Synchronising Timing Signals In Cellular Sensornets Using A Hybrid Algorithm

Jonathan Tate and Iain Bate

November 2009

Abstract

Interaction between sensornet nodes and the physical environment in which they are embedded implies real-time requirements. Application tasks are divided into smaller subtasks and distributed among the constituent nodes. These subtasks must be executed in the correct place, and in the correct order, for correct application behaviour. Sensornets generally have no global clock, and incur unacceptable cost if traditional synchronisation protocols are implemented. We present a lightweight primitive which generates a periodic sequence of synchronisation events which are coordinated across large sensornets structured into clusters or cells. Two biologically-inspired mechanisms are combined; *desynchronisation* within cells, and *synchronisation* between cells. This hierarchical coordination provides a global basis for local application-driven timing decisions at each node.

1 Introduction

Sensornets are distributed systems composed of many independent nodes. Each is equipped with sensors, and limited processing and energy resources. Distributed sensing applications require data from numerous physical locations to synthesise conclusions about the physical environment.

The processing and storage overhead is too large for any single node. Consequently, we must divide the application into many small subtasks and distribute these throughout the network. This is necessary, but difficult. Consider the interrelationships within, and between, data flows and processing chains. Putting aside the feasibility and method of task division, we must consider the correctness properties of software which may be compromised when turning a large serial task into a set of smaller concurrent tasks [2]. Data must be produced, processed, and consumed, at the right place and at the right time.

Real-time sensornet applications must extract data from the physical environment, and deliver the processed results, within specified bounds of a specified time. Even if real-time properties are not explicit in the application

requirements, we require that subtasks are scheduled in the correct order. This is particularly important in energy-starved systems in which we may switch off node subsystems, or switch off entire nodes, at managed times to conserve energy.

Temporal coordination is difficult in sensornets. We cannot assume the existence of a global clock. We might harness GPS transmissions to extract global timing data, or construct a dedicated time broadcast infrastructure. However, inclusion of GPS hardware may be impossible owing to cost constraints or the physical properties of GPS antennae, and the physical location of nodes may render signal receipt impossible [10]. Maintaining dedicated central transmission infrastructure may be infeasible, or unacceptably expensive, and is contrary to the notion of a decentralised network.

It is reasonable to assume nodes are equipped with cheap commodity quartz crystal timers, or can extract timing data from the operating system scheduler. However, this is error prone. Firstly, we cannot guarantee that all nodes are initialised simultaneously, owing to deferred wakeup, event-driven wakeup, logistical difficulty, and so on. The system begins in an uncoordinated condition. Secondly, individual nodes will drift out of synchrony owing to the imperfections in individual clocks. Quartz timer crystals are manufactured with finite tolerances, usually in the order of 1×10^{-6} *seconds per second* [5]. Ambient temperature may vary across the network, inducing individual timers to run fast or slow. Longer running networks will suffer more drift.

The *Dynamic Cellular Accord Protocol* (DCAP), described in this paper, addresses this problem. We arrange for a sequence of periodic timing events to occur within each network cell, applying the *desynchronisation* principle [17]. We then arrange for the relative phase difference between adjacent cells to be minimised, applying the *synchronisation* principle [11]. Over time, each set of equivalent synchronisation events across all cells is pushed closer and closer, reaching an arbitrary specified level of equality in finite time within a given error margin bounded by the synchronisation packet length. Eventually, all cells become equivalent.

These synchronisation events can be exploited by the sensornet application designer to coordinate time- and ordering-sensitive behaviours distributed throughout the network. No global clock is required. Imperfections in local clocks fitted to individual nodes are expected and handled implicitly.

The remainder of the paper is structured as follows. Section 2 discusses related work. Section 3 enumerates the novel contributions. Section 4 defines the algorithm components and their interrelationships. Section 5 discusses algorithm features and implications. Section 6 presents experimental results and analysis. Finally, section 7 presents conclusions.

2 Related work

Many sensornet tasks and data flows are at least approximately periodic [3], typically as a consequence of periodic interaction with the physical environment. In this paper we consider distributed synchronisation protocols applied in closed finite systems of cooperating sensornet nodes.

A rich and diverse body of literature exists on the scheduling of periodic tasks in general systems; a comprehensive survey can be found in [12]. The periodic nature of sensornets suggests a *cyclic schedule* rather than a *priority-driven* or *deadline-driven* approach. A *distributed* algorithm is necessary without a central controller to enforce shared schedules. *Dynamic* algorithms are required in dynamic networks with changing populations of mobile or unreliable nodes.

Conventional wired networks may employ protocols such as the *Network Time Protocol* (NTP) [14] to synchronise the local clocks of network entities. Although NTP works well in conventional LANs, it is not well suited to sensornets. It assumes the existence of a hierarchy of *time servers*, nodes equipped with local clocks against which those of other network entities are synchronised. The increased network traffic around time servers may cause congestion, and quickly deplete the finite energy reserves of nodes in the surrounding region. Individual time servers represent single points of failure, which is unacceptable in highly dynamic or unreliable sensornets. Furthermore, if the local clocks of time servers are inaccurate, all timing within the sensornet is inaccurate.

The POCSAG pager protocol [22] is an early example of a distributed system combining many mobile nodes requiring temporal coordination. Nodes equipped with local clocks coordinate their behaviour against that of a centralised messaging architecture, equipped with its own clock. Node radio hardware is switched off at predetermined times when communication should not occur. However, all communication is strictly single-hop and unidirectional under a publisher-subscriber model, which is a poor fit for typical sensornets.

Karp *et al.* describe the *Reference-Broadcast Synchronization* (RBS) model [10], which provides *on-demand pairwise synchronization with low overhead and high precision.* RBS uses pairwise linear regressions of time-of-arrival data from a shared broadcast source. RBS assumes that communications are locally broadcast, that the delay between receiving and timestamping packets observes a Gaussian distribution, and that the maximum end-to-end delay between sender and receiver is small compared to the desired synchronisation precision. The latter assumption does not hold in multicellular sensornets where node connectivity cannot be represented by a fully-connected graph. It follows that RBS may perform poorly in larger networks where multihop paths are unavoidable owing to network size or structure.

Caccamo *et al.* [3] propose a hybrid scheduling approach for multicellular sensornets. A *Frequency Division Multiplex* (FDM) strategy allocates different channels to adjacent cells by map colouring. Within each system-wide epoch an *Earliest Deadline First* (EDF) algorithm, distributed and replicated exactly at each node in a cell, allocates a proportion of equal-length frames to intra- and inter-cellular traffic. Traffic between adjacent cell pairs is managed under strict geographic *cyclic executive*. Whereas this approach is based on well-established scheduling theory, it is inflexible and is prone to poor performance in highly dynamic networks.

PalChaudhuri *et al.* [16] define a protocol for clock synchronisation which is *adaptive* to the needs of a distributed application. It supports *relative synchronisation* where network nodes minimise the relative difference between local clocks, and *external synchronisation*. The overhead is relatively high; during each synchronisation iteration each node requires $O(n^2)$ bidirectional data packet exchange with all neighbours, and execution of a linear regression calculation. This cost may be justified if the application requires nodes to collaborate at a *specific* time, rather than the lesser requirement that they collaborate at the *same* time.

Synchronisation [11] and *desynchronisation* [17] are biologically inspired primitives in which a closed finite system of periodic oscillators converge to a steady equilibrium state. System level coordination is an emergent property of independent agents implementing simple rules. Under *synchronisation* all oscillators fire simultaneously in the steady state [15], whereas under *desynchronisation* the oscillator firing times are evenly distributed in time in the steady state [6].

DESYNC-TDMA is a TDMA algorithm based on *desynchronisation* to *perfectly interleave periodic events to occur in a round-robin schedule* in a fully-connected network [6]. Each node acts as a periodic oscillator. Synchronisation signals are exchanged with peers defined by physical connectivity rather than logical network topology. The relative phase of signals measured within cyclical epochs is used to dynamically correct perceived error. Rapid convergence on a stable limit-cycle is guaranteed under ideal conditions, but disproportionately lengthy restabilisation periods result from small signal timing perturbations or network errors.

Christensen *et al.* [4] suggest that similar approaches can be applied in self-configuring systems of highly mobile robots. The physical topography of the implicit network can change very quickly owing to the high mobility of nodes. These self-organising strategies are particularly beneficial in highly dynamic and unpredictable situations, such as *Vehicular Ad-Hoc Networks*, where less agile approaches would struggle to maintain coordinated schedules.

Wang and Aspel [24] observe that these primitives converge rapidly without global clocks, adapting automatically to changing cell population. Unlike the *Phase-Locked Loop* (PLL) and *Delay-Locked Loop* (DLL) ap-

proaches, which offer similarly predictable and lightweight synchronisation behaviour, there is no requirement to maintain continuous contact between peers in the wireless medium.

A decentralised algorithm is defined by Lucarelli and Wang [13]. An arbitrary partially connected graph of nodes applies a variant of the synchronisation seeking algorithm defined in [15]; it is not required that the network graph is fully connected. Each sensornet node acts as a periodic oscillator but propagates its synchronisation signal only to nodes that are one hop away in the network topology. Over time, the entire system converges on a *synchronised* state.

By contrast, the algorithm discussed in this paper works in conjunction with the natural hierarchical structure of cellular or clustered sensornets. Greater priority is given to local coordination within cells, allowing effective collaborative working to continue within a cell regardless of temporary disruption within neighbouring cells. Exploiting this hierarchy also reduces the effective size of the problem, as there are generally fewer non-empty cells than nodes, enabling the desired level of coordination to be reached more quickly.

Many other sensornet synchronisation approaches exist; a detailed survey by Sundararaman *et al.* can be found in [18].

3 Contributions

We define the following research objectives as the primary contributions of this paper.

Obj 1: Define a lightweight primitive for global temporal coordination within a multicellular sensornet.

Obj 2: Evaluate the primitive experimentally and theoretically to assess its efficacy in typical large sensornets.

4 Algorithm

We now define the algorithm by which we obtain a periodic sequence of timing events which are globally coordinated across a multi-cellular sensornet. This is achieved by having each node produce a sequence of approximately periodic events, each sequence being of approximately equal period, and then coordinating activity between nodes so as to synchronise these event sequences.

The algorithm consists of three parts. The first part of the algorithm is responsible for producing the approximate sequence of events locally at each node, and coordinating these sequences within cells or clusters. The second part of the algorithm is responsible for coordinating these event sequences

between cells, exploiting the fact that within a cell each member has a similar view of the surrounding network, and should have a similar relationship with members of neighbouring cells. The third part merges the calculations of the first two parts, exerting influence on the locally controlled sequence at each node.

4.1 Part 1: Intracellular timing calculation

Part 1 of the algorithm is the primitive described in [20] which we refer to as the *Lightweight Improved Synchronisation Primitive* (LISP), which is in turn based on the *desynchronisation* concept [17, 6]. We summarise the definition here, but the full definition and analyses are given in [20].

LISP assumes a fully-connected network. Whereas this condition is unlikely to hold across large multi-cellular sensornets, it is reasonable to assume that the constituent nodes of a network cell are fully connected. It follows that a separate instance of LISP is implemented within each cell; parts 2 and 3, described in sections 4.2 and 4.2 respectively, are responsible for synchronising these separate instances.

LISP is able to function effectively in non-ideal environments, rejecting significant levels of timing errors such as clock drift and jitter, and communication errors such as lost and phantom synchronisation signals. Details are given in [20] but are not included here in the interests of brevity.

Assume a network cell consists of a set Σ of nodes $S_1 \cdots S_n$ where $n \geq 2$. If $n = 1$, there is obviously no need for inter-node coordination; the algorithm will function correctly but will do nothing. Each node S_i acts independently but shares an identical set of behavioural rules. The running time of the system is divided into a set of system *epochs* of equal period e such that $\forall j : E_j = e$. The sequence of system epochs E_j is defined by the natural ordering of $j \in \mathbb{N}$.

Within each system epoch E_j it is required that each node $S_i \in \Sigma$ shall execute a single instance of a periodic synchronisation event V_i exactly once. These events are used only by the protocols described in this paper, and are not related to any events used by the sensornet application. All events V_i are periodic with identical period $p_i = e$. The occurrence of a specific event at a specific node i within a specific system epoch j is labelled V_{ij}. It is required that all events V_{ij} are executed within epoch E_j.

Each node $S_i \in \Sigma$ has a local clock used to measure the *local phase* ϕ_i which increases from 0 to ϕ_{max} in time $p_i = e$. When $\phi_i = \phi_{max}$ at node S_i in epoch E_j, node S_i transmits a *synchronisation message* and resets its local phase as $\phi_i = 0$, corresponding to event V_{ij}. No global clock is required. Peer nodes $T \in (\Sigma \setminus S_i)$ receive the V_{ij} synchronisation message at their local phase ψ_{ij} but do not know the identity of S_i.

Within every epoch E_j all nodes S_i record the local phase of peer node synchronisation messages, using this information to modulate their local

phase to coordinate behaviour within the cell [15]. *Desynchronisation* protocols maximise the time between synchronisation events for all nodes in a given epoch, converging on an *equilibrium state* in which synchronisation events occur spaced evenly in time [6].

Each node S_i records the local phase of the peer synchronisation events $V_{i\beta}$ and $V_{i\gamma}$, occurring immediately before and immediately after the local synchronisation event V_i respectively, and discarding all others. The corresponding peer nodes $S_{i\beta}$ and $S_{i\gamma}$ are labelled the *phase neighbours* of node S_i. The phase difference between V_i and $V_{i\beta}$ is calculated as $\phi_{i\beta}$, and the phase difference between V_i and $V_{i\gamma}$ is calculated as $\phi_{i\gamma}$. Note that $\phi_{i\beta}$ is always negative and $\phi_{i\gamma}$ always positive owing to natural ordering of events.

The *phase error* $\theta_i = \phi_{i\beta} + \phi_{i\gamma}$ is the phase difference between the local synchronisation event V_i and the target midpoint of phase neighbour synchronisation events $V_{i\beta}$ and $V_{i\gamma}$. Each node S_i alters its local phase by $\Delta\phi_i = -f_\alpha\theta_i$ upon observing $V_{i\gamma}$, where $f_\alpha \in (0, 1]$ is the *feedback proportion* governing the balance between responsiveness and stability. Given an otherwise unchanging network, $\forall i : ||\Delta\phi_{ij}|| \to 0$ as $j \to \infty$ in successive epochs [15]. Observe that $\forall i : \theta_i = 0$ in the *desynchronised equilibrium state*. If phase error $\theta_i = 0$ then phase change $\Delta\phi_i = 0$ also; no action is required when the system has converged.

Algorithm 1 defines the primitive behaviour executing at each node $S_i \in \Sigma$ under the original version of the primitive. Variables not defined within the algorithm itself take the standard meanings used elsewhere in this document.

4.2 Part 2: Intercellular timing calculation

Part 2 of the algorithm is referred to as the *Dynamic Cellular Accord Protocol* (DCAP). This works in tandem with LISP, defined in [20] and summarised in section 4.1. Whereas LISP achieves *desynchronisation* within a network cell, DCAP achieves *synchronisation* between network cells [11, 1]. *Synchrony* is the complement of *desynchrony* [17]. In this section we discuss how DCAP utilises the *sync pulse* transmissions implemented by LISP to align the relative phase of equivalent nodes located in adjacent cells to achieve the desired *synchronised equilibrium* condition.

4.2.1 Cellular network building blocks

Consider a multicellular network consisting of a set Γ of cells $C_1 \cdots C_n$. We assume any given cell is adjacent to one or more other cells, but it is not necessarily true that all cells are adjacent to all other cells. The set A contains adjacent cell pairs (C_x, C_y) where C_x and C_y are two adjacent cells. The (C_x, C_y) tuple is commutatively equal to (C_y, C_x) as all sensornet

Algorithm 1 : LISP executing at node S_i

Require: Observed predecessor sync phase, $\phi_{i\beta} = nil$
Require: Observed successor sync phase, $\phi_{i\gamma} = nil$
while monitoring local phase ϕ_i increasing over time **do**
  if sync event $\neq V_i$ observed **then**
    if $\phi_{i\gamma} = nil$ **then**
      $\phi_{i\gamma} \Leftarrow \phi_i$
      if $\phi_{i\beta} \neq nil$ **then**
        $\theta_i \Leftarrow \phi_{i\beta} + \phi_{i\gamma}$
        $\Delta\phi_i \Leftarrow -f_\alpha \theta_i$
        $\phi_i \Leftarrow (\phi_i + \Delta\phi_i) \bmod \phi_{max}$
        $\phi_{i\gamma} \Leftarrow (\phi_{i\gamma} + \Delta\phi_i) \bmod \phi_{max}$
      end if
    else
      $\phi_{i\gamma} \Leftarrow \phi_i$
    end if
  end if
  if $\phi_i \geq \phi_{max}$ **then**
    if $\phi_{i\beta} = nil$ **then**
      $\phi_{i\beta} \Leftarrow \phi_{i\gamma}$
    end if
    $\phi_{i\gamma} \Leftarrow nil$
    $\phi_i \Leftarrow 0$
    fire own sync event V_i
  end if
end while

nodes are equipped with transceivers and bidirectional message exchange between cells is possible. The reflexive (C_x, C_x) tuple is disallowed as it is meaningless for a cell to be adjacent to itself.

It is necessary for nodes in a given cell to determine which observed synchronisation transmissions originate from an adjacent cell, as per section 5.2, but it is not necessary for nodes to determine whence these extracellular transmissions originate. The set D_x contains all cells that are logically adjacent to a given cell C_x. The membership of D_x is defined by taking the subset of tuples from A in which C_x features as exactly one entity, and taking the other entity from all such tuples. The value $d_x = |D_x|$ gives the number, though not the identity, of cells adjacent to C_x. The maximum number of cells which can be adjacent to a given cell is a.

Extending the LISP definitions in section 4.1, each cell $C_x \in \Gamma$ contains a set Σ_x of n active nodes $S_{x1} \cdots S_{xn}$. A separate instance of LISP operates in each cell. Each node $S_{xy} \in \Sigma_x$ executes a single instance of a periodic event V_{xy} exactly once per system epoch with period e. This occurs when the local timer of node S_{xy} reaches the condition $\phi_{xy} = \phi_{max}$, at which point a synchronisation pulse is broadcast and the local timer is reset to $\phi_{xy} = 0$. From a global viewpoint this occurs within some system epoch

at time ψ_{xy}, although nodes have no concept of a global clock. The LISP mechanism forces these periodic events V_{xy} to be equidistantly spaced in the time domain, each separated by a delay of $\frac{e}{n}$ time units or equivalently $\frac{\phi_{max}}{n}$ phase units.

4.2.2 Intercellular phase error

Each cell contains the same number of active nodes, n, each transmitting exactly one synchronisation pulse per system epoch of length e. As per the definition of LISP given in section 4.1, in each cell these are spaced evenly in time with an interpulse delay of $\frac{e}{n}$ time units, or equivalently $\frac{\phi_{max}}{n}$ phase units. As stated in section 5.2 only the synchronisation pulse transmissions originating from extracellular nodes, in addition to a node's own transmissions, are visible to the DCAP algorithm. A node can trivially measure the delay between any two observed sync pulse transmissions using its local timer.

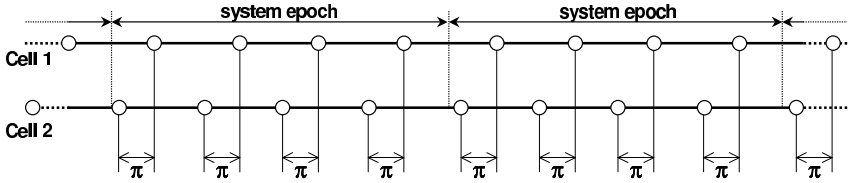

Figure 1: Intercellular phase error for nodes of two adjacent cells

Figure 1 illustrates the sequence of sync pulse transmissions in two adjacent cells over a duration of around two system epochs. Each cell has reached the *desynchronised equilibrium* state for its internal LISP instance, as defined in section 4.1. Both cells exhibit a similar sequence of periodic transmissions, equal in frequency but unequal in phase. Each synchronisation pulse transmission, and hence the node responsible for its transmission, is paired with its equivalent in the other cell. Note that this pairing is merely from the viewpoint of DCAP, and does not imply any messages are passed between paired nodes, or indeed that any paired node is aware of the identity of its partner. Also note that any pairing is transitory; the identity of the nodes paired between a pair of cells may vary from epoch to epoch.

For each node α in *Cell 1* the paired node is selected as the node β in *Cell 2* for which the interpulse delay between the *Cell 1* and *Cell 2* pulses is minimal. As the sync pulses in both cells are equivalent in all aspects except phase, the interpulse delay is identical for all paired nodes. This shared delay is labelled as the *intercellular phase error*, π. DCAP aims to minimise π by shifting sync pulse transmission times in a similar, though different, manner to that employed by LISP in section 4.1.

Provided each cell is fully desynchronised, the interpulse delay between

each node-node pair across the cellular boundary is identical and equal to π. It follows that measurement data from only one such pairing is sufficient for calculating the appropriate response, so there is no need for nodes within a cell to share measurement data. It also follows that if it is not possible for all nodes in *Cell 1* to be paired with a node in *Cell 2*, for example where there is not full connectivity between all nodes of the two adjacent cells, provided that one such pairing exists it will exert influence indirectly on all cell members. This allows DCAP to function effectively under non-ideal conditions, though of course it is advantageous for as many nodes to be paired between cells as possible.

4.2.3 Synchronised equilibrium

Section 4.1 discusses the equilibrium state for *desynchronised equilibrium*. To recap, n periodic events occur within each epoch of length e. Under *desynchronisation* these events are mutually repellent; in the *desynchronised equilibrium* state they are distributed evenly in the time domain, occurring at intervals of $\frac{e}{n}$. Under *synchronisation* the set of synchronisation events are mutually attractive; in the *synchronised equilibrium* state all n events occur simultaneously in each epoch, with a delay of e time units between each cluster of simultaneous equivalent events. It follows that $\pi = 0$ for a cell pair under the *synchronised equilibrium* state.

Sections 4.2.4 to 4.2.6 discuss the application of *synchronisation* to the problem of coordinating activity between cells of a multicellular network. We define π_{xy} as the *intercellular phase error* between cells C_x and C_y, as measured from C_x, at some time during the active lifetime of a multicellular network. It is obvious $\pi_{xy} = -\pi_{yx}$ as they measure the same magnitude of phase error, but the ordering of equivalent synchronisation events between the cells is reversed. The absolute value of these measurements tends toward zero. When the absolute value is approximately equal to zero, within some acceptable margin defined by inherent timing inaccuracy, the sign of this evanescent quantity is irrelevant.

4.2.4 Synchronisation between adjacent cells

Figure 2 illustrates system convergence on the synchronised equilibrium condition for a network consisting of 5 cells, all mutually adjacent, and all containing 5 active nodes. Each cell is depicted by one of the concentric circles; note that this is merely to illustrate the relationship between equivalent nodes in cells, and the physical regions occupied by cells do not actually overlap. Each circle represents the progress of time within a single system epoch, where the angle from the x-axis represents the progress of time from 0 to ϕ_{max} phase units. Blobs positioned around each of the concentric circles represent the firing of a synchronisation event (see section 4.1) by one of the

constituent nodes of the cell represented by the circle.

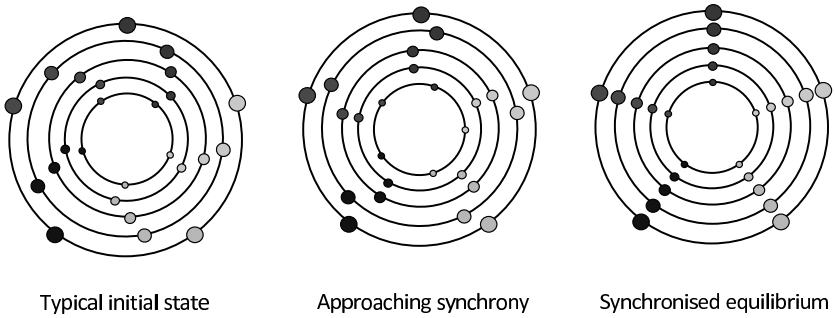

Figure 2: Progressive cell phase alignment for multiple adjacent cells

The leftmost element of figure 2 illustrates the initial condition for DCAP in which each cell is in the *desynchronised equilibrium* state (see section 4.1). There exists a non-zero *intercellular phase error* between pairs of cells (see section 4.2.2) such that $|\pi_{xy}| > 0$ between all pairs of cells in Γ.

The middle element of figure 2 illustrates the system of cells Γ after some time has passed. Within each set of *equivalent* synchronisation pulse transmissions, these have been driven closer in the time domain, but there still exists the condition $|\pi_{xy}| > 0$ between all pairs of cells in Γ.

The rightmost element of figure 2 illustrates the synchronised equilibrium condition of the system of cells Γ after further time has passed. Within each set of *equivalent* synchronisation pulse transmissions, these have been driven sufficiently close in the time domain that the condition $|\pi_{xy}| \approx 0$ is maintained between all pairs of cells in Γ within the measurement error implied by timing inaccuracies such as clock granularity and the length κ of each transmission.

Although figure 2 illustrates a system in which all i cells $C_1 \cdots C_i \in \Gamma$ are mutually adjacent, such that all possible pairings are present as tuples in the adjacency set D, this is not necessary for the system to reach the synchronised equilibrium condition under DCAP. Provided the graph of pulse-coupled oscillators is connected [15], which is true if the graph of cells is connected, the system is guaranteed to converge on the synchronised equilibrium state [1] to an arbitrary level of precision, although this may take more epoch cycles than the fully connected case.

4.2.5 Timings for equivalent events in adjacent cells

Consider a specific cell C_α with adjacent cells given by the set D_α. Each node $S_{\alpha y}$ observes both intracellular sync pulse transmissions, used by LISP, and extracellular sync pulse transmissions originating from all d_α adjacent cells $C_i \in D_\alpha$. Each node $S_{\alpha y}$ measures the time at which extracellular nodes,

anonymous to $S_{\alpha y}$, in cells $C_\beta, C_\gamma, \cdots \in D_\alpha$ broadcast sync pulses, with the difference ρ measured in phase units in the interval $[-\frac{\phi_{max}}{2}, +\frac{\phi_{max}}{2}]$. The phase difference between the sync pulse transmission of $S_{\alpha y}$ and some node $S_{\beta z}$ where $z \in [1, n]$ is given by $\rho_{\alpha y \beta z}$. It is obvious that $\rho_{\alpha y \beta z} = -\rho_{\beta z \alpha y}$ as the two events $V_{\alpha y}$ and $V_{\beta z}$ are separated by a fixed distance in the time domain and have a fixed ordering, but opposite sign depending on whether the relative measurement is taken from the viewpoint of node $S_{\alpha y}$ or from node $S_{\beta z}$.

Within the duration of a system epoch, a given node $S_{\alpha y}$ will have collected at most nd_α measurements of ρ, as there are d_α adjacent cells and each of these contains n active nodes. The set of all ρ values measured by node $S_{\alpha y}$ in epoch j is given by $R_{\alpha y j}$. The node can discard all ρ measurements at the end of each local epoch, which would place an upper bound on DCAP storage costs of na per node. This cost grows linearly in active cell population, n, and maximum adjacent cell count, a, which is a desirable property in the resource-constrained environment of sensornets.

However, for a node in a cell C_α with d_α adjacent cells, it is not necessary to store all nd_α measurements of ρ. Each node $S_{\alpha y}$ will coordinate its synchronisation event time with those of the *equivalent* nodes in adjacent cells in D_α; the measurements for all *non-equivalent* nodes in adjacent cells can be ignored. This reduces the DCAP storage overhead to d_α for a specific cell C_α, with an upper bound of a storage units in the worst case. The set of ρ values in $R_{\alpha y j}$ which originate at equivalent nodes in adjacent cells and are not discarded is given by $P_{\alpha y j}$ such that $|P_{\alpha y j}| = d_\alpha \leq a$.

The issue of discarding all ρ measurements except those d_α measurements from *equivalent* nodes in adjacent cells must now be addressed. As sync pulse transmissions are anonymous, and there is no coordination of LISP instances between cells, there is no explicit binding between equivalent nodes between two cells; the property of equivalence is an artefact of two or more nodes independently setting the firing of their LISP synchronisation events at similar times.

Recall from section 4.2.1 that the delay between consecutive synchronisation events within a cell is $\frac{e}{n}$ time units or equivalently $\frac{\phi_{max}}{n}$ phase units, when the LISP instance in that cell has reached the *desynchronised equilibrium* condition defined in section 4.1. We therefore have node $S_{\alpha y}$ consider all extracellular synchronisation pulse transmissions occurring within $\pm\frac{\phi_{max}}{2n}$ phase units of its own synchronisation event to originate at *equivalent* nodes in adjacent cells. Nodes implicitly reject non-equivalent extracellular synchronisation pulse transmissions by simply ignoring any which do not occur within these $\pm\frac{\phi_{max}}{2n}$ phase unit boundaries.

If an adjacent cell should produce more than one synchronisation event within the timing bounds outlined above, perhaps as an artefact of the adjacent cell being temporarily unstable and not yet having attained the

desynchronised equilibrium state, node $S_{\alpha y}$ has no mechanism with which to reject one or more of the resulting supernumerary synchronisation pulse transmissions. In the short term this will prevent DCAP reaching its synchronised equilibrium condition. However, the unstable cell will quickly become stable in subsequent epochs (see section 4.1) at which point this problematic condition will disappear.

4.2.6 Calculating influence of adjacent cells

Section 4.2.5 defines the activity undertaken by each node to obtain the set $P_{\alpha yj}$ of ρ values obtained by node S_y in cell C_α in epoch j, which correspond to the d_α equivalent synchronisation events in cells immediately adjacent to C_α. Recall from section 4.1 that LISP amends the local phase ϕ_i of each participating node S_i when the *successor phase neighbour* event is observed by S_i at $\phi_{i\gamma}$. From section 4.2.5 we know that, by this point, all *equivalent* synchronisation events in adjacent cells will have completed. It would be possible for nodes to adjust their local phase immediately upon observing an equivalent extracellular synchronisation transmission, but this would adversely affect the correct functioning of LISP; instead, the local phase change induced by DCAP is implemented simultaneously with that of LISP.

In section 4.1 we define that LISP calculates the value θ_i for each node S_i as the midpoint of the *predecessor* and *successor* phase neighbour synchronisation events. We now calculate ι_i as the midpoint of the *equivalent* synchronisation events in adjacent cells for node S_i, where i is shorthand for the unique identification of node S_y in cell C_α in epoch j. We can simply define $\iota_i = \text{avg}(P_{\alpha yj})$ as each node measures its ρ values relative to the firing time of its own synchronisation event, as specified in section 4.2.5.

LISP uses the feedback parameter $f_\alpha \in (0, 1]$ to specify the proportion of perceived intracellular phase error to be fed back into the system at each epoch, in order to manage the tradeoff between responsiveness and stability. DCAP uses the feedback parameter $f_\beta \in (0, 1]$ for a similar purpose. When the DCAP-driven phase adjustment is applied to a given node each node moves the timing of its own synchronisation event closer to that of the equivalent synchronisation events in adjacent nodes. Kuramoto [1] proved that such systems are guaranteed to converge on the desired synchronised equilibrium state to an arbitrary specified level of precision within finite time.

Algorithm 2 defines the DCAP ι_i calculation and local phase adjustment at each node $S_i \in \Sigma$. We assume that each node implements LISP, as defined in section 4.1. Variables not defined within the algorithm itself take the standard meanings used elsewhere in this document.

Algorithm 2 : DCAP executing at node S_i

Require: Buffer of size a, $P_{\alpha yj} = \varnothing$
Require: Intercellular feedback parameter, f_β
Require: Cell population size, n
1: **while** monitoring local phase ϕ_i increasing over time **do**
2: **if** $\phi_i \geq -\frac{\phi_{max}}{2n} \wedge \phi_i \leq +\frac{\phi_{max}}{2n}$ **then**
3: **if** extracellular synchronisation pulse is heard **then**
4: $P'_{\alpha yj} \Leftarrow P_{\alpha yj} \cup \{\phi_i\}$
5: **end if**
6: **end if**
7: **if** $\phi_i = \phi_{max}$ **then**
8: $\iota_i \Leftarrow \text{avg}(P_{\alpha yj})$
9: $\phi'_i \Leftarrow \phi_i + f_\beta \iota_i$
10: $P'_{\alpha yj} \Leftarrow \varnothing$
11: **end if**
12: **end while**

4.3 Part 3: Local event timing adjustment

Algorithm 2 describes the DCAP adjustment of node local phase ϕ_i in step 9. Similarly, LISP adjusts ϕ_i in step 7 of algorithm 1 for the original LISP variant defined in section 4.1. We can unify the DCAP and LISP effects by incorporating the DCAP influence into the LISP local phase adjustment calculation. We substitute equation 2 for equation 1 in LISP, calculate average intercellular phase error ι_i as defined in algorithm 2, omitting step 9 in algorithm 2. This is functionally equivalent to running LISP and DCAP separately; we unify for ease of analysis.

$$\Delta\phi_i \Leftarrow -f_\alpha \theta_i \tag{1}$$

$$\Delta\phi_i \Leftarrow -f_\alpha \theta_i + f_\beta \iota_i \tag{2}$$

LISP and DCAP exert influence on participating nodes simultaneously, exploiting similar coordination strategies to achieve similar but orthogonal goals. The net influence is similar to that of two partial derivatives on some measured quantity. In the following section 5.1 we consider the interaction of these influences, and the resulting consequences for selection of appropriate values of f_α and f_β.

5 Discussion

We now discuss the operation of the algorithm defined in section 4, and consider the interrelationship of the competing influences. We also consider the costs, strengths and weaknesses of the algorithm.

5.1 Interplay of influences

LISP attempts to drive each cell into the *desynchronised equilibrium* state, as defined in section 4.1, whereas DCAP attempts to drive the set of all cells into the *synchronised equilibrium* state. These goals are not contradictory, as the latter acts on sets of *equivalent* nodes rather than the set of all nodes. However, the LISP and DCAP mechanisms may suggest that the internal phase measurement should be adjusted in opposing directions. We now consider the resolution of these potentially conflicting influences.

Recall from section 4.1 that LISP takes a feedback proportion parameter f_α, and from section 4.2.6 that DCAP takes a feedback proportion parameter f_β. Each f value determines the proportion of the appropriate *phase error* which is fed back from epoch to epoch at each node. We arrange for $f_\alpha \gg f_\beta$, for example such that f_β is an order of magnitude smaller than f_α. It follows that LISP exerts greater influence per unit time than DCAP. As LISP drives nodes within a cell toward desynchronised equilibrium, DCAP exerts insignificant influence.

However, as cells approach this desynchronised equilibrium state, the amount of phase change induced by LISP in each subsequent epoch becomes smaller. Although the absolute influence exerted by DCAP does not change, the relative magnitude of the DCAP-induced phase change grows in comparison to the LISP-induced phase change. It follows that LISP achieves intracellular desynchronised equilibrium relatively quickly, and DCAP achieves intercellular synchronised equilibrium relatively slowly. In each of the intra- and inter-cellular cases, however, as the system approaches the appropriate equilibrium state the rate of change per epoch decreases, such that when equilibrium is reached no timing changes are induced by either mechanism.

Equation 2 defines the change $\Delta\phi_i$ induced as node S_i fires its synchronisation event V_{ij} in epoch j. $\Delta\phi_i$ is implemented at node S_i as an instantaneous and discrete jump in ϕ_i. From the viewpoint of all other nodes, this is equivalent to a gradual change introduced continuously at rate $\frac{d\phi_i}{dt}$ until the S_i synchronisation event V_{ij+1} in the next epoch, labelled $j+1$. This is acceptable as instances of LISP and DCAP running at each node do not interact other than through these synchronisation events.

We label the elapsed time between V_{ij} and V_{ij+1} as δt_i where t is a measure of time passing for the complete system. Differentiating equation 2 with respect to t we obtain equation 3 which describes $\frac{d\phi_i}{dt}$ for node S_i. $t_i \to e$ as the cell stabilises.

$$\frac{d\phi_i}{dt} = \frac{-f_\alpha\theta_i + f_\beta\iota_i}{\delta t_i} \tag{3}$$

As differentiation is a linear operation, we can trivially rewrite equation 3 as equation 4 and consider the θ_i and ι_i components separately.

$$\frac{d\phi_i}{dt} = -\frac{f_\alpha \theta_i}{\delta t_i} + \frac{f_\beta \iota_i}{\delta t_i} \tag{4}$$

If $f_\alpha \gg f_\beta$, it is obvious that $|f_\alpha \theta_i| \gg |f_\beta \iota_i|$ unless $\iota_i \gg \theta_i$. The influence of DCAP on ϕ_i is less significant than that of LISP until LISP approaches its stable equilibrium, as defined in section 4.1, at which point θ_i becomes small. At this point the influence of DCAP becomes significant. As all nodes sharing the cell with S_i have a similar view of neighbouring cells, all will make similar DCAP-induced adjustments in the ι_i component of ϕ_i, and hence θ_i will remain small as ϕ_i is varied by DCAP.

5.2 Intra- and extra-cellular discrimination

DCAP utilises the synchronisation pulse transmissions implied by running a distinct instance of LISP within every cell. As it does not induce any transmissions of its own it is very lightweight, does not cause additional contention for the wireless medium, and does not consume additional energy. However, it cannot function adequately if nodes cannot determine whether a given observed synchronisation pulse transmission originates from within or from without the cell. Note that this is the *only* identification information that is required. It is not necessary for a node observing a sync pulse to identify the transmitting node or cell.

As DCAP does not need to extract much information from the observed sync pulse transmission, only whether it originates from within the same cell, it is possible to achieve this without significant overhead. We assume that identifiers are assigned to each cell such that no adjacent pair of cells is assigned the same identifier. This identifier could be globally unique for the cell, perhaps derived from its geographical location, but this is not necessary. Using fewer identifiers allows a shorter encoding of identity data in each sync pulse transmission. Allocating identifiers to cells is beyond the scope of DCAP but is addressed in the literature [8].

Finding the minimal number of unique identifiers required is depends on the spatial configuration of the cells, and is an instance of the *map colouring problem*. By the Four Colour Theorem [25] any planar map requires only 4 unique cell identifiers; maps covering spherical or cylindrical surfaces are equivalent to planar surfaces. Toroidal maps require 7.

An ideal scheme would have each cell transmit on an identifier-specific frequency and have each cell listen to the frequencies associated with all of its adjacent neighbour cells. This would allow the minimal sync pulse transmission to contain no actual information; the synchronisation event time is implicitly conveyed by the transmission time, and the cell identity is implicitly conveyed by the transmission frequency. This would avoid clashes between transmitting cells, or confusion as to transmitting cell identity.

However, this is not practical for sensornet motes which have only one transceiver and are therefore only able to listen on one frequency at any given time. An alternate solution is to encode the cell identifier into each sync pulse transmission, perhaps implemented as the smallest possible packet under the appropriate network stack with a payload of nothing more than the identifier. As per section 4.2.4 these minimal packets traverse the network stack in time κ.

As a given node observes a sync pulse transmission, it decides whether that transmission originates from within or without the cell. If the former, the information is passed to the LISP protocol defined in section 4.1. If the latter, the information is passed to the DCAP protocol described in section 4.2. This identification and channelling is not strictly necessary for the correct functioning of DCAP, as the correct behaviour would be observed provided that nodes do not act on their own sync pulse transmissions, but it is necessary for the correct functioning of LISP.

5.3 Transmission clashes in adjacent cells

If communication is implemented by broadcasts in a shared medium, it is possible for two or more nodes to have packets ready for transmission simultaneously. MAC protocols are responsible for preventing simultaneous broadcast and the resulting packet loss or corruption from interference [19], although the *hidden terminal* problem [9] may render it impossible to guarantee this does not happen. The specific MAC protocol selected is not significant to DCAP. MAC protocols which prevent clashes are beyond the scope of this paper, but are widely discussed in the literature [7].

We use the broadcast times of synchronisation pulse transmissions as a proxy for the firing time of the underlying synchronisation event occurring at the broadcasting node. We must therefore give some consideration to the timing of these broadcasts. LISP requires that the sync pulse packets are of minimal length κ (see section 4.1) which minimises the opportunity for clashes or delays. Furthermore, LISP specifically seeks to spread these sync pulse transmissions as far apart in time as is possible. Transmission clashes within unicellular networks are therefore unlikely.

Avoiding transmission clashes is more complicated in a multicellular network. The DCAP protocol requires that the equivalent synchronisation pulses in each cell be brought as close together as possible in the time domain, as discussed in section 4.2.3. It is obvious that as DCAP approaches synchronised equilibrium the sync pulse transmissions originating at the equivalent nodes in a set of adjacent cells will get closer and closer, such that the MAC protocol must intervene to prevent overlapping broadcasts. If a node is transmitting, the others must wait until the wireless medium becomes free before another may begin.

It follows that DCAP cannot guarantee to reduce the *intercellular phase*

error between cells below a threshold value of $a\kappa$, where κ is the length of a sync pulse transmission and a is the maximum number of adjacent cells. At most $a+1$ nodes, one from a given cell and one from each of the a adjacent cells, must transmit. The first node claims the wireless medium and completes its broadcast. Other nodes must wait for the wireless medium to become available, the waiting process being managed by the MAC protocol. Eventually all $a+1$ transmissions complete, the final node having waited for a other transmissions to complete in at least $a\kappa$ time units. Any overhead or inefficiency of a non-perfect MAC protocol will increase this time. However, as κ is orders of magnitude smaller than e, it follows that $a\kappa$ remains small in comparison with e.

If the order in which cells are selected to make these broadcasts remains unchanged between system epochs, a small constant phase difference of at most $\frac{a\kappa}{\phi_{max}}$ will exist between pairs of cells. However, if the order is not unchanged, this phase difference will vary between epochs, inducing a small amount of jitter with the same upper bound on magnitude.

5.4 Cost analysis

LISP [20], as defined in algorithm 1 in section 4.1, requires only two items of data to be stored. As the local phase ϕ_i increases from 0 to ϕ_{max} for some given node S_i any number of intracellular pulse events might be observed, but only the first and last are retained. The first corresponds to the successor event, and the second corresponds to the predecessor event, that surround the local event of S_i. Each value will be overwritten with new data during each epoch. Therefore, the storage overhead is $O(1)$ in cell population, n. The algorithmic complexity is also $O(1)$ in n because the algorithm requires a small fixed number of steps to be executed during each epoch; there are no loops or other recursive constructs.

DCAP, as defined in algorithm 2 in section 4.2, requires at most a items of data to be stored, where a is the *cell adjacency degree*. As the local phase ϕ_i increases from 0 to ϕ_{max} for some given node S_i, the maximum number of extracellular events which might be observed is given by a where each adjacent cell contributes exactly one such event. Therefore, the storage overhead is $O(a)$ in cell adjacency degree, a. The algorithmic complexity is $O(1)$ in a because the algorithm requires a small fixed number of steps to be executed during each epoch; again, there are no loops or other recursive constructs.

These low overheads are highly desirable in typical sensornet systems which have limited resources to allocate.

6 Results

We now present experimental results obtained by implementing and measuring DCAP through simulation.

6.1 Metrics

We define the metric U_1 to measure the effectiveness of DCAP. As DCAP has the single goal of synchronising sets of equivalent synchronisation pulse transmissions between adjacent cells we need only the single metric U_1 to measure the extent to which the system of cells Γ conforms at any given time. DCAP was implemented in a modelled multicellular sensornet and evaluated by experiment.

U_1: The average magnitude of intercellular phase error π_{xy} as measured between all adjacent pairs of cells (C_x, C_y) in D. This measure is obtained by first measuring ρ_{xpyq} for all nodes $S_p \in C_x$ and all nodes $S_q \in C_y$, which are identical within measurement error, and then taking the average of these ρ values as π_{xy} to minimise the influence of measurement error as measured values are equally likely to be larger or smaller than the true values. We then find the magnitude of each π_{xy}, and take the average of all such $|\pi|$ values as U_1 for the system of all cells in Γ. Measured in *phase units*. Defined in the range $[0, \frac{\phi_{max}}{n}]$ for cells of n active nodes. The ideal value of $U_1 = 0$.

6.2 Value of U_1 as a function of time

In this section we consider the convergence of U_1 toward its limiting value of 0 as a function of time measured in system epochs of length e time units. A network of 30 cells was constructed with the cell boundaries taking the shape of a hexagonal planar tiling. Each cell has a fixed cell population $n = 10$ because this is an energy-efficient cluster size for sensornets containing hundreds to thousands of nodes [23].

In an infinite hexagonal planar tiling each hexagonal cell is surrounded by 6 neighbouring cells, but in a finite example the cells around the perimeter may have 2, 3, or 4, neighbouring cells depending on position. Real-world sensornets may take any spatial configuration as required to fit the physical environment, but in this section we are interested only in assessing the behaviour of DCAP. To ensure that each cell has all 6 neighbours we join the north and south edges, folding the plane into a cylinder, then join the east and west edges, folding the cylinder into a torus. This removes nonuniformity of cell adjacency degree, and avoids the influence of edge effects in the resulting measurements.

We set the LISP feedback parameter $f_\alpha = 0.9$ and epoch length $e = 10s$, as [20] indicates these values are effective for cells similar to those considered

here. Section 5.1 states that the value of the DCAP feedback parameter f_β should generally be significantly smaller than f_α to allow correct interoperation, so f_β was set in the range $(0, 0.1]$. Larger values $f_\beta > 0.1$ are permitted but not considered here.

Figure 3 shows the measured value of U_1 as simulated time progressed from $t = 0$ epochs for $f_\beta = \{0.0005, 0.001, 0.01\}$. It can be seen that each f_β value results in U_1 converging on the ideal value $U_1 = 0$ over time, with higher values of f_β yielding a faster decline in U_1. Over time, the cells approach the desired *synchronised equilibrium* condition discussed in section 4.2.3; the paired equivalent periodic synchronisation events *between* two adjacent cells become progressively closer in the time domain, while synchronisation events *within* a given cells remain evenly spaced. Smaller values than $f_\beta = 0.005$ also yield this desired behaviour, but more slowly, and are omitted from figure 3 for clarity.

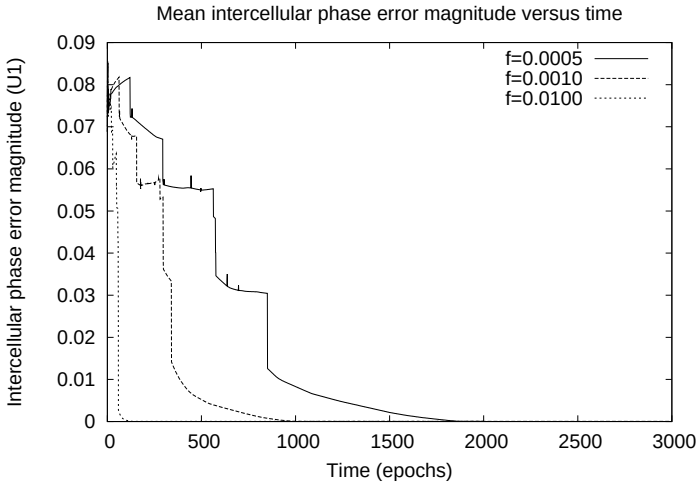

Figure 3: U_1 convergence (linear y-axis)

Figure 4 displays the same experimental data but with a logarithmic y-axis. The sharp decline observed in U_1, even under this logarithmic scale, indicates that DCAP is highly efficient at managing the firing times of sets of equivalent synchronisation events between adjacent cells. It can also be seen that higher values of f_β tend to bring about the desired synchronised equilibrium condition more quickly.

Note that the traces do not actually reach zero. This is simply an artefact of the finite time resolution of the simulation; if the calculations are performed with infinite precision then U_1 approaches 0 asymptotically. However, real sensornet hardware is also subject to similar timing artefacts as a consequence of continuous real-world time being quantised by mote timers such as CPU clocks. All synchronisation mechanisms are subject to similar quantisation artefacts.

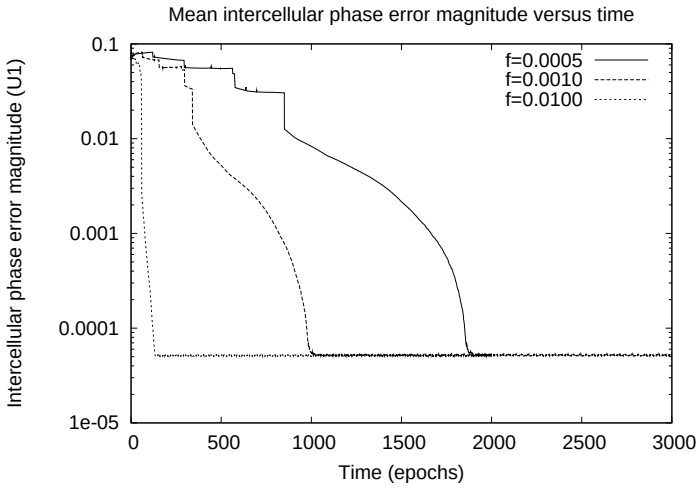

Figure 4: U_1 convergence (logarithmic y-axis)

Sensornet designers may wish to select higher values of f_β to allow the network to approach the desired stable state more quickly, but taking care to prevent $f_\beta \approx f_\alpha$ which may otherwise prevent LISP from functioning effectively. Any such f_β value leads to an equivalent stable state so the selection of a specific value is a matter of balancing efficiency and stability, though if correct behaviour of distributed applications is dependent on adjacent cells being coordinated it may be beneficial to favour the former over the latter.

6.3 Time to synchrony for varying f_β

In this section we consider the time t required for a system to reach the synchronised equilibrium condition, discussed in section 4.2.3, as a function of varying *feedback parameter*, f_β. We employ the same cellular network considered in section 6.2, again considering DCAP behaviour for $f_\beta \in (0, 0.1]$. $U_1 \to 0$ asymptotically as $t \to \infty$ so we must define a threshold value for U_1 at which point the network of cells is sufficiently converged, and compare the value of t at which this condition is reached for each value of f_β of interest.

We define the threshold condition as $U_1 < 10^{-4}$ phase units because, when converted to 10^{-3} seconds for epochs of $10s$, this is similar to the $\kappa = 0.01s$ value employed in [20] to define the limit of convergence for LISP in similar network systems. We define the system as converged at the point U_1 becomes smaller than 10^{-4} units, and does not subsequently become larger. Figure 5 illustrates the number of epochs, on the y-axis, which must pass before convergence is reached for varying f_β, given on the x-axis.

In figure 6 we see the number of epochs required is relatively large for small f_β, declining rapidly as f_β increases; after $f_\beta \approx 0.01$ little further improvement is observable. This echoes the result given in section 6.2. Ap-

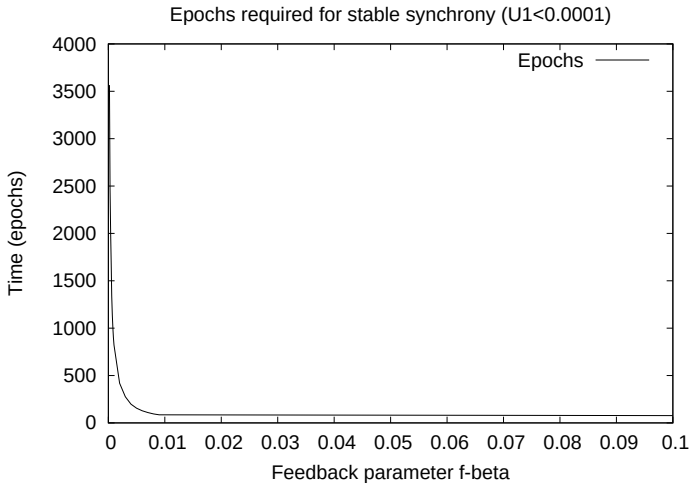

Figure 5: Epochs to equilibrium versus f_β (linear)

proximately linear relationships are observed between $\log f_\beta$ and $\log t$, with the response curve being divided into two segments of different, but always negative, gradient around $f_\beta \approx 0.01$. This matches the observation from figure 5 that comparatively small decrease in t is observed for f_β beyond this point.

If network response is reasonably insensitive to a wide range of f_β values, sensornet designers can select any value from this interval and achieve broadly similar results. It follows that sensornet designers may not need to allocate large amounts of effort into tuning DCAP to find suitable near-optimal values, enabling an efficient design process. The near-linear relationship is also useful for predicting the number of epochs required to reach synchronous equilibrium states for an arbitrary f_β during the design process.

6.4 Time to synchrony for varying a

In this section we consider the time t required for a system to reach the synchronised equilibrium condition, discussed in section 4.2.3, as a function of varying *cell adjacency degree*, a, in the range $a \in [0, b]$. For an arbitrary network b could be any value $b \in \mathbb{N}^*$, depending on the interaction between the spatial configuration and communication characteristics. We set $b = 9$, including the hexagonal cell configuration common to many planar cellular wireless networks [21] in which $a = 6$, but also allowing for other more exotic configurations which might arise in wired sensornets.

We reuse the 16-cell network considered in section 6.2. However, rather than define cell adjacency by position in a toroidal hexagonal grid, pairs of cells are randomly selected to be mutually adjacent, such that each cell is adjacent to a other cells. We measure time t to reach the stable synchronous

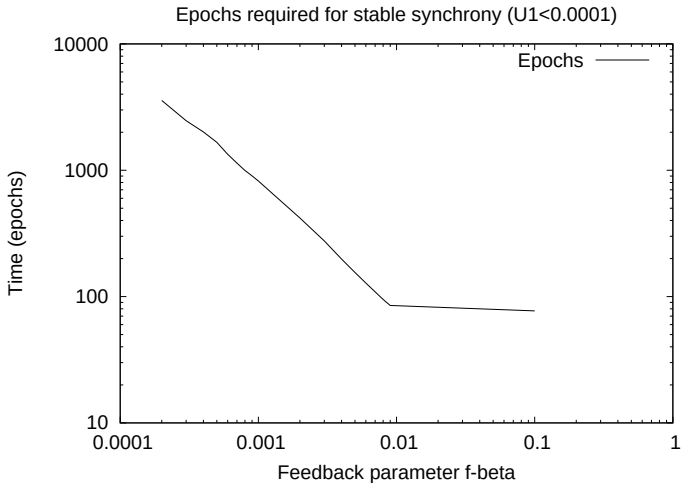

Figure 6: Epochs to equilibrium versus f_β (log)

equilibrium condition as discussed in section 6.3, but set $f_\beta = 0.01$ and vary a. We ignore the case of $a = 0$ as no intercellular interaction is possible, and the case of $a = 1$ as this creates a disconnected network in which paired cells can interact with their partners but no other intercellular interactions are possible. For $a = 2$ the network must take the form of a ring to be fully connected. For each $a \geq 3$ we take the configuration of $a - 1$ and add randomly selected cell pair adjacencies, continuing until we reach $a = b$.

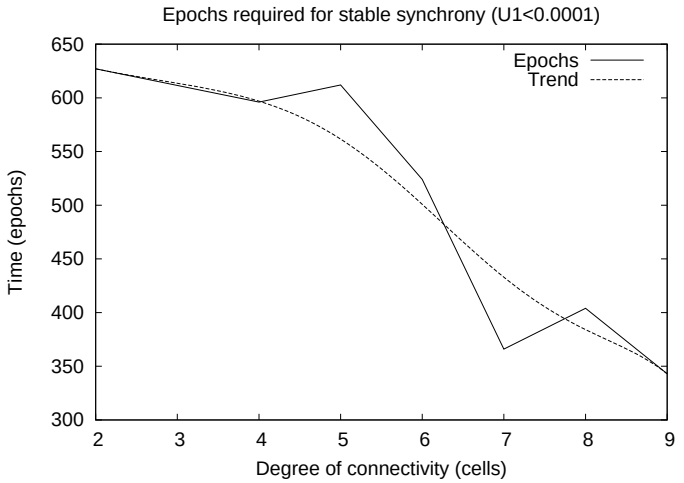

Figure 7: Epochs to stable synchronous equilibrium versus degree of cell connectivity, a

Figure 7 illustrates the relationship between t and a. In general, t tends to decline with increasing a, although the relationship is not smooth. This

is explained by the mechanism used employed by DCAP to calculate inter-cellular phase error; the greater the number of cells that are adjacent to a given cell, the greater the proportion of the complete network which is visible to nodes in that cell.

More specifically, a greater number of adjacent cells implies a greater number of measured π values (see section 4.2.2) from which to calculate local phase adjustments as per section 4.2.6. Every cell influences, and is influenced by, every other cell in a fully connected network of cells. Where two cells are directly adjacent, the mutual influence is stronger than that of two non-adjacent cells connected only indirectly through other cells. Increasing the value of a increases the extent to which cells can directly influence other cells, and hence reduces the time required for the DCAP mechanism to achieve the synchronised equilibrium state.

7 Conclusions

In section 3 a set of desired research objectives was defined, against which we now state our findings.

We first consider objective **Obj 1**. The *Dynamic Cellular Accord Protocol* (DCAP) was defined in section 4. The algorithm consists of two parts, each based on well known biologically-inspired mechanisms. The first generates a periodic sequence of synchronisation events within cells using *desynchronisation*, and the second coordinates these event sequences between cells using *synchronisation*. This enables temporal coordination in hierarchical cellular sensornets.

We now consider objective **Obj 2**. Theoretical estimates of protocol performance are defined in sections 4 and 5, against which empirical measurements are compared in section 6. The protocol achieves its aims, obtaining a globally coordinated sequence of periodic events with which local time-sensitive behaviours and decisions can be managed.

Acknowledgements

This work is partially supported by an EPSRC grant, and BAE Systems plc by the grant "Hierarchical System Management for Integrated Modular Systems".

References

[1] J. Acebrón, L. Bonilla, C. Pérez Vicente, F. Ritort, and R. Spigler. The Kuramoto model: A simple paradigm for synchronization phenomena. *Reviews of Modern Physics*, 77(1):137–185, April 2005.

[2] M. Ben-Ari. *Principles of Concurrent and Distributed Programming.* Prentice Hall, Harlow, 1990.

[3] M. Caccamo, L. Zhang, L. Sha, and G. Buttazzo. An implicit prioritized access protocol for wireless sensor networks. In *Proc. 23rd Real-Time Systems Symposium*, pages 39–48, 3-5 December 2002.

[4] D. Christensen, D. Brandt, U. Schultz, and K. Stoy. Neighbor detection and crosstalk elimination in self-reconfigurable robots. In *Proc. 1st International Conference on Robot Communication and Coordination*, pages 1–8, 15-17 October 2007.

[5] F. Cristian, H. Aghili, and R. Strong. Clock synchronization in the presence of omission and performance failures, and processor joins. In *Proc. 16th IEEE International Symposium on Fault-Tolerant Computing Systems*, pages 218–223, 1-4 July 1986.

[6] J. Degesys, I. Rose, A. Patel, and R. Nagpal. DESYNC: self-organizing desynchronization and TDMA on wireless sensor networks. In *Proc. 6th IEEE International Conference on Information Processing in Sensor Networks*, pages 11–20, 25-27 April 2007.

[7] I. Demirkol, C. Ersoy, and F. Alagoz. MAC protocols for wireless sensor networks: a survey. *IEEE Communications Magazine*, 44(4):115–121, April 2006.

[8] H. Frey and D. Görgen. Planar graph routing on geographical clusters. *Ad Hoc Networks*, 3(5):560–574, December 2005.

[9] C. Fullmer and J. Garcia-Luna-Aceves. Solutions to hidden terminal problems in wireless networks. In *Proc. 22nd ACM SIGCOMM Conference*, pages 39–49, 16-18 September 1997.

[10] R. Karp, J. Elson, C. Papadimitriou, and S. Shenker. Global synchronization in sensornets. In *Proc. 6th Latin American Symposium on Theoretical Informatics*, pages 609–624, 5-8 April 2004.

[11] Y. Kuramoto. Self-entrainment of a population of coupled non-linear oscillators. In H. Araki, editor, *Proc. International Symposium on Mathematical Problems in Theoretical Physics*, volume 39, pages 420–422, 23-29 January 1975.

[12] J. Liu. *Real-Time Systems.* Prentice Hall, Upper Saddle River, NJ, 2000.

[13] D. Lucarelli and I. Wang. Decentralized synchronization protocols with nearest neighbor communication. In *Proc. 2nd International Conference on Embedded Networked Sensor Systems*, pages 62–68, 3-5 November 2004.

[14] D. Mills. RFC 1305: Network Time Protocol (version 3): Specification, implementation and analysis. Downloaded from http://www.ietf.org/rfc/rfc1305.txt, 1992. Accessed 29/10/2009.

[15] R. Mirollo and S. Strogatz. Synchronization of pulse-coupled biological oscillators. *SIAM Journal Of Applied Mathematics*, 50(6):1645–1662, Dec 1990.

[16] S. PalChaudhuri, A. Saha, and D. Johnson. Adaptive clock synchronization in sensor networks. In *Proc. 3rd International Conference on Information Processing in Sensor Networks*, pages 340–348, 22-23 April 2004.

[17] A. Patel, J. Degesys, and R. Nagpal. Desynchronization: The theory of self-organizing algorithms for round-robin scheduling. In *Proc. 1st International Conference on Self-Adaptive and Self-Organizing Systems*, pages 87–96, 9-11 July 2007.

[18] B. Sundararaman, U. Buy, and A. Kshemkalyani. Clock synchronization for wireless sensor networks: a survey. *Ad Hoc Networks*, 3(3):281–323, 2005.

[19] A. Tanenbaum. *Computer Networks*. Pearson, Harlow, 3rd edition, March 1996.

[20] J. Tate and I. Bate. An improved lightweight synchronisation primitive for sensornets. In *Proc. 6th IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, pages 448–457, 12-15 October 2009.

[21] H. Tian and H. Shen. An optimal coverage scheme for wireless sensor network. In *Proc. 4th International Conference on Networking*, pages 722–730, April 2005.

[22] R. Tridgell. The CCIR radio-paging code no. 1: A new world standard. In *Proc. 42nd IEEE Vehicular Technology Conference*, volume 32, pages 403–406, 23-26 May 1982.

[23] D. Wang. An energy-efficient clusterhead assignment scheme for hierarchical wireless sensor networks. *International Journal of Wireless Information Networks*, 15(2):61–71, June 2008.

[24] X. Wang and A. Apsel. Pulse coupled oscillator synchronization for low power UWB wireless transceivers. In *Proc. 50th IEEE International Midwest Symposium on Circuits and Systems*, pages 1524–1527, 5-8 August 2007.

[25] R. Wilson. *Four Colours Suffice*. Allen Lane, London, 2002.