the results for the fully connected and for the line topology. The mean synchronization precisions are for all algorithms much better compared to the case where no phase rate equalization is applied. Compare, e.g., the synchronization precision of IES_{PRE} (Fig. 8, top row) with that of IES (Figs. 5 and 6). After 1 000 cycles, the mean synchronization precision is, for all three algorithms, at about 400 ns for the fully connected and at about 1 µs for the line topology.

Algorithm 3 $\text{IES}_{\text{PRE}}/\text{WD}_{\text{PRE}}/\text{WD}^*_{\text{PRE}}$

1. An oscillator increases its phase $\phi(t)$ from 0 to 1.
2. Whenever $\phi(t) = 1$ and no packet was received within the last $\overline{\tau} - \tau_{\min}$ seconds, the oscillator sends a packet with probability $p < 1$.
3. Upon detection of a synchronization word at t:
   - the oscillator executes algorithm PRE, and
   - adjusts its phase according to:

$$\phi(t^+) = \begin{cases} H_{\text{IES}}(\phi(t)) & \text{for } \text{IES}_{\text{PRE}}, \\ H_{\text{WD}}(\phi(t)) & \text{for } \text{WD}_{\text{PRE}}, \\ \widetilde{H}_{\text{WD}^*}(\phi(t)) & \text{for } \text{WD}^*_{\text{PRE}}. \end{cases}$$

Table 4
Mean synchronization times (in cycles) for $\zeta = 100\,\mu\text{s}$

Topology	IES_{PRE}	WD_{PRE}	WD^*_{PRE}
Fully connected	9.4	3.2	2.0
Line	101.9	30.9	26.6

In terms of synchronization precision we have seen that all three algorithms reach similar precision. With respect to synchronization time, the algorithms, however, show considerably different performance. In all topologies, the synchronization time for a threshold of $\zeta = 100\,\mu\text{s}$ is much faster for WD_{PRE} and WD^*_{PRE} than for IES_{PRE}. This is due to the non-linear update function of WD, which leads to bigger phase adjustments of each radio when detecting packets. Therefore, the radios converge faster with such a non-linear update function than when applying a linear update function as used by IES_{PRE}. The synchronization times in number of cycles are shown in Table 4.

7.3. Impact of stochastic coupling

We have seen that stochastic pulse emission [7] is an essential building block for convergence of synchronization. The sending probability $p < 1$ is a parameter to be specified. We now investigate by experiments the impact of this parameter on the precision for IES_{PRE} in the fully connected topology.

Fig. 9 shows the results. Sending probabilities from $p = 0.2$ to 0.5 yield the best precision. A probability $p \ge 0.8$ deteriorates the precision considerably.

This impact of p on the achieved precision is due to the fact that once the radios become more and more synchronized, packets overlap, and more collisions occur for inappropriate sending probabilities p. Due to these collisions, the duration in between detected synchronization words becomes longer and the synchronization precision reduces due to phase rate deviations. For example, at cycle 1 000, a mean synchronization precision of about 300 ns is achieved for $p = 0.2$ and 3 µs for $p = 0.8$. Note that we employ IES_{PRE} and therefore adjust the phase rates of the radios. The phase rate deviations are reduced (cf. Fig. 7), but are still nonzero.

It is clear that the synchronization time increases when reducing the sending probability. Thus, a strategy to achieve both fast synchronization and high precision at the same time, is to start with, e.g., $p = 0.5$ and to gradually decrease the probability until the optimal probability is

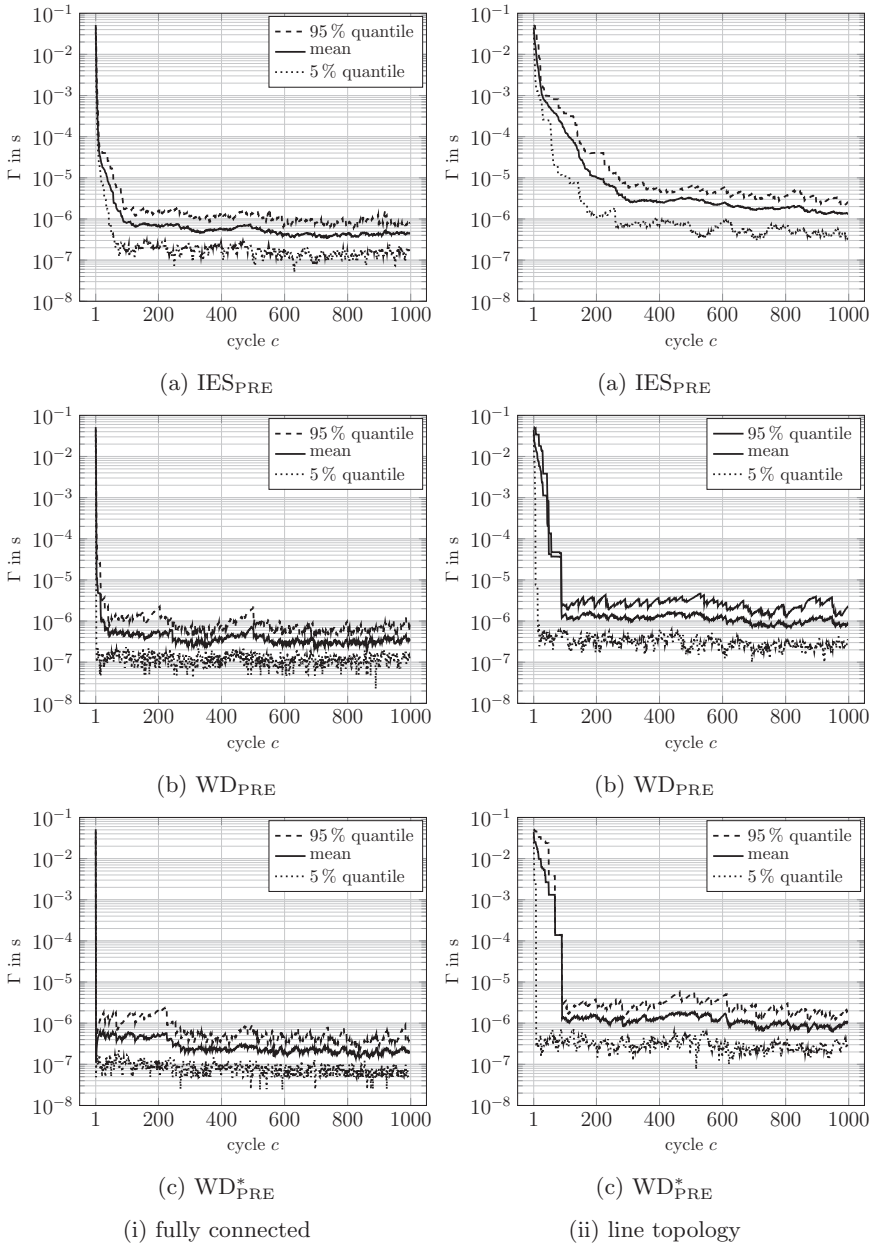


Fig. 8. Precision of the IES$_{\mathrm{PRE}}$ and WD$_{\mathrm{PRE}}$ algorithms.

reached. From the discussion above, it is clear that in a fully connected network of n nodes, the optimal p is given as $1/n$. For this probability on average only one node is transmitting in a given cycle. Thus, few packet collisions occur when the nodes become increasingly synchronized.

7.4. Comparison to centralized synchronization

As a reference we compare the precision of the PCO synchronization algorithms to the performance achieved when applying a simple centralized synchronization algorithm, where a dedicated node (master) continuously sends out messages to all other nodes (slaves). The specification of the centralized synchronization algorithm is shown in Algorithm 4. Note that we apply the phase rate equalization (PRE) algorithm.

Fig. 10 shows the synchronization precisions achieved when applying Algorithm 4. Again we use five radios, which are all in communication range of each other. The radio with identification number 0 is the master; all other radios are slaves. The results are averaged over 100 measurement runs.

The figure shows that the synchronization precisions are better than that of all previously discussed PCO

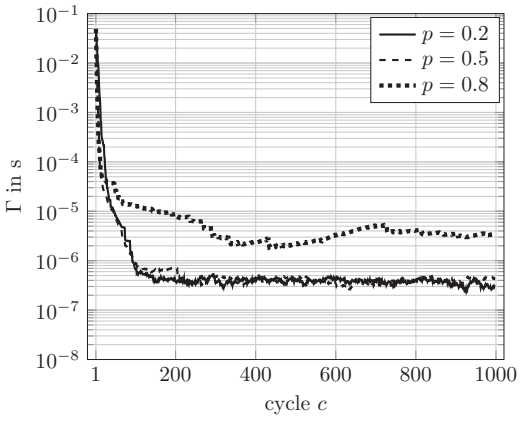


Fig. 9. Impact of sending probability p on the precision of IES_{PRE} (fully connected topology).

Algorithm 4 Centralized synchronization algorithm

1. An oscillator increases its phase $\phi(t)$ from 0 to 1.
2. Whenever $\phi(t) = 1$, the oscillator with identification number 0 sends a packet.
3. Upon detection of a synchronization word at t:
   - the oscillator executes algorithm PRE, and
   - adjusts its phase according to:

$$\phi(t^+) = f_\phi(\overline{\tau}).$$

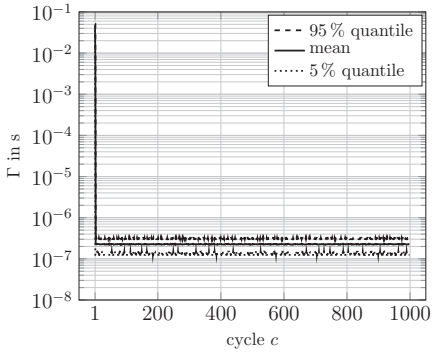


Fig. 10. Centralized synchronization with PRE.

synchronization algorithms. For the centralized synchronization, the mean precision is about 125 ns. Furthermore, synchronization among the radios is achieved immediately after the slaves receive the first packet. This clearly demonstrates the trade-off between synchronization performance on the one hand, and the benefits of distributed approaches for synchronization on the other hand.

7.5. Performance in networks of many nodes

So far, all presented results have been based on real measurements with five WARP radios. It is, however, of interest how the performance of the synchronization algorithms changes, if the number of nodes in the network increases. Increasing the number of radios significantly is for us, however, not possible, due to the fact that the radios are quite expensive. Therefore, we simulate the synchronization performance in fully connected and line topology networks with up to 50 nodes. For this simulation we incorporate the physical layer parameters of the FPGA implementation of the WARP radios. The phase rates of the WARP radios are modeled to be Gaussian distributed with zero mean and standard deviation of 2.5 ppm, which is a typical specification for standard oscillators; the phase rate equalization algorithm is simulated by modeling the uncertainty of the estimated phase rate deviation $\widehat{\nu}_{ji}$ at receiver i as

$$\widehat{\nu}_{ji} = \nu_{ji} + \nu_{ji}\mathcal{X},$$

where $\mathcal{X}$ is Gaussian with zero mean and standard deviation $\sigma = 0.03$. Thus, approximately 99% of all random values of $\mathcal{X}$ are within $[-0.1, 0.1]$. This models $\widehat{\nu}_{ji}$ to be exact to $\pm$ 10 % in 99% of all cases.

To verify the simulation, we have compared the experimental results with simulation-based results for $n = 5$ nodes. The simulated performance closely matches, for all algorithms, the experimental results. Due to the insights gained from Section 7.3, we now apply stochastic coupling with time-variant sending probability for the fully connected topology: We start with $p = 0.5$ and steadily decrease the probability until the optimal probability, i.e., $p = 1/n$ is reached. More specifically, we model p as a time-variant function $p(c)$ of the cycle c as

$$p(c) = 0.5 - \frac{(0.5 - 1/n)\ \min(c, 500)}{500},$$

which steadily decreases the probability from $p(0) = 0.5$ until $p(500) = 1/n$; from that cycle onward the probability remains at $1/n$. In the line topology we use the time-invariant stochastic sending probability of $p = 0.33$.

Fig. 11 shows the mean precisions of both IES_{PRE} and WD_{PRE} in both fully connected (a) and line (b) topology. For IES_{PRE} we reach steady-state mean precisions of about 400 ns ($n = 5$), 700 ns ($n = 10$), and 3 µs ($n = 50$) in the fully connected topology. For WD_{PRE}, we get about 300 ns ($n = 5$), 600 ns ($n = 10$), and 3 µs ($n = 50$). In the line topology we get the following steady-state mean precisions: 1 µs ($n = 5$), 2.3 µs ($n = 10$), and 14 µs ($n = 50$) for IES_{PRE}; 0.8 µs ($n = 5$), 2 µs ($n = 10$), and 10 µs ($n = 50$) for WD_{PRE}.

Table 5 shows the synchronization times in cycles for reaching precisions better than $\zeta = 100\,\mu\text{s}$.

Table 5
Simulated mean synchronization times (in cycles) for $\zeta = 100\,\mu\text{s}$.

	IES_{PRE}	WD_{PRE}		IES_{PRE}	WD_{PRE}
$n = 5$	10.8	4.2	$n = 5$	126.5	27.9
$n = 10$	7.5	4.5	$n = 10$	381.8	112.4
$n = 50$	6.2	5.3	$n = 50$	3264.5	1121.2
(a) Fully connected			(b) Line		

8. Conclusions and outlook

Our experiments show that PCO algorithms can achieve convergence to synchrony with precisions below one

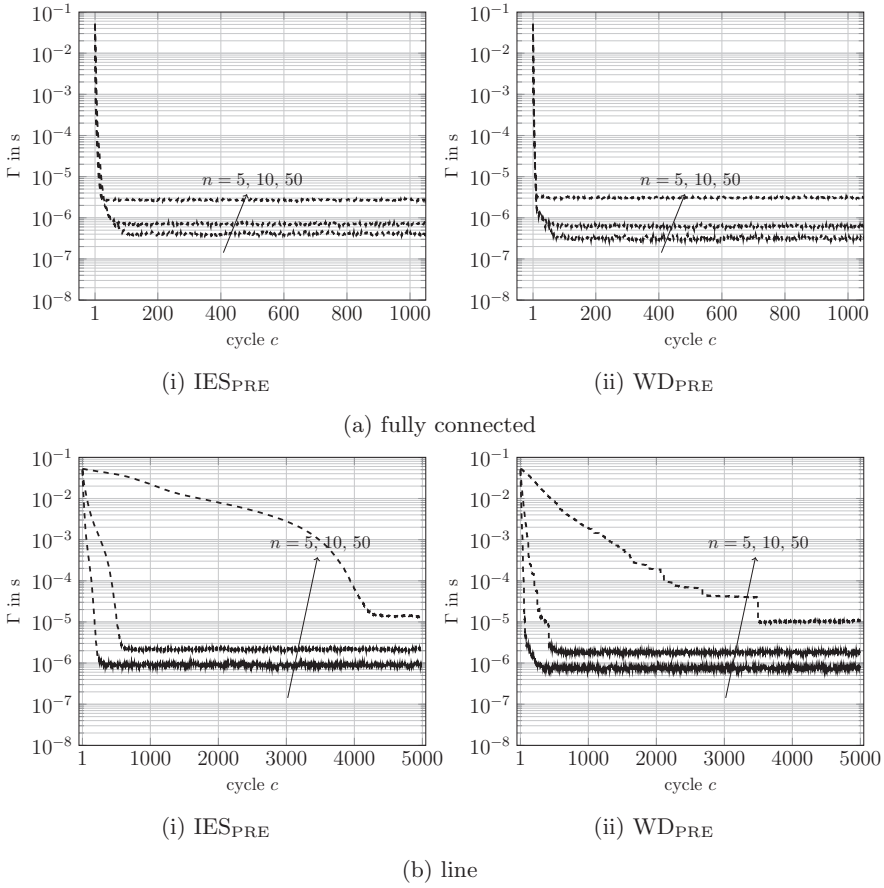


Fig. 11. Simulated precisions of IES_{PRE} and WD_{PRE} in fully connected topology (a) and line topology (b). For (a) we use time-variant stochastic sending probability $p(c)$; for (b) we use time-invariant stochastic sending probability $p = 0.33$.

microsecond in real-world settings. A key ingredient for stable convergence is stochastic coupling, which can be achieved by introducing the feature that synchronization packets are not always sent but only with a certain probability. A limiting factor for high precision is non-homogeneous phase rates among radios. Hence, another key ingredient for high precision is phase rate equalization, for which a novel distributed algorithm has been proposed, integrated, demonstrated, and analyzed.

These insights are important for the design and assessment of PCO algorithms. We can state that algorithms containing stochastic coupling and phase rate equalization along with an update function that combines excitatory and inhibitory coupling (as WD and IES do) are able to reach precisions that are sufficient for many applications, such as timing of sleep cycles and transmission slots, while they are still conceptually simple and completely distributed.

Acknowledgment

This work was supported by Lakeside Labs with funding from ERDF, KWF, BABEG under Grant 20214/23794/35530.

References

[1] R.E. Mirollo, S.H. Strogatz, Synchronization of pulse-coupled biological oscillators, SIAM J. Appl. Math. 50 (6) (1990) 1645–1662.
[2] C.A. van Vreeswijk, L.F. Abbott, G.B. Ermentrout, When inhibition, not excitation, synchronizes neural firing, J. Comput. Neurosci. 1 (1994) 313–321.
[3] U. Ernst, K. Pawelzik, T. Geisel, Synchronization induced by temporal delays in pulse-coupled oscillators, Phys. Rev. Lett. 74 (9) (1995) 1570–1573.
[4] W. Gerstner, Rapid phase locking in systems of pulse-coupled oscillators with delays, Phys. Rev. Lett. 76 (1996) 1755–1758.
[5] U. Ernst, K. Pawelzik, T. Geisel, Delay-induced multistable synchronization of biological oscillators, Phys. Rev. E 57 (2) (1998) 2150–2162.
[6] J. Nishimura, E.J. Friedman, Robust convergence in pulse-coupled oscillators with delays, Phys. Rev. Lett. 106 (19) (2011) 194101, doi:10.1103/PhysRevLett.106.194101.
[7] J. Klinglmayr, C. Kirst, C. Bettstetter, M. Timme, Guaranteeing global synchronization in networks with stochastic interactions, New J. Phys. 14 (2012) 073031.
[8] C.S. Peskin, Mathematical Aspects of Heart Physiology, Courant Institute of Mathematical Sciences, pp. 268–278.
[9] D. Falconer, F. Adachi, B. Gudmundson, Time division multiple access methods for wireless personal communications, IEEE Commun. Mag. 33 (1) (1995) 50–57.
[10] E. Sourour, M. Nakagawa, Mutual decentralized synchronization for intervehicle communications, IEEE Trans. Veh. Technol. 48 (6) (1999) 2015–2027.

[11] J.-I. Chuang, Autonomous time synchronization among radio ports in wireless personal communications, IEEE Trans. Veh. Technol. 43 (1) (1994) 27–32.
[12] T. Tank, J.-P. Linnartz, Vehicle-to-vehicle communications for AVCS platooning, IEEE Trans. Veh. Technol. 46 (2) (1997) 528–536.
[13] R. Mathar, J. Mattfeldt, Pulse-coupled decentral synchronization, SIAM J. Appl. Math. 56 (4) (1996) 1094–1106.
[14] Y.-W. Hong, A. Scaglione, A scalable synchronization protocol for large scale sensor networks and its applications, IEEE J. Sel. Areas Commun. 23 (5) (2005) 1085–1099.
[15] D. Lucarelli, I.-J. Wang, Decentralized synchronization protocols with nearest neighbor communication, in: Proceedings of the Second International Conference on Embedded Networked Sensor Systems (SenSys), Baltimore, MD, USA, 2004, pp. 62–68.
[16] G. Werner-Allen, G. Tewari, A. Patel, M. Welsh, R. Nagpal, Firefly-inspired sensor network synchronicity with realistic radio effects, in: Proceedings of the Third International Conference on Embedded Networked Sensor Systems (SenSys), San Diego, CA, USA, 2005, pp. 142–153.
[17] A. Tyrrell, G. Auer, C. Bettstetter, Emergent slot synchronization in wireless networks, IEEE Trans. Mob. Comput. 9 (5) (2010) 719–732.
[18] A. Tyrrell, G. Auer, Decentralized inter-base station synchronization inspired from nature, in: Proceedings of the Sixty-eighth IEEE Vehicular Technology Conference (VTC), Calgery, BC, Canada, 2008, pp. 1–5.
[19] L. Sanguinetti, A. Tyrrell, M. Morelli, G. Auer, On the performance of biologically-inspired slot synchronization in multicarrier ad hoc networks, in: Proceedings of the IEEE Vehicular Technology Conference (VTC), Singapore, 2008, pp. 21–25.
[20] A. Tyrrell, G. Auer, Imposing a reference timing onto firefly synchronization in wireless networks, in: Proceedings of the IEEE Vehicular Technology Conference (VTC), Dublin, Ireland, 2007, pp. 222–226.
[21] R. Leidenfrost, W. Elmenreich, Firefly clock synchronization in an 802.15.4 wireless network, EURASIP J. Embed. Syst. 2009 (2009) 1–17.
[22] R. Pagliari, A. Scaglione, Scalable network synchronization with pulse-coupled oscillators, IEEE Trans. Mob. Comput. 10 (3) (2011) 392–405.
[23] G. Brandner, J. Klinglmayr, D. Egarter, U. Schilcher, C. Bettstetter, Precision of pulse-coupled oscillator synchronization on FPGA-based radios, in: Proceedings of the International Conference on Systems, Communications, and Coding (SCC), Hamburg, Germany, 2015, pp. 1–6.
[24] J. Buck, E. Buck, J. Case, F. Hanson, Control of flashing in fireflies. V. Pacemaker synchronization in *pteroptyx cribellata*, J. Comp. Physiol. A 144 (3) (1981) 630–633.
[25] L.F. Abbott, C. van Vreeswijk, Asynchronous states in neural networks of pulse-coupled oscillators, Phys. Rev. E 48 (1993) 1483–1490.
[26] D. Golomb, G.B. Ermentrout, Bistability in pulse propagation in networks of excitatory and inhibitory populations, Phys. Rev. Lett. 86 (18) (2001) 4179–4182, doi:10.1103/PhysRevLett.86.4179.
[27] A. Tyrrell, G. Auer, C. Bettstetter, Fireflies as role models for synchronization in ad hoc networks, in: Proceedings of the International Conference on Bio-Inspired Models of Network, Information, and Computing Systems (BIONETICS), Cavalese, Italy, 2006.
[28] O. Simeone, U. Spagnolini, Y. Bar-Ness, S.H. Strogatz, Distributed synchronization in wireless networks, IEEE Trans. Signal Process. 25 (5) (2008) 81–97.
[29] Y. Wang, F. Núñez, F.J. Doyle III, Statistical analysis of the pulse-coupled synchronization strategy for wireless sensor networks, IEEE Trans. Signal Process. 61 (21) (2013a) 5193–5204.
[30] Y. Wang, F. Núñez, F.J. Doyle III, Increasing sync rate of pulse-coupled oscillators via phase response function design: Theory and application to wireless networks, IEEE Trans. Control Syst. Technol. 21 (4) (2013b) 1455–1462.
[31] Z. An, H. Zhu, X. Li, C. Xu, Y. Xu, X. Li, Nonidentical linear pulse-coupled oscillators model with application to time synchronization in wireless sensor networks, IEEE Trans. Ind. Electron. 58 (6) (2011) 2205–2215.
[32] R. Pagliari, A. Scaglione, Correction to 'scalable network synchronization with pulse-coupled oscillators', IEEE Trans. Mob. Comput. 10 (5) (2011) 749, doi:10.1109/TMC.2011.42.
[33] Y. Wang, F.J. Doyle III, Optimal phase response functions for fast pulse-coupled synchronization in wireless sensor networks, IEEE Trans. Signal Process. 60 (10) (2012) 5583–5588.
[34] Y. Kuramoto, Collective synchronization of pulse-coupled oscillators and excitable units, Physica B 50 (1991) 15–30.
[35] J. Klinglmayr, C. Bettstetter, Self-organizing synchronization with inhibitory-coupled oscillators: Convergence and robustness, ACM Trans. Auton. Adapt. Syst. 7 (3) (2012) 30:1–30:23.
[36] WARP Project: Wireless Open-Access Research Platform, Rice University, http://warp.rice.edu, (accessed 06.02.14).
[37] J.R. Barry, E.A. Lee, D.G. Messerschmitt, Digital Communication, Springer, 2003.
[38] E. Everett, M. Wu, Carrier Offset Recovery, Rice University, 2013 Technical Report. http://cmclab.rice.edu/433/slides/7-CarrierRecovery.pdf, (accessed 01.10.14)
[39] Agilent 33220A User's Guide, Technical Report, Agilent Technologies, 2007 http://cp.literature.agilent.com/litweb/pdf/33220-90002.pdf, (accessed 19.09.14).
[40] J.H. Reed, Software Radio: A Modern Approach to Radio Engineering, first ed., Prentice Hall, 2002.
[41] E. Everett, M. Wu, Frequency Upconversion and Downconversion, Rice University, 2013 Technical Report. http://cmclab.rice.edu/433/slides/6-UpDownConversion.pdf, (accessed 1.10.14)
[42] Y. Wang, F. Núñez, F.J. Doyle III, Energy-efficient pulse-coupled synchronization strategy design for wireless sensor networks through reduced idle listening, IEEE Trans. Signal Process. 60 (10) (2012) 5293–5306.

Günther Brandner received the Dipl.-Ing. degree (with distinction) in applied computing and mathematics from the University of Klagenfurt, Klagenfurt, Austria, in 2007 and 2008. Since 2007, he has been a research and teaching staff member and doctoral student with the Networked and Embedded Systems Institute, University of Klagenfurt. His main interests are relay selection methods for cooperative relaying and the implementation and evaluation of protocols on hardware platforms.

Udo Schilcher studied applied computing and mathematics at the University of Klagenfurt, where he received two Dipl.-Ing. Degrees with distinction (2005 and 2006). Since 2005, he has been research staff member at the Networked and Embedded Systems institute at the University of Klagenfurt. His main interests are interference and node distributions in wireless networks. His doctoral thesis on inhomogeneous node distributions and interference correlation in wireless networks and has been awarded with a Dr. techn. degree with distinction in 2011. He received a best paper award from the IEEE Vehicular Technology Society.

Christian Bettstetter received the Dipl.-Ing. degree in electrical engineering and information technology and the Dr.-Ing. degree (summa cum laude) in electrical engineering and information technology from the Technische Universität München (TUM), Munich, Germany, in 1998 and 2004, respectively. He was a Staff Member with the Communications Networks Institute, TUM, until 2003. From 2003 to 2005, he was a Senior Researcher with the DOCOMO Euro-Labs. Since 2005, he has been a Professor and the Head of the Institute of Networked and Embedded Systems, University of Klagenfurt, Klagenfurt, Austria. He is also the Scientific Director and Founder of Lakeside Labs, Klagenfurt: a research cluster on self-organizing networked systems. He coauthored the textbook GSM-Architecture, Protocols and Services (Wiley). He received Best Paper Awards from the IEEE Vehicular Technology Society and the German Information Technology Society (ITG).