# The Power of Synchronisation: Formal Analysis of Power Consumption in Networks of Pulse-Coupled Oscillators

Paul Gainer, Sven Linker, Clare Dixon, Ullrich Hustadt, and Michael Fisher

Department of Computer Science, University of Liverpool

Liverpool, L69 3BX, United Kingdom

*Abstract*—**Nature-inspired synchronisation protocols have been widely adopted to achieve consensus within wireless sensor networks. We assess the power consumption of such protocols, particularly the energy required to synchronise all nodes across a network. We use the widely adopted model of bio-inspired, pulse-coupled oscillators to achieve network-wide synchronisation and provide an extended formal model of just such a protocol, enhanced with structures for recording energy usage. Exhaustive analysis is then carried out through formal verification, utilising the PRISM model-checker to calculate the resources consumed on each possible system execution. This allows us to assess a range of parameter instantiations and to explore trade-offs between power consumption and time to synchronise. This provides a principled basis for the formal analysis of a much broader range of large-scale network protocols.**

## I. INTRODUCTION

Minimising power consumption is a critical design consideration for wireless sensor networks (WSNs) [1], [2]. Once deployed a WSN is generally expected to function independently for long periods of time. In particular, regular battery replacement can be costly and impractical for remote sensing applications. Hence, it is of the utmost importance to reduce the power consumption of the individual nodes by choosing low-power hardware and/or energy efficient protocols. However, to make informed choices, it is also necessary to have good estimations of the power consumption for individual nodes. While the general power consumption of the hardware can be extracted from data sheets, estimating the overall power consumption of different protocols is more demanding.

Surveys conducted by Irani and Pruhs [3] and Albers [2] investigated algorithmic problems in power management, in particular power-down mechanisms at the system and device level. Soua and Minet provided a general taxonomy for the analysis of wireless network protocols with respect to energy efficiency [4] by identifying the contributing factors of energy wastage, for instance packet collisions and unnecessary idling. These detrimental effects can be overcome by allocating time slots for communication between nodes. That is, nodes within a network synchronise their clock values and use different time slots for communication to avoid packet collisions [6], [7].

A number of biologically inspired protocols for synchronisation have been proposed [8], [9], [10], [11] and have been shown to be robust with respect to the topology of the network [12]. They are well-suited for WSNs since centralised control is not required to achieve synchrony. The protocols build on the underlying mathematical model of pulse-coupled oscillators (PCOs); integrate-and-fire oscillators with pulsatile coupling, such that when an oscillator fires it induces some phase-shift response determined by a *phase response function*. Over time the mutual interactions can lead to all oscillators firing synchronously. The PCO synchronisation model we employ was first proposed by Peskin [13] and later extended by Mirollo and Strogatz [14] who proved that several oscillators with the same frequency would always synchronise under the assumption of a fully coupled network. Later work by Lucarelli and Wang showed that this assumption could be relaxed, by proving that oscillators would always achieve synchrony if the coupling graph of the network was connected [15].

Simulating such a system provides good estimates for its typical behaviour, but may exclude corner cases where some unexpected behaviour is exhibited. To mitigate against this, we analyse the energy-consumption for the synchronisation of a network of PCOs using formal methods. Instead of using simulations, we use *probabilistic model checking* [16] to exhaustively examine all possible runs of the system. Probabilistic model checking can be used to formally specify performance measures and to analyse trade-offs in Markovian models [17], [18]. Using this technique we can calculate expected mean and worst-case energy costs for a network.



Fig. 1. The MICAz wireless measurement system.

In this work we abstract away from the modelling of individual oscillators and use a *population model* [28], [29], [30], [25] to encode information about groups of oscillators sharing the same configuration. Furthermore, we introduce *broadcast failures* where an oscillator may fail to broadcast its message. Since WSNs operate in stochastic environments under uncertainty we encode these failures within a *probabilistic* model. Our model also encapsulates means to associate different current draws with its states, thus enabling us to measure the energy consumption of the overall network. We employ the probabilistic model checker PRISM [31] to analyse the average and worst-
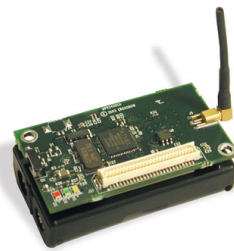
case energy consumption for both the synchronisation of arbitrarily configured networks, and restabilisation of a network, where a subset of oscillators desynchronised. To that end, we instantiate the model to analyse the power consumption of the MICAz wireless measurement system (see Fig. 1).

Exact time synchronisation, where all clocks always agree on their value, is never achieved for real-world deployments of synchronising devices [10]. Hardware imperfections result in different clock frequencies, environmental factors influence radio transmission, and network congestion leads to package collisions and loss [32]. Consequently, the precision of synchronisation is not required to be exact, and it is sufficient for all oscillators to fire within some defined time window [10]. The size of this window depends on the application. Some applications may require a very small window, for instance distributed sensing of mobile objects, while others may prefer energy efficiency at the cost of synchronisation precision [7]. To this end we extend the binary notion of synchronisation discussed in [25] by defining a metric derived from the complex order parameter of Kuramoto [33], [34] that captures the degree of synchrony of a fully connected network of oscillators as a real value in the interval $[0, 1]$.

The structure of the paper is as follows. In Sect. II we discuss related work, and in Sect. III we introduce the general PCO model, from which we derive population models in Sect. IV. Section V introduces the derived synchronisation metric. The construction of the formal model used for the analysis is presented in Sect. VI. Subsequently, in Sect. VII we evaluate the results for certain parameter instantiations and discuss their trade-offs with respect to power consumption and time to synchronise. Section VIII concludes the paper.

## II. RELATED WORK

While formal methods, in particular model checking, have been successfully used to model and analyse protocols for wireless sensor systems, the number of possible configurations that needs to be considered for larger WSNs impacts their feasibility. Chen et al. reviewed how different formal methods may be used to investigate ad-hoc routing protocols [19], suggesting that model checking is suitable for small networks, while analytical methods are necessary for larger networks. Yue and Katoen [21] used probabilistic model checking to optimise the energy consumption of a leader election protocol in networks of up to nine nodes. Probabilistic model checking was also used by Fehnker and Gao [20] to analyse flooding and gossiping protocols in networks of up to eight nodes, however it was necessary to use Monte Carlo simulations for the analysis of larger networks. Höfner and Kamali [?] used statistical model checking, an approach combining model checking, Monte Carlo sampling and hypothesis testing, to analyse a routing protocol for a network of sixteen nodes. Heidarian et al. used model checking to analyse clock synchronisation for medium access protocols [22]. They considered both fully-connected networks and line topologies with up to four nodes. Model checking of biologically inspired coupled oscillators has also been investigated by Bartocci et al. [23].

They present a subclass of timed automata [24] suitable to model biological oscillators, and an algorithm to detect synchronisation properties. However, their analysis was restricted to a network of three oscillators.

In [25] we introduced a formal population model for a network of PCOs, and investigated the expected time to achieve synchronisation and the probability for an arbitrarily configured population of oscillators to synchronise. In our model the oscillators synchronise over a finite set of discrete clock values, and the oscillation cycle includes a *refractory period* at the start of the oscillation cycle where an oscillator cannot be perturbed by other firing oscillators. This corresponds to a period of time where a WSN node enters a low-power idling mode. In this work we extend this approach by introducing a metric for global power consumption and discuss refinements of the model that allows us to formally reason about much larger populations of oscillators.

Wang et al. proposed an energy-efficient strategy for the synchronisation of PCOs [26]. In contrast to our work, they consider real-valued clocks and *delay-advance phase response* functions, where both positive and negative phase shifts can occur. A result of their choice of phase response function is that synchronisation time is independent of the length of the refractory period, in contrast to our model. Furthermore, they assume that the initial phase difference between oscillators has an upper bound. They achieve synchrony for refractory periods larger than half the cycle, while our models do not always synchronise in these cases, as we do not impose a bound on the phase difference of the oscillators. We consider all possible differences in phase since we examine the energy consumption for the resynchronisation of a subset of oscillators.

Konishi and Kokame conducted an analysis of PCOs where a perceived pulse immediately resets the oscillators to the start of their cycle [27]. Their goal was to maximise refractory period length, while still achieving synchronisation within some number of clock cycles. Similarly to our work, they restricted their analysis to a fully coupled network. They assumed that the synchronisation protocol was implemented as part of the physical layer of the network stack by using capacitors to generate pulses, therefore their clocks were continuous and had different frequencies. We assume that the synchronisation protocol resides on a higher layer, where the clock values are discretised and oscillate with the same frequency.

The energy consumption of the MICAz mote varies with the mode of its RF transceiver. The node has a receive mode, three transmission modes, and two low power idling modes. While Kramer and Geraldy [35] conducted an empirical investigation into the energy consumption of the MICAz, with respect to the different modes, Webster et al. [36] used probabilistic model checking to formally analyse very small numbers (not populations) of MICAz nodes, particularly the effect of clock drift on synchronisation.

## III. DISCRETE OSCILLATOR DYNAMICS

We consider a fully-coupled network of PCOs with identical dynamics over discrete time. The *phase* of an oscillator $i$ at

time $t$ is denoted by $\phi_i(t)$. The phase of an oscillator progresses through a sequence of discrete integer values bounded by some $T \geq 1$. The phase progression over time of a single uncoupled oscillator is determined by the successor function, where the phase increases over time until it equals $T$, at which point the oscillator will fire in the next moment in time and the phase will reset to one. The phase progression of an uncoupled oscillator is therefore cyclic with period $T$, and we refer to one cycle as an *oscillation cycle*.

When an oscillator fires, its firing may not be perceived by any of the other oscillators coupled to it. We call this a *broadcast failure* and denote its probability by $\mu \in [0,1]$. Note that $\mu$ is a global parameter, hence the chance of broadcast failure is identical for all oscillators. When an oscillator fires, and a broadcast failure does not occur, it perturbs the phase of all oscillators to which it is coupled; we use $\alpha_i(t)$ to denote the number of all other oscillators that are coupled to $i$ and will fire at time $t$. The *phase response function* is a positive increasing function $\Delta : \{1, \ldots, T\} \times \mathbb{N} \times \mathbb{R}^+ \to \mathbb{N}$ that maps the phase of an oscillator $i$, the number of other oscillators perceived to be firing by $i$, and a real value defining the strength of the coupling between oscillators, to an integer value corresponding to the perturbation to phase induced by the firing of oscillators where broadcast failures did not occur.

We can introduce a refractory period into the oscillation cycle of each oscillator. A refractory period is an interval of discrete values $[1, R] \subseteq [1, T]$ where $1 \leq R \leq T$ is the size of the refractory period, such that if $\phi_i(t)$ is inside the interval, for some oscillator $i$ at time $t$, then $i$ cannot be perturbed by other oscillators to which it is coupled. If $R = 0$ then we set $[1, R] = \emptyset$, and there is no refractory period at all. The *refractory function* ref $: \{1, \ldots, T\} \times \mathbb{N} \to \mathbb{N}$ is defined as $\text{ref}(\Phi, \delta) = 0$ if $\Phi \in [0, R]$, or $\text{ref}(\Phi, \delta) = \delta$ otherwise, and takes as parameters $\delta$, the degree of perturbance to the phase of an oscillator, and $\phi$, the phase, and returns zero if $\phi$ is in the refractory period defined by $R$, or $\delta$ otherwise.

We now introduce the *update function* and *firing predicate*, which respectively denote the updated phase of an oscillator $i$ at time $t$ in the next moment in time, and the firing of oscillator $i$ at time $t$,

$$update_i(t) = 1 + \text{ref}(\phi_i(t), \Delta(\phi_i(t), \alpha_i(t), \epsilon)) \quad (1)$$

$$fire_i(t) = update_i(t) > T. \quad (2)$$

The phase evolution of an oscillator $i$ over time is given by

$$\phi_i(t+1) = \begin{cases} 1 & \text{if } fire_i(t) \\ update_i(t) & \text{otherwise.} \end{cases} \quad (3)$$

## IV. POPULATION MODEL

Let $\Delta$ be a phase response function for a network of $N$ identical oscillators, where each oscillator is coupled to all other oscillators, and where the coupling strength is given by the constant $\epsilon$. Each oscillator has a phase in $1, \ldots, T$, and a refractory period defined by $R$. The probability of broadcast failure in the network is $\mu \in [0,1]$. We define a *population model* of the network as $\mathcal{S} = (\Delta, N, T, R, \epsilon, \mu)$. Oscillators

in our model have identical dynamics, and two oscillators are indistinguishable if they share the same phase. We therefore encode the global state of the model as a tuple $\langle k_1, \ldots, k_T \rangle$ where each $k_\Phi$ is the number of oscillators with phase $\Phi$.

A *global state* of $\mathcal{S}$ is a $T$-tuple $\sigma \in \{0, \ldots, N\}^T$, where $\sigma = \langle k_1, \ldots, k_T \rangle$ and $\sum_{\Phi=1}^{T} k_\Phi = N$. We denote by $\Gamma(\mathcal{S})$ the set of all global states of $\mathcal{S}$, and will simply use $\Gamma$ when $\mathcal{S}$ is clear from the context. Fig. 2 shows four global states of a population model of $N = 8$ oscillators with $T = 10$ discrete values for their phase and a refractory period of length $R = 2$. For example $\sigma_0 = \langle 2, 1, 0, 0, 5, 0, 0, 0, 0, 0 \rangle$ is the global state where two oscillators have a phase of one, one oscillator has a phase of two, and five oscillators have a phase of five. The starred node indicates the number of oscillators with phase ten that will fire in the next moment in time, while the shaded nodes indicate oscillators with phases that lie within the refractory period (one and two). If no oscillators have some phase $\Phi$ then we omit the 0 in the corresponding node.

We distinguish between states where one or more oscillators are about to fire, and states where no oscillators will fire at all. We refer to these states as *firing states* and *non-firing* states respectively. Given a population model $\mathcal{S}$, a global state $\langle k_1, \ldots, k_T \rangle \in \Gamma$ is a *firing state* if, and only if, $k_T > 0$. We denote by $\Gamma^\mathsf{F}(\mathcal{S})$ the set of all firing states of $\mathcal{S}$, and denote by $\Gamma^\mathsf{NF}(\mathcal{S})$ the set of all non-firing states of $\mathcal{S}$. Again we will simply use $\Gamma^\mathsf{F}$ or $\Gamma^\mathsf{NF}$ when $\mathcal{S}$ is clear from the context.

### A. Successor States

We now define how the global state of a population model evolves over time. Since our population model encodes uncertainty in the form of broadcast failures, firing states may have more than one possible successor state. We denote the transition from a firing state $\sigma$ to a possible successor state $\sigma'$ by $\sigma \to \sigma'$. With every firing state $\sigma \in \Gamma^\mathsf{F}$ we associate a non-empty set of *failure vectors*, where each failure vector is a tuple of broadcast failures that could occur in $\sigma$. A *failure vector* is a $T$-tuple where the $\Phi^\text{th}$ element denotes the number of broadcast failures that occur for all oscillators with phase
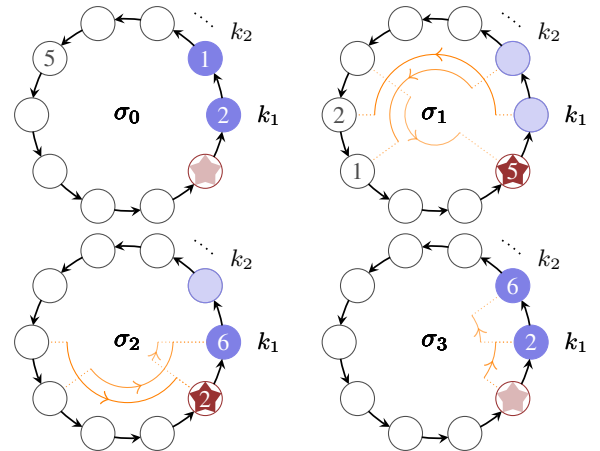


Fig. 2. Evolution of the global state over four discrete time steps.

$\Phi$. If the $\Phi^{\text{th}}$ element is $\star$ then no oscillators with a phase of $\Phi$ fired. We denote the set of all possible failure vectors by $\mathcal{F}$. Oscillators with phase less than $T$ may fire due to being perturbed by the firing of oscillators with a phase of $T$. This is discussed in detail later in this section.[1]

*a) Non-Firing States:* A non-firing state will always have exactly one successor state, as there is no oscillator that is about to fire. Therefore, the dynamics of all oscillators in that state are determined solely by the successor function. That is, the phase of every oscillator is simply updated by one in the next time step. This continues until one or more oscillators fire and perturb the phase of other oscillators. Given a sequence of global states $\sigma_0, \sigma_1, \ldots, \sigma_{n-1}, \sigma_n$ where $\sigma_0, \ldots \sigma_{n-1} \in \Gamma^{\mathsf{NF}}$ and $\sigma_n \in \Gamma^{\mathsf{F}}$, we omit transitions between $\sigma_i$ and $\sigma_{i+1}$ for $0 \le i < n$, and instead introduce a direct transition $\sigma_0 \twoheadrightarrow \sigma_n$ from the first non-firing state to the next firing state in the sequence. This is a refinement of the model presented in [25], as omitting these intermediate transitions results in smaller models. While the state space remains the same the number of transitions in the model is substantially decreased. Hence the time and resources required to check desirable properties are reduced. We denote the transition from a non-firing state $\sigma$ to its single successor state $\sigma'$ by $\sigma \twoheadrightarrow \sigma'$. For example, in Fig. 2 state $\sigma_0$ is a non-firing state, and its successor $\overrightarrow{\text{succ}}(\sigma_0) = \sigma_1$ is a firing state where all oscillator phases have been increased by 5.

*b) Encoding Chain Reactions:* For real deployments of protocols for synchronisation the effect of one or more oscillators firing may cause other oscillators to which they are coupled to fire in turn. This may then cause further oscillators to fire, and so forth, and we refer to this event as a *chain reaction*. When a chain reaction occurs it can lead to multiple groups of oscillators being triggered to fire and being *absorbed* by the initial group of firing oscillators.

These chain reactions are usually near-instantaneous events. Since we model the oscillation cycle as a progression through a number of discrete states, we choose to encode chain reactions by updating the phases of all perturbed oscillators in a single time step. Since we only consider fully-connected topologies, any oscillators sharing the same phase will always perceive the same number of other oscillators firing.

For the global state $\sigma_1$ of Fig. 2 we can see that five oscillators will fire in the next moment in time. In the successive state $\sigma_2$, the single oscillator with a phase of seven in $\sigma_1$ perceives the firing of the five oscillators. The induced perturbation causes the single oscillator to also fire and therefore be absorbed by the group of five. The remaining two oscillators with a phase of six in $\sigma_1$ perceive six oscillators to be firing, but the induced perturbation is insufficient to cause them to also fire, and they instead update their phases to ten.

*c) Firing States:* With every firing state we have by definition that at least one oscillator is about to fire in the next moment in time. Since the firing of this oscillator may, or may not, result in a broadcast failure we can see that at least two

[1]Due to space limitations we refer the reader to [25] for a detailed description of how the set of all possible failure vectors for a firing state can be constructed.

failure vectors will be associated with any firing state, and that additional failure vectors will be associated with firing states where more than one oscillator is about to fire. Given a firing state $\sigma$ and a failure vector $F$ associated with that state, we can compute the successor of $\sigma$. For each phase $\Phi \in \{1, \ldots, T\}$ we calculate the number of oscillators with a phase greater than $\Phi$ perceived to be firing by oscillators with phase $\Phi$. We simultaneously calculate $update^{\Phi}(\sigma, F)$, the updated phase of oscillators with phase $\Phi$, and $fire^{\Phi}(\sigma, F)$, the predicate indicating whether or not oscillators with phase $\Phi$ fired. Details of these constructions are given in [25].

We can then define the function that maps phase values to their updated values in the next moment in time. Since we do not distinguish between oscillators with the same phase we only calculate a single updated value for their phase. The *phase transition function* $\tau : \Gamma^{\mathsf{F}} \times \{1, \ldots, T\} \times \mathcal{F} \to \mathbb{N}$ maps a firing state $\sigma$, a phase $\Phi$, and a failure vector $F$ for $\sigma$, to the updated phase in the next moment in time, with respect to the broadcast failures defined in $F$, and is defined as

$$\tau(\sigma, \Phi, F) = \begin{cases} 1 & \text{if } fire^{\Phi}(\sigma, F) \\ update^{\Phi}(\sigma, F) & \text{otherwise.} \end{cases} \quad (4)$$

Let $\mathcal{U}_{\Phi}(\sigma, F)$ be the set of phase values $\Psi$ where all oscillators with phase $\Psi$ in $\sigma$ will have the updated phase $\Phi$ in the next time step, with respect to the broadcast failures defined in $F$. Formally, $\mathcal{U}_{\Phi}(\sigma, F) = \{\Psi \mid \Phi \in \{1, \ldots, T\} \wedge \tau(\sigma, \Psi, F) = \Phi\}$. We can now calculate the successor state of a firing state $\sigma$ and define how the model evolves over time. Observe that the population model does not encode oscillators leaving or joining the network, therefore the population $N$ remains constant. The *firing successor function* $\overrightarrow{\text{succ}} : \Gamma^{\mathsf{F}} \times \mathcal{F} \to \Gamma$ maps a firing state $\sigma$ and a failure vector $F$ to a global state $\sigma'$, and is defined as $\overrightarrow{\text{succ}}(\langle k_1, \ldots, k_T \rangle, F) = \langle k'_1, \ldots, k'_T \rangle$, where $k'_{\Phi} = \sum_{\Psi \in \mathcal{U}_{\Phi}(\sigma, F)} k_{\Phi}$ for $1 \le \Phi \le T$.

### B. Transition Probabilities

We now define the probabilities that will label the transitions in our model. Given a global state $\sigma \in \Gamma$, if $\sigma$ is a non-firing state then it has exactly one successor state. If $\sigma$ is a firing state then to construct the set of possible successor states we must first construct $\mathcal{F}_{\sigma}$, the set of all possible failure vectors for $\sigma$. Given a global state $\sigma \in \Gamma$ we define $next(\sigma)$, the set of all successor states of $\sigma$, as

$$next(\sigma) = \begin{cases} \{\overrightarrow{\text{succ}}(\sigma, F) \mid F \in \mathcal{F}_{\sigma}\} & \text{if } \sigma \in \Gamma^{\mathsf{F}} \\ \{\overrightarrow{\text{succ}}(\sigma)\} & \text{if } \sigma \in \Gamma^{\mathsf{NF}}. \end{cases} \quad (5)$$

For every non-firing state $\sigma \in \Gamma^{\mathsf{NF}}$ we have $|next(\sigma)| = 1$, since there is always exactly one successor state $\overrightarrow{\text{succ}}(\sigma)$, and we label the transition $\sigma \twoheadrightarrow \overrightarrow{\text{succ}}(\sigma)$ with probability one. We now consider each firing state $\sigma = \langle k_1, \ldots, k_n \rangle \in \Gamma^{\mathsf{F}}$, and for every successor $\overrightarrow{\text{succ}}(\sigma, F) \in next(\sigma)$, we calculate the probability that will label $\sigma \to \overrightarrow{\text{succ}}(\sigma, F)$. Recalling that $\mu$ is the probability of a broadcast failure occurring, let $\mathrm{PMF} : \{1, \ldots, N\}^2 \to [0, 1]$ be a probability mass function
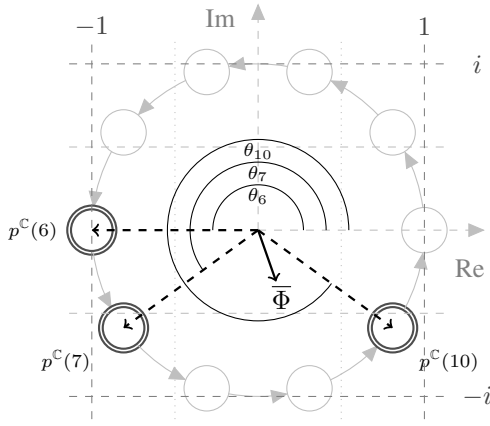
Fig. 3. Argand diagram of the phase positions for global state $\sigma_1 = \langle 0, 0, 0, 0, 0, 2, 1, 0, 0, 5 \rangle$.

where $\mathsf{PMF}(k, f) = \mu^f (1-\mu)^{k-f} \binom{k}{f}$ is the probability that $f$ broadcast failures occur given that $k$ oscillators fire. Then let $\mathsf{PFV} : \Gamma^\mathsf{F} \times \mathcal{F} \to [0, 1]$ be the function mapping a firing state $\sigma = \langle k_1, \ldots, k_T \rangle$ and a failure vector $F = \langle f_1, \ldots, f_T \rangle \in \mathcal{F}$ to the probability of the failures in $F$ occurring in $\sigma$, given by

$$\mathsf{PFV}(\sigma, F) = \prod_{\Phi=1}^{T} \begin{cases} \mathsf{PMF}(k_\Phi, f_\Phi) & \text{if } f_\Phi \neq \star \\ 1 & \text{otherwise.} \end{cases} \quad (6)$$

We can now describe the evolution of the global state over time. A *run* of a population model $\mathcal{S}$ is an infinite sequence $\sigma_0, \sigma_1, \sigma_2, \cdots$, where $\sigma_0$ is called the *initial state*, and $\sigma_{i+1} \in next(\sigma_i)$ for all $i \geq 0$.

## V. SYNCHRONISATION AND METRICS

Given a population model $\mathcal{S} = (\Delta, N, T, R, \epsilon, \mu)$, and a global state $\sigma \in \Gamma$, we say that $\sigma$ is *synchronised* if all oscillators in $\sigma$ share the same phase. We say that a run of the model $\sigma_0, \sigma_1, \sigma_2, \cdots$ *synchronises* if there exists an $i > 0$ such that $\sigma_i$ is synchronised. Note that if a state $\sigma_i$ is synchronised then any successor state $\sigma_{i+1}$ of $\sigma_i$ will also be synchronised. The population model does not encode oscillators leaving or joining the network, therefore the population $N$ remains constant. That is, the global state remains synchronised forever.

### A. Synchronisation Metric

We can extend this binary notion of synchrony by introducing a metric called *phase coherence* to quantitatively measure the level of synchrony of a global state. Our metric is derived from the *order parameter* introduced by Kuramoto [33], [34] as a measure of synchrony for a population of coupled oscillators. If we consider the phases of the oscillators as positions on the unit circle in the complex plane then we can represent the positions as complex numbers with magnitude 1. The function $p^\mathbb{C} : \{1, \ldots, T\} \to \mathbb{C}$ maps a phase value to its corresponding position on the unit circle in the complex plane, and is defined as $p^\mathbb{C}(\Phi) = e^{i\theta_\Phi}$, where $\theta_\Phi = \frac{2\pi}{T}(\Phi - 1)$. A measure of synchrony $r$ can then be obtained by calculating the magnitude of the complex number corresponding to the

mean of the phase positions. A global state has a maximal value of $r = 1$ when all oscillators are synchronised and share the same phase $\Phi$, mapped to the position defined by $p^\mathbb{C}(\Phi)$. It then follows that the mean position is also $p^\mathbb{C}(\Phi)$ and $|p^\mathbb{C}(\Phi)| = 1$. A global state has a minimal value of $r = 0$ when all of the positions mapped to the phases of the oscillators are uniformly distributed around the unit circle, or arranged such that their positions achieve mutual counterpoise. The *phase coherence function* $\mathsf{PCF} : \Gamma \to [0, 1]$ maps a global state to a real value in the interval $[0, 1]$, and is given by

$$\mathsf{PCF}(\langle k_1, \ldots, k_T \rangle) = \left| \frac{1}{N} \sum_{\Phi=1}^{T} k_\Phi p^\mathbb{C}(\Phi) \right|. \quad (7)$$

Note that for any global state $\sigma$ where $synch(\sigma)$ we have that $\mathsf{PCF}(\sigma) = 1$, since all oscillators in $\sigma$ share the same phase.

Figure 3 shows a plot on the complex plane of the positions of the phases for $N = 8$, $T = 10$, and the global state $\sigma_1 = \langle 0, 0, 0, 0, 0, 2, 1, 0, 0, 5 \rangle$. The phase positions are given by $p^\mathbb{C}(6) = e^{i\pi}$ for 2 oscillators with phase 6, $p^\mathbb{C}(7) = e^{\frac{6i\pi}{5}}$ for 1 oscillator with phase 7, and $p^\mathbb{C}(10) = e^{\frac{9i\pi}{5}}$ for 5 oscillators with phase 10. We can then determine the phase coherence as $\mathsf{PCF}(\sigma) = |\frac{1}{8}(2e^{i\pi} + e^{\frac{6i\pi}{5}} + 5e^{\frac{9i\pi}{5}})| = 0.4671$. The mean phase position is indicated on the diagram by $\overline{\Phi}$.

### B. Correspondence with Real-Valued Oscillators

Consider a clock synchronisation protocol for a cluster of $N$ fully-coupled WSN nodes, where the clocks range over real values in $[0, 2\pi]$. A *configuration* for $N$ oscillators is an $N$-tuple $\langle \theta_1, \ldots, \theta_N \rangle \in [0, 2\pi]^N$, where $\theta_j$ is the phase of oscillator $j$, for $1 \leq j \leq N$. Let $\Theta^N$ be the set of all possible configurations for $N$ oscillators. If the model for synchronisation can be defined as some phase response function $\Delta$, then given values for $R$, $\epsilon$, and $\mu$ we can construct a population model $\mathcal{S}$ that is a discrete abstraction of the continuous system. When selecting a value for $T$ there is a trade off between the size of the resulting model and the granularity of the abstraction used to represent the oscillation cycle. Since the population model $\mathcal{S}$ is an abstraction of a system of oscillators with real values for phase, and since the oscillation cycle is represented as a sequence of $T$ discrete states, then each discrete phase value $\Phi$ in the abstraction corresponds to an interval of phase values $[\frac{2\pi(\Phi-1)}{T}, \frac{2\pi\Phi}{T})$ in the continuous system, having length $\frac{2\pi}{T}$. It then follows that if some global state $\sigma = \langle k_1, \ldots, k_T \rangle \in \Gamma$ is synchronised, that is, $k_\Phi = N$ for some $1 \leq \Phi \leq T$, then this corresponds to the set $\Theta_\Phi^{N,T}$ of possible configurations for the oscillators in the continuous system, where $\Theta_\Phi^{N,T} = \{\langle \theta_1, \ldots, \theta_N \rangle \in \Theta^N \mid \theta_j \in [\frac{2\pi(\Phi-1)}{T}, \frac{2\pi\Phi}{T})$ for $1 \leq j \leq N\}$. The minimum phase coherence for all configurations in $\Theta_\Phi^{N,T}$ is then given by

$$r_{\min} = \min \left\{ \left| \frac{1}{N} \sum_{j=1}^{N} e^{i\theta_j} \right| \mid \langle \theta_1, \ldots, \theta_N \rangle \in \Theta_\Phi^{N,T} \right\}, \quad (8)$$

where the maximum phase coherence is one, since all oscillators may share the same phase. We therefore conclude that for some $N$, $T$, and $\sigma$, if the phase coherence of the discrete model $\mathsf{PCF}(\sigma) = 1$, then this corresponds to the phase coherence of the continuous system being in the interval $[r_{\min}, 1]$.

## VI. MODEL CONSTRUCTION

We use the probabilistic model checker PRISM [31] to formally verify properties of our model. Given a probabilistic model of a system, PRISM can be used to reason about temporal and probabilistic properties of the input model, by checking requirements expressed in a suitable formalism against all possible runs of the model. We define our input models as *Discrete Time Markov Chains* (DTMCs). A DTMC is a tuple $(Q, \sigma_0, \mathsf{P})$ where $Q$ is a set of states, $\sigma_0 \in Q$ is the initial state, and $\mathsf{P} : Q \times Q \to [0, 1]$ is the function mapping pairs of states $(q, q')$ to the probability with which a transition from $q$ to $q'$ occurs, where $\sum_{q' \in Q} \mathsf{P}(q, q') = 1$ for all $q \in Q$.

Given a population model $\mathcal{S} = (\Delta, N, T, R, \epsilon, \mu)$ we construct a DTMC $D(\mathcal{S}) = (Q, \sigma_0, \mathsf{P})$. We define the set of states $Q$ to be $\Gamma(\mathcal{S}) \cup \{\sigma_0\}$, where $\sigma_0$ is the initial state of the DTMC. In the initial state all oscillators are *unconfigured*. That is, oscillators have not yet been assigned a value for their phase. For each $\sigma = \langle k_1, \ldots, k_T \rangle \in Q \setminus \{\sigma_0\}$ we define

$$\mathsf{P}(\sigma_0, q) = \frac{1}{T^N} \binom{N}{k_1, \ldots, k_T} \tag{9}$$

to be the probability of moving from $\sigma_0$ to a state where $k_i$ arbitrary oscillators are configured with the phase value $i$ for $1 \leq i \leq T$. The multinomial coefficient defines the number of possible assignments of phases to distinct oscillators that result in the global state $\sigma$. The fractional coefficient normalises the multinomial coefficient with respect to the total number of possible assignments of phases to all oscillators. In general, given an arbitrary set of initial configurations (global states) for the oscillators, the total number of possible phase assignments can be calculated by computing the sum of the multinomial coefficients for each configuration (global state) in that set. Since $\Gamma$ is the set of all possible global states, we have that

$$\sum_{\langle k_1, \ldots, k_T \rangle \in \Gamma} \binom{N}{k_1, \ldots, k_T} = T^N. \tag{10}$$

We assign probabilities to the transitions as follows: for every $\sigma \in Q \setminus \{\sigma_0\}$ we consider each $\sigma' \in Q \setminus \{\sigma_0\}$ where $\sigma' = \overrightarrow{\mathrm{succ}}(\sigma, F)$ for some $F \in \mathcal{F}_\sigma$, and set $\mathsf{P}(\sigma, \sigma') = \mathsf{PFV}(\sigma, F)$. For all other $\sigma \in Q \setminus \{\sigma_0\}$ and $\sigma' \in Q$, where $\sigma \neq \sigma'$ and $\sigma' \notin next(\sigma)$, we set $\mathsf{P}(\sigma, \sigma') = 0$.

To facilitate the analysis of parameterwise-different population models we provide a Python script that allows the user to define ranges for parameters. The script then automatically generates a model for each set of parameter values, checks given properties in the model using PRISM, and writes user specified output to a comma separated value file which can be used by statistical analysis tools.[2]

### A. Reward Structures

We can annotate DTMCs with information about rewards (or costs) by assigning values to states and transitions. By

[2]The scripts to create and analyse the data, along with the verification results, can be found at https://github.com/PaulGainer/mc-bio-synch/tree/master/energy-analysis

calculating the expected value of these rewards we can reason about quantitative properties of the models. For a network of WSN nodes we are interested in the time taken to achieve a synchronised state and the power consumption of the network. Given a population model $\mathcal{S} = (\Delta, N, T, R, \epsilon, \mu)$, and its corresponding DTMC $D(\mathcal{S}) = (Q, \sigma_0, \mathsf{P})$, we define the following reward structures:

*a) Synchronisation Time:* We are interested in the average and maximum time taken for a population model to synchronise. By accumulating the reward along a path until some synchronised global state is reached we obtain a measure of the time taken to synchronise. Recall that every global state is either a firing state or a non-firing state, and for non-firing states we omit transitions to successor states where no oscillators fire; instead a transition is taken to the next global state where one or more oscillators do fire. By assigning a reward of $\frac{1}{T}$ to each transition from each firing state, and assigning a reward of $\frac{T-\delta}{T}$ to transitions from non-firing states to successor states, where $\delta$ is the highest phase of any oscillator in the non-firing state, and hence $T - \delta$ is the number of omitted transitions where no oscillators fire, we obtain a measure of synchronisation time for a population model.

*b) Power Consumption:* Let $I_I$, $I_R$, and $I_T$ be the current draw in amperes for the idle, receive, and transmit modes, $V$ be the voltage, $C$ be the length of the oscillation cycle in seconds, and $M_t$ be the time taken to transmit a synchronisation message in seconds. Let $W_I = \frac{I_I V C}{3600 T}$ and $W_R = \frac{I_R V C}{3600 T}$ be the power consumption in Watt-hours of one node for one discrete step within its oscillation cycle in idle and receive mode, and let $W_T = \frac{I_T V M_t}{3600}$ be the power consumption in Watt-hours to transmit one synchronisation message.

The function $\mathrm{pow} : Q \setminus \{\sigma_0\} \to \mathbb{R}$ maps a state to the power consumption of the network in that state, given by

$$\mathrm{pow}(\sigma) = \sum_{\Phi=1}^{R} k_\Phi W_I + \sum_{\Phi=R+1}^{T} k_\Phi W_R. \tag{11}$$

The function $\overrightarrow{\mathrm{pow}} : Q \cap \Gamma^{\mathsf{NF}} \to \mathbb{R}$ maps a non-firing state to the total power consumed by the network to reach the next firing state. Given a non firing state $\sigma = \langle k_1, \ldots, k_T \rangle$ and the maximal phase $\delta$ of any oscillator in that state, we define

$$\overrightarrow{\mathrm{pow}}(\sigma) = \sum_{j=0}^{(T-\delta)-1} \left( \sum_{\Phi=1}^{R-j} k_\Phi W_I + \sum_{\Phi=(R+1)-j}^{\delta} k_\Phi W_R \right). \tag{12}$$

From a non-firing state $\sigma \in Q \cap \Gamma^{\mathsf{NF}}$ the power consumed by the network to reach the next firing state is equivalent to the accumulation of the power consumption of the network in $\sigma$ and any successive non-firing states that are omitted in the transition from $\sigma$ to $\overrightarrow{\mathrm{succ}}(\sigma)$. Furthermore, for each firing state $\sigma \in Q \cap \Gamma^{\mathsf{F}}$ we assign a reward of $k_1 W$ to every transition from $\sigma$ to a successor state $\sigma' = \langle k_1, \ldots, k_T \rangle$. This corresponds to the total power consumption for the transmission of $k_1$ synchronisation messages.

## B. Restabilisation

A network of oscillators is *restabilising* if it has reached a synchronised state, synchrony has been lost due to the occurrence of some external event, and the network must then again achieve synchrony. We could, for instance, imagine the introduction of additional nodes with arbitrary phases to an established and synchronised network. We define the parameter $U$ to be the number of oscillators with arbitrary phase values that have been introduced into a network of $N - U$ synchronised oscillators, or to be the number of oscillators in a network of $N$ oscillators whose clocks have reset to an arbitrary value, where $U \in \mathbb{N}$ and $1 \leq U < N$. Destabilising $U$ oscillators in this way results in configurations where *at least* $N - U$ oscillators are synchronised, since the destabilised oscillators may coincidentally be assigned the phase of the synchronised group. We can restrict the set of initial configurations by identifying the set $\Gamma_U = \{\langle k_1, \ldots, k_T \rangle \mid \langle k_1, \ldots, k_T \rangle \in \Gamma$ and $k_i \geq N - U$ for some $1 \leq i \leq T\}$, where each $\sigma \in \Gamma_U$ is a configuration for the phases such that at least $N - U$ oscillators share some phase and the remaining oscillators have arbitrary phase values.

As we decrease the value of $U$ we also decrease the number of initial configurations for the phases of the oscillators. Since our model does not encode the loss or addition of oscillators we can observe that all global states where there are less than $N - U$ oscillators sharing the same phase are unreachable by any run of the system beginning in some state in $\Gamma_U$.

## VII. Evaluation

In this section, we present the model checking results for instantiations of the model given in the previous section. To that end, we instantiate the phase response function presented in Sect. III for a specific synchronisation model, and vary the length of the refractory period $R$, coupling constant $\epsilon$, and the probability $\mu$ of broadcast failures. All of these parameters are global, since we assume a homogeneous network where all oscillators have identical dynamics and technical specifications. We use a synchronisation model where the perturbation induced by the firing of other oscillators is linear in the phase of the perturbed oscillator and the number of firing oscillators [15]. That is, $\Delta(\Phi, \alpha, \epsilon) = \lfloor \Phi \cdot \alpha \cdot \epsilon \rfloor$, where $\lfloor \_ \rfloor$ denotes rounding of a value to the nearest integer. The coupling constant determines the slope of the linear dependency.

For many experiments we set $\epsilon = 0.1$ and $\mu = 0.2$. We could, of course, have conducted analyses for different values for these parameters. For a real system, the probability $\mu$ of broadcast failure occurrence is highly dependent on the deployment environment. For deployments in benign environments we would expect a relatively low rate of failure, for instance a WSN within city limits under controlled conditions, whilst a comparably high rate of failure would be expected in harsh environments such a a network of off-shore sensors below sea level. The coupling constant $\epsilon$ is a parameter of the system itself. Our results suggest that higher values for $\epsilon$ are always beneficial, however this is because we restrict our analysis to fully connected networks. High values for $\epsilon$ may be
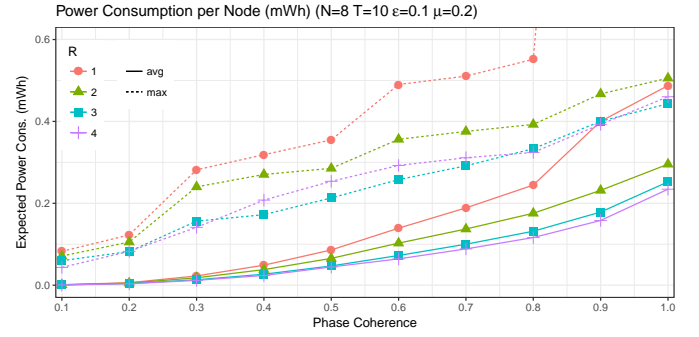


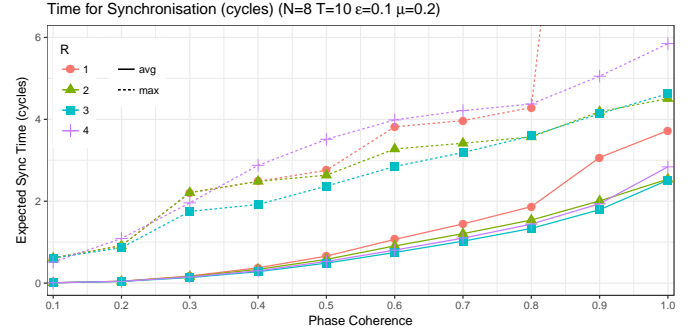Fig. 4. Power Consumption per Node to Achieve Synchronisation



Fig. 5. Time in Cycles to Achieve Synchronisation

detrimental when considering different topologies, since firing nodes may perturb synchronised subcomponents of a network. However we defer such an analysis to future work.

As an example we analyse the power consumption of the *MICAz* mote[3]. The transceiver of the MICAz mote possesses several modes. It can either transmit, receive, or remain idle. In transmit mode, it draws $17.4 \ mA$, while in receive mode, it draws $19.7 \ mA$. If the transceiver is idling it uses $20 \ \mu A$[4]. The MICAz is powered by two AA batteries or an external power supply with a voltage of $2.7 - 3.3 \ V$. For consistency, we assume that the voltage of its power supply is $3.0 \ V$.

### A. Synchronisation of a whole network

We analyse the power consumption and time to synchronise for a network of fully connected MICAz nodes. We set the size of the network to be eight oscillators with a cycle period of $T = 10$. Increasing the granularity of the cycle period, or the size of the network, beyond these values leads to models where it is infeasible to check properties due to time and memory constraints[5]. However, compared to our previous work [25],

---

[3]The technical datasheet is available at www.memsic.com/userfiles/files/Datasheets/WSN/micaz_datasheet-t.pdf

[4]The idle and transmit modes are composed of several submodes. The transmit mode has three submodes for different transmission ranges, each of which influence the amount of current draw. The current draw of the idle mode depends on whether the voltage regulator is turned on or off. To account for the worst-case, we only consider submodes with the maximal current draw.

[5]While most individual model checking runs finished within a minute, the cumulative model checking time over all analysed models was very large. The results shown in Fig. 4 already amount to 80 distinct runs.
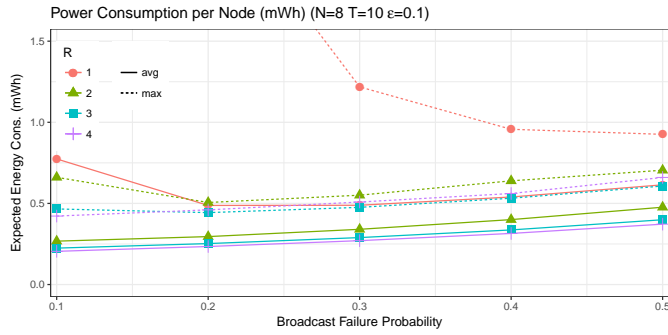
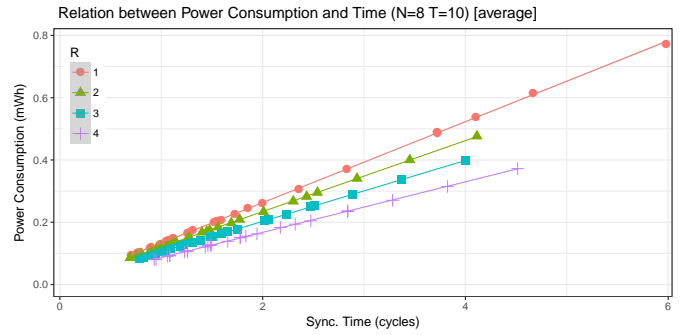Fig. 6. Power Consumption in Relation to Broadcast Failure Probability



Fig. 7. Average Power Consumption to Average Time for Synchrony



Fig. 8. Maximal Power Consumption to Maximal Time for Synchrony

we were able to increase the network size.

Figures 4 and 5 show both the average and maximal power consumption per node (in mWh) and time (in cycles) needed to synchronise, in relation to the phase coherence of the network with respect to different lengths of the refractory period, where $\epsilon = 0.1$ and $\mu = 0.2$. That is, they show how much power is consumed (time is needed, resp.) for a system in an arbitrary state to reach a state where some degree of phase coherence has been achieved. The much larger values obtained for $R = 1$ and phase coherence $\geq 0.9$ are not shown here, to avoid distortion of the figures. The energy consumption for these values is roughly $2.4mWh$, while the time needed is around 19 cycles. Observe that we only show values for the refractory period $R$ with $R < \frac{T}{2}$. For larger values of $R$ not all runs synchronise [25], resulting in an infinitely large reward being accumulated for both the maximal and average cases. We do not provide results for the minimal power consumption (or time) as it is always zero. Since we consider all initial configurations (global states) for oscillator phases there will always be a run of the system such that the phase coherence of its initial state equals or exceeds some desired degree of phase coherence $\lambda \in [0, 1]$. This follows from the observation that for any $\lambda$ there is always an initial global state $\sigma$ with phase coherence $\mathsf{PCF}(\sigma) \geq \lambda$, namely any state $\sigma'$ where all oscillators share the same phase, and hence $\mathsf{PCF}(\sigma') = 1$.

As would be expected when starting from an arbitrary state, the expected time and expected power consumption increases monotonically with the order of synchrony to be achieved. On average, networks with a higher refractory period require less power for synchronisation, and take less time to achieve it. The only exception is that the average time to achieve synchrony with a refractory period of four is higher than for two and three. However, if lower phase coherence is sufficient then this trend is stable. In contrast to that, the maximal power consumption of networks with $R = 4$ is consistently higher than of networks with $R = 3$. In addition, the maximal time needed to achieve synchrony for networks with $R = 4$ is generally higher than for lower refractory periods, except when the phase coherence is greater than or equal to 0.9. We find that networks with a refractory period of three will need the smallest amount of time to synchronise, regardless of whether we consider the maximal or average values. Furthermore, the
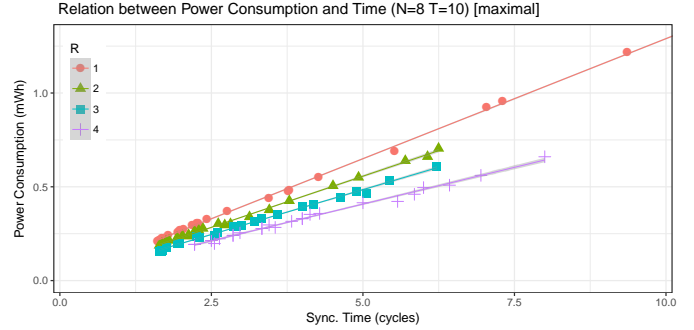
average power consumption for full synchronisation (phase coherence one) differs only slightly between $R = 3$ and $R = 4$ (less than 0.3 mWh). Hence, for the given example, $R = 3$ gives the best results. These relationships are stable even for different broadcast failure probabilities $\mu$, while the concrete values increase only slightly. This is illustrated in Fig. 6, which shows the average and maximal power consumption for different broadcast failure probabilities when $\epsilon = 0.1$.

The general relationship between power consumption and time needed to synchronise is shown in Figs. 7 and 8. Within these figures, we do not distinguish between different coupling constants and broadcast failure probabilities. We omit the two values for $R = 1$, $\epsilon = 0.1$ and $\mu \in \{0.1, 0.2\}$ in Fig. 8 to avoid distortion of the graph, since the low coupling strength and low probability of broadcast failure leads to longer synchronisation times and hence higher power consumption. While this might seem surprising it has been shown that uncertainty in discrete systems often aids convergence [25], [37].

It is easy to see that the relationship between power consumption and time to synchronise is linear, and that the slope of the relation decreases for higher refractory periods. While the linearity of the relation is almost perfect for the average values, the maximal values have greater variation. These relationships again suggest that $R = 3$ is a sensible and reliable choice for the length of refractory period, since it provides the greatest stability of power consumption and time to synchronise. In particular, if the broadcast failure probability changes, the variations in power consumption and
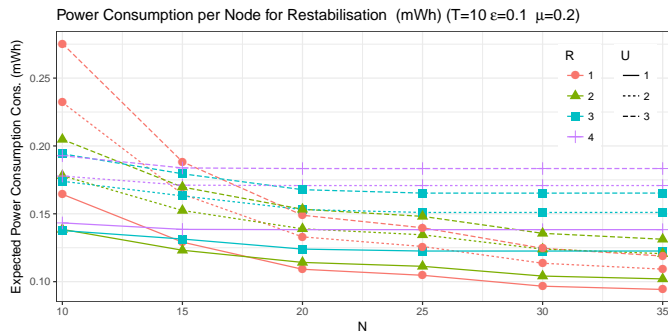
Fig. 9. Average Power Consumption for Resynchronisation to Network Size

synchronisation time are less severe for $R = 3$ than for the other refractory period lengths.

### B. Resynchronisation of a small number of nodes

In this section, we present an analysis of the power consumption if the number of redeployed nodes is small compared to the size of the network. The approach presented in Sect. VI-B allows us to significantly increase the network size. In particular, the smallest network we analyse is already larger than that in Sect. VII-A, while the largest is almost five times as large. This is possible because the model checker only needs to construct a much smaller number of initial states.

The average power consumption per node for networks of size $10, 15, \ldots, 35$, where the oscillators are coupled with strength $\epsilon = 0.1$ and the probability of broadcast failures is $\mu = 0.2$ is shown in Fig. 9. The solid lines denote the results for a single redeployed node, while the dashed lines represent the results for the redeployment of two and three nodes, respectively. As expected, the more nodes need to resynchronise, the more energy is consumed. However, we can also extract that for higher refractory periods, the amount of energy needed is more or less stable, in particular, in case $R = 4$, which is already invariant for more than ten nodes. For smaller refractory periods, increasing the network size, decreases the average energy consumption. This behaviour can be explained as follows. The linear synchronisation model implies that oscillators with a higher phase value will be activated more and thus are more likely to fire. For example, consider a network of $N = 15$ oscillators with a cycle length of $T = 10$ and a coupling constant of $\epsilon = 0.1$, where one oscillator has to resynchronise. Furthermore, assume that the 14 synchronised oscillators have phase ten and that the single oscillator has just left its refractory period. If $R = 1$, then this means the single oscillator has a phase of two, and the perturbation is $[2 \cdot 14 \cdot 0.1] = [2.8] = 3$. Hence, it will not synchronise with the other oscillators in this cycle. However, if $R = 4$, then the perturbation is $[5 \cdot 14 \cdot 0.1] = 7$, which is large enough to let the oscillator fire as well, i.e., it is absorbed by rest of the oscillators and hence synchronised. This means that in general a larger network will force the node to resynchronise faster. The refractory period determines how large the network has to be for this effect to stabilise.

## VIII. CONCLUSION

We presented a formal model to analyse power consumption in fully connected networks of PCOs. To that end, we extended an existing model for synchrony convergence with a reward structure to reflect the energy consumption of wireless sensor nodes. Furthermore, we showed how to mitigate the state-space explosion typically encountered when model-checking. In particular, the state space can be reduced by ignoring states where there are no interactions between oscillators. When investigating the restabilisation of a small number of oscillators in an already synchronised network we can reduce the state space significantly, since only a small subset of the initial states needs to be considered. We used these techniques to analyse the power consumption for synchronisation and restabilisation of a network of MICAz motes, using the pulse-coupled oscillator model developed by Mirollo and Strogatz [14] with a linear phase response function. By using our model we were able to extend the size of the network compared with previous work [25] and discuss trade-offs between the time and power needed to synchronise for different lengths of the refractory period (or duty cycle).

Results obtained using these techniques can be used by designers of WSNs to estimate the overall energy efficiency of a network during its design phase. That is, unnecessary energy consumption can be identified and rectified before deployment of the network. Additionally, our results provide guidance for estimating the battery life expectancy of a network depending on the anticipated frequency of restabilisations. Of course, these considerations only hold for the maintenance task of synchronising the network. The energy consumption of the functional behaviour has to be examined separately.

Our current approach is restricted to fully connected networks of oscillators. While this is sufficient to analyse the behaviour of strongly connected components within a network, further investigation is needed to assess the effect of different topological properties on the network. To that end, we could use several interconnected population models thus modelling the interactions of the networks subcomponents. Furthermore, topologies that change over time are of particular interest. However, it is not obvious how we could extend our approach to consider such dynamic networks. The work of Lucarelli and Wang may serve as a starting point for further investigations [15]. Stochastic node failure, as well as more subtle models of energy consumption, present significant opportunities for future extensions. For example, in some cases, repeatedly powering nodes on and off over short periods of time might use considerably more power than leaving them on throughout.

REFERENCES

[1] S. Rhee, D. Seetharam, and S. Liu, "Techniques for minimizing power consumption in low data-rate wireless sensor networks," in *Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE*, vol. 3. IEEE, 2004, pp. 1727–1731.

[2] S. Albers, "Energy-efficient algorithms," *Communications of the ACM*, vol. 53, no. 5, pp. 86–96, 2010.

[3] S. Irani and K. R. Pruhs, "Algorithmic problems in power management," *ACM Sigact News*, vol. 36, no. 2, pp. 63–76, 2005.

[4] R. Soua and P. Minet, "A survey on energy efficient techniques in wireless sensor networks," in *Wireless and Mobile Networking Conference (WMNC), 2011 4th Joint IFIP*. IEEE, 2011, pp. 1–9.

[5] J. Oller, I. Demirkol, J. Casademont, J. Paradells, G. U. Gamm, and L. Reindl, "Has time come to switch from duty-cycled mac protocols to wake-up radio for wireless sensor networks?" *IEEE/ACM Transactions on Networking*, vol. 24, no. 2, pp. 674–687, 2016.

[6] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292 – 2330, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128608001254

[7] I.-K. Rhee, J. Lee, J. Kim, E. Serpedin, and Y.-C. Wu, "Clock synchronization in wireless sensor networks: An overview," *Sensors*, vol. 9, no. 1, pp. 56–85, 2009.

[8] Y. Taniguchi, N. Wakamiya, and M. Murata, "A distributed and self-organizing data gathering scheme in wireless sensor networks," in *6th Asia-Pacific Symposium on Information and Telecommunication Technologies*. IEEE, 2005, pp. 299–304.

[9] A. Tyrrell, G. Auer, and C. Bettstetter, "Fireflies as role models for synchronization in ad hoc networks," in *Proceedings of the 1st international conference on Bio inspired models of network, information and computing systems*. ACM, 2006, p. 4.

[10] I. Bojic, T. Lipic, and M. Kusek, "Scalability issues of firefly-based self-synchronization in collective adaptive systems," in *Proc. SASOW 2014*. IEEE, 2014, pp. 68–73.

[11] N. Lipa, E. Mannes, A. Santos, and M. Nogueira, "Firefly-inspired and robust time synchronization for cognitive radio ad hoc networks," *Computer Communications*, vol. 66, pp. 36–44, 2015.

[12] G. Werner-Allen, G. Tewari, A. Patel, M. Welsh, and R. Nagpal, "Firefly-inspired sensor network synchronicity with realistic radio effects," in *Proc. SenSys 2005*. ACM, 2005, pp. 142–153.

[13] C. Peskin, *Mathematical aspects of heart physiology*, ser. Courant Lecture Notes. Courant Institute of Mathematical Sciences, New York University, 1975.

[14] R. E. Mirollo and S. H. Strogatz, "Synchronization of pulse-coupled biological oscillators," *SIAM J. App. Math.*, vol. 50, no. 6, pp. 1645–1662, 1990.

[15] D. Lucarelli, I.-J. Wang *et al.*, "Decentralized synchronization protocols with nearest neighbor communication," in *Proc. SenSys 2004*. ACM, 2004, pp. 62–68.

[16] M. Kwiatkowska, G. Norman, and D. Parker, "Stochastic model checking," in *SFM*, vol. 7. Springer, 2007, pp. 220–270.

[17] C. Baier, L. Cloth, B. R. Haverkort, H. Hermanns, and J.-P. Katoen, "Performability assessment by model checking of markov reward models," *Formal Methods in System Design*, vol. 36, no. 1, pp. 1–36, 2010.

[18] C. Baier, C. Dubslaff, and S. Klüppelholz, "Trade-off analysis meets probabilistic model checking," in *Proceedings of the Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*. ACM, 2014, p. 1.

[19] Z. Chen, D. Zhang, R. Zhu, Y. Ma, P. Yin, and F. Xie, "A review of automated formal verification of ad hoc routing protocols for wireless sensor networks," *Sensor Letters*, vol. 11, no. 5, pp. 752–764, 2013.

[20] A. Fehnker and P. Gao, "Formal verification and simulation for performance analysis for probabilistic broadcast protocols," in *Proc. ADHOC-NOW 2006*, ser. LNCS, vol. 4104. Springer, 2006, pp. 128–141.

[21] H. Yue and J.-P. Katoen, "Leader election in anonymous radio networks: Model checking energy consumption." in *ASMTA*. Springer, 2010, pp. 247–261.

[22] F. Heidarian, J. Schmaltz, and F. Vaandrager, "Analysis of a clock synchronization protocol for wireless sensor networks," *Theor. Comput. Sci.*, vol. 413, no. 1, pp. 87–105, 2012.

[23] E. Bartocci, F. Corradini, E. Merelli, and L. Tesei, "Detecting synchronisation of biological oscillators by model checking," *Theor. Comput. Sci.*, vol. 411, no. 20, pp. 1999–2018, 2010.

[24] R. Alur and D. L. Dill, "A theory of timed automata," *Theor. Comput. Sci.*, vol. 126, no. 2, pp. 183–235, 1994.

[25] P. Gainer, S. Linker, C. Dixon, U. Hustadt, and M. Fisher, "Investigating parametric influence on discrete synchronisation protocols using quantitative model checking," in *Proc. QEST 2017*, ser. LNCS. Springer, 2017.

[26] Y. Wang, F. Nuñez, and F. J. Doyle, "Energy-efficient pulse-coupled synchronization strategy design for wireless sensor networks through reduced idle listening," *IEEE Transactions on Signal Processing*, vol. 60, no. 10, pp. 5293–5306, 2012.

[27] K. Konishi and H. Kokame, "Synchronization of pulse-coupled oscillators with a refractory period and frequency distribution for a wireless sensor network," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 18, no. 3, 2008. [Online]. Available: http://aip.scitation.org/doi/abs/10.1063/1.2970103

[28] E. A. Emerson and R. J. Trefler, "From asymmetry to full symmetry: New techniques for symmetry reduction in model checking." in *CHARME*, vol. 99. Springer, 1999, pp. 142–156.

[29] A. F. Donaldson and A. Miller, "Symmetry reduction for probabilistic model checking using generic representatives," in *ATVA*, vol. 6. Springer, 2006, pp. 9–23.

[30] P. Gainer, C. Dixon, and U. Hustadt, "Probabilistic model checking of ant-based positionless swarming," in *Proc. TAROS 2016*, ser. LNCS, vol. 9716. Springer, 2016, pp. 127–138.

[31] M. Kwiatkowska, G. Norman, and D. Parker, "Prism 4.0: Verification of probabilistic real-time systems," in *Proc. CAV 2011*, ser. LNCS, vol. 6806. Springer, 2011, pp. 585–591.

[32] B. Hull, K. Jamieson, and H. Balakrishnan, "Mitigating congestion in wireless sensor networks," in *Proc. SenSys 2004*. ACM, 2004, pp. 134–147.

[33] Y. Kuramoto, *Chemical oscillations, waves, and turbulence*. Springer Science & Business Media, 2012, vol. 19.

[34] ——, "Self-entrainment of a population of coupled non-linear oscillators," in *International Symposium on Mathematical Problems in Theoretical Physics*, ser. LNP, vol. 39. Springer, 1975, pp. 420–422.

[35] M. Kramer and A. Geraldy, "Energy measurements for MICAz node," *University of Kaiserslautern, Kaiserslautern, Germany, Technical Report KrGe06*, 2006.

[36] M. Webster, M. Breza, C. Dixon, M. Fisher, and J. McCann, "Performance evaluation of gossip-synchronization algorithms for wireless sensor networks using formal verification," Submitted for publication.

[37] N. Fatès, "Remarks on the cellular automaton global synchronisation problem," in *Proc. AUTOMATA 2015*, ser. LNCS, vol. 9099. Springer, 2015, pp. 113–126.