A Feedback-Driven Timing Synchronisation Protocol For Cellular Sensornets

Jonathan Tate
Department of Computer Science, University of York
York, North Yorkshire, YO10 5DD, United Kingdom
jt@cs.york.ac.uk

Iain Bate
Department of Computer Science, University of York
York, North Yorkshire, YO10 5DD, United Kingdom
iain.bate@cs.york.ac.uk

***Abstract*—Interaction between sensornet nodes and the physical environment in which they are embedded implies real-time requirements. Application tasks 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.**

I. INTRODUCTION

Sensornets are distributed systems composed of many independent nodes. Each is equipped with sensors, and limited processing and energy resources [1]. Distributed sensing applications require data from numerous physical locations to synthesise conclusions about the physical environment. Data must be produced, processed, and consumed, at the right place and right time. This is necessary, but difficult, particularly if large-scale sensornets are to be constructed.

Cellular sensornets [2] are a class of clustered sensornet [3] which is scalable, and can support predictable and dependable operation. A set of independent nodes communicate through a shared wireless medium. The physical region occupied by the sensornet is divided into a set of non-overlapping cells, such that each node resides in exactly one cell. Each node knows in which cell it resides, but does not know the full membership of its own cell, or any other cell.

The nodes of each cell form clusters [4] which cooperate to implement distributed sensing, processing, and storage. All nodes within a cell can exchange messages, forming a fully connected graph [5], and for each adjacent cell there is at least one node which can exchange messages. We assume the network of cells is not disjoint, but we do not assume the complete network of nodes is fully connected.

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. Each node is equipped with an independent local timer, such as a real-time clock or a software timer, but we cannot assume the existence of a global clock. We cannot guarantee that all nodes are initialised simultaneously, owing to logistical difficulty and deferred or event-driven node wakeup, so the system may begin in an uncoordinated condition.

Even if coordinated system start-up was possible, individual nodes would drift out of synchrony owing to imperfections in individual clocks. Quartz timer crystals are manufactured with finite tolerances [6]. Ambient temperature may vary across the network, inducing individual timers to run fast or slow. Longer running networks will suffer more drift.

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 GPS signal receipt impossible [7]. Maintaining dedicated central transmission infrastructure may be infeasible, or unacceptably expensive, and is contrary to the notion of a decentralised network central to the sensornet paradigm [1].

The primary contribution of this paper is the *Dynamic Cellular Accord Protocol* (DCAP). A pair of biologically-inspired algorithms are combined to address the problem of distributed timing synchronisation in sensornets. The first algorithm has previously been published [8], whereas the second algorithm builds upon this and is a novel contribution of this paper. Unlike other published approaches, DCAP is lightweight and fully decentralised, and thus highly suited to sensornets composed of unreliable and resource-poor nodes.

The first algorithm, LISP [8], arranges for a sequence of periodic timing events to occur within each network cell, applying the *desynchronisation* principle [9]. The second algorithm, DCAP, minimises the relative phase difference between adjacent cells, applying the *synchronisation* principle [10]. 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 to coordinate distributed time- and ordering-sensitive activity. No global clock is required. Imperfections in local clocks of individual nodes are expected and handled implicitly. If clocks are equally likely to run fast as to run slow, these imperfections are cancelled out over the sensornet.

II. RELATED WORK

Conventional wired networks may employ protocols such as the *Network Time Protocol* (NTP) [11] 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*. Increased 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. If time servers' local clocks are inaccurate, all timing within the sensornet is inaccurate.

The *Reference-Broadcast Synchronization* (RBS) [12] approach *synchronizes a set of receivers with one another, as opposed to traditional protocols in which senders synchronize with receivers*, and hence is better suited to homogeneous sensornets in which all nodes have equivalent status.

The RBS implementation described by Karp *et al.* [7] provides *on-demand pairwise synchronization with low overhead and high precision*, using pairwise linear regressions of time-of-arrival data from a shared broadcast source. It 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.

Caccamo *et al.* [13] 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*. This is inflexible and is thus prone to poor performance in highly dynamic networks.

PalChaudhuri *et al.* [14] 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 $\mathrm{O}(n^2)$ 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 [10] and *desynchronisation* [9] 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. *Synchrony* is the complement of *desynchrony* [9]. Under *synchronisation* all oscillators fire simultaneously in the steady state [10], whereas under *desynchronisation* the oscillator firing times are evenly distributed in time [15].

Lucarelli and Wang [16] define a decentralised algorithm based on *synchronisation*. An arbitrary partially connected graph of nodes applies a variant of the algorithm defined in [10]; 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. However, any natural structure or hierarchy in clustered sensornets is ignored, potentially lengthening stabilisation periods.

The *Reachback Firefly Algorithm* (RFA) [17] is another decentralised algorithm based on *synchronisation*. Rather than explicitly provide *time synchronisation*, RFA supports *synchronicity*, defined as *the ability to organize simultaneous collective action across a sensor network*. However, *time synchronisation* and *synchronicity* are complementary, and each can be derived from its complement. Each node acts as a periodic oscillator. Synchronisation signals are exchanged with peers defined by physical connectivity rather than logical network topology. However, a fully-connected network is assumed, which is unrealistic for typical sensornets.

DESYNC is an algorithm to *perfectly interleave periodic events to occur in a round-robin schedule* [15]. The underlying *synchronicity* mechanism is similar to that of RFA, with similar strengths and weaknesses, but is based on *desynchronisation* rather than *synchronisation*. The main benefit is greater precision in the time domain for coordinated action, as DESYNC events occur with greater frequency than RFA events for a given cost and network size. Rapid convergence on a stable limit-cycle is guaranteed under ideal conditions, but minor errors induce lengthy restabilisation periods.

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

III. 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, and then coordinating activity between nodes so as to synchronise these event sequences.

Each node is equipped with a single radio transceiver. We assume that all nodes, in all cells, are configured to operate on the same communications channel in the shared wireless medium. Broadcasts can be heard by other nodes within the same cell, but also spill out into neighbouring cells where they can be overheard. The same broadcasts can therefore coordinate behaviour both within cells, and between cells.

The algorithm consists of three parts, discussed in sections III-A to III-C. 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.

A. Part 1: Intracellular timing calculation

Part 1 of the algorithm is the primitive described in [8] which we refer to as the *Lightweight Improved Synchronisation Primitive* (LISP), which is in turn based on the *desynchronisation* concept [9], [15]. We summarise the definitions, but the full definition and analyses are in [8].

LISP assumes a fully-connected network cluster within each cell, which is reasonable [5] as the constituent nodes of each cell are by definition physically close. We do not assume the complete network of all cells is fully-connected. A separate instance of LISP is implemented within each cell; parts 2 and 3, described in sections III-B and III-B respectively, synchronise these separate instances.

LISP is able to function effectively in non-ideal environments. The robustness of LISP to communication errors, such as lost and phantom synchronisation signals, and timing errors, such as clock drift and jitter, is demonstrated in [8].

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}. 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.

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 $\phi_i$ 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
```

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 within the cell [10]. *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 [15].

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 [10]. 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 LISP primitive behaviour, as originally published [8], executing at each node $S_i \in \Sigma$.

B. 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 [8] and summarised in section III-A. Whereas LISP achieves *desynchronisation* among nodes within a cell, DCAP achieves *synchronisation* between entire cells. This cell-level effect is dissimilar to existing *synchronisation*-based approaches [16], [17] which operate between any arbitrary nodes in fully-connected networks. DCAP utilises the *sync pulse* transmissions implemented by LISP to synchronise the relative phase of nodes that are to act as communication conduits between adjacent cells.

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 not necessarily 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 nodes are equipped with transceivers. Bidirectional message exchange between cells is possible.

Nodes must determine which observed synchronisation transmissions originate from an adjacent cell, by methods which we discuss in section IV-B, but it is not necessary that the specific source cell be identified. 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, as defined by the cellular structure.

Extending the LISP definitions in section III-A, 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. Consider node S_{xy}, the yth node in the xth 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.

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 III-A, 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 III-B1 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.

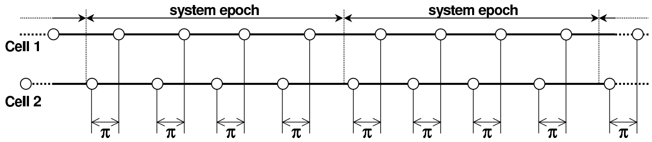


Figure 1. Intercellular phase error for nodes of two adjacent cells

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 III-A. 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. 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 III-A.

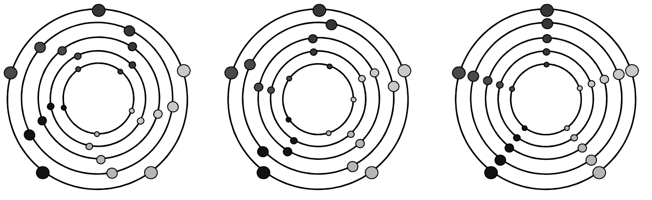
Provided each cell is fully desynchronised, the interpulse delay between each node-node pair across the cellular boundary is identical and equal to π. It follows that measurement data from only one such pairing is sufficient for calculating the appropriate response, so there is no need for nodes within 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.

3) Synchronised equilibrium: Section III-A 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 III-B4 to III-B6 discuss the application of *synchronisation* to 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) Synchronisation between adjacent cells: Figure 2 illustrates system convergence on the synchronised equilibrium condition for a 5 cell network, all mutually adjacent, 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 system epoch, where the angle from the x-axis represents time passing from 0 to ϕ_{max} phase units. Blobs positioned around each of the concentric circles represent the firing of a synchronisation event (see section III-A) by one of the constituent nodes of the cell represented by the circle.



(a) Typical initial state (b) Approaching synchrony (c) Synchronised equilibrium

Figure 2. Progressive cell phase alignment for multiple adjacent cells

Figure 2(a) illustrates the initial condition for DCAP in which each cell is in the *desynchronised equilibrium* state (see section III-A). There exists a non-zero *intercellular phase error* between pairs of cells (see section III-B2) such that $|\pi_{xy}| > 0$ between all pairs of cells in Γ.

Figures 2(b) and figure 2(c) illustrate the system of cells, Γ, as time passes following system startup. Sets of *equivalent* synchronisation pulse transmissions are driven closer in the time domain, such that $|\pi_{xy}| \to 0$. Eventually 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 [10], which is true if the graph of cells is connected, the system is guaranteed to converge on the synchronised equilibrium state to an arbitrary level of precision, although this may take more epoch cycles than the fully connected case.

5) Timings for equivalent events in adjacent cells: Consider cell C_α with adjacent cells given by the set D_α. Each node $S_{\alpha y}$ observes both intracellular sync pulses, used by LISP, and extracellular sync pulses 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 $C_\beta, C_\gamma, \dots \in D_\alpha$ 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 events $V_{\alpha y}$ and $V_{\beta z}$ are separated by a fixed delay with a fixed ordering.

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 $R_{\alpha yj}$. 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. Typical cells may have 10-20 active nodes [4], though other values are acceptable.

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 $R_{\alpha yj}$ which originate at equivalent nodes in adjacent cells and are not discarded is given by $P_{\alpha yj}$ such that $|P_{\alpha yj}| = d_\alpha \leq a$.

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 III-B1 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 III-A. 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 outside 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 owing to a temporarily unstable adjacent cell, node $S_{\alpha y}$ has no mechanism with which to reject the resulting supernumerary synchronisation 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 III-A) at which point this problematic condition will disappear.

6) Calculating influence of adjacent cells: Section III-B5 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 III-A 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 III-B5 we know that, by this point, all *equivalent* synchronisation events in adjacent cells will have completed.

In section III-A 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 III-B5.

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, damping the phase adjustment magnitude where $f_\beta < 1$.

Algorithm 2 defines the DCAP ι_i calculation and local phase adjustment at each node $S_i \in \Sigma$.

Algorithm 2 : DCAP executing at node S_i

```
Require: Buffer of size $a$, $P_{\alpha yj} = \varnothing$
Require: Intercellular feedback parameter, $f_\beta$
Require: Cell population size, $n$
 1: while monitoring local phase $\phi_i$ 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
```

C. 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 III-A. 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. Section IV-A considers the interaction of these influences, and the consequences for selection of appropriate values of f_α and f_β.

IV. DISCUSSION

A. Interplay of influences

LISP attempts to drive each cell into the *desynchronised equilibrium* state, as defined in section III-A, 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 III-A that LISP takes a feedback proportion parameter f_α, and from section III-B6 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 step change 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 measures system time. From equation 2, calculating rate of change of ϕ_i with respect to t we obtain equation 3 which gives $\frac{d\phi_i}{dt}$ for node S_i. $\delta t_i \rightarrow e$ as the cell stabilises.

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

As differentiation is a linear operation, we can trivially rewrite equation 3 as equation 4 to 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 III-A, 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.

B. Intra- and extra-cellular discrimination

DCAP 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.

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. Allocating identifiers to cells is beyond the scope of DCAP but is addressed in the literature [19].

Ideally, each cell would transmit on an identifier-specific frequency, and would listen to all frequencies associated with adjacent cells. 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 packet. Upon receiving a sync pulse packet, the node compares its own cell identifier with that encoded into the packet. If equal, the packet originates from within the same cell, and the timing information is passed to the LISP protocol defined in section III-A. Otherwise, the packet originates from a different cell, and the timing information is passed to the DCAP protocol described in section III-B.

C. Transmission clashes in adjacent cells

If communication is implemented by broadcasts in a shared medium, two or more nodes may have packets ready for transmission simultaneously. MAC protocols are responsible for preventing broadcast collisions, and minimising packet loss or corruption from interference. The specific MAC protocol selected is not significant to DCAP.

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 therefore consider the timing of these broadcasts. LISP requires that the sync pulse packets are of minimal length κ (see section III-A) which minimises the opportunity for clashes or delays. LISP specifically seeks to spread these sync pulse transmissions as far apart in time as is possible.

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 III-B3. 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.

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, 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 make these broadcasts never changes, a small constant phase difference of at most $\frac{a\kappa}{\phi_{max}}$ will exist between pairs of cells. Otherwise, this phase difference will vary between epochs, inducing a small amount of jitter with the same upper bound on magnitude.

D. Cost analysis

LISP [8], as defined in algorithm 1 in section III-A, 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 $\mathrm{O}(1)$ in cell population, n. The algorithmic complexity is also $\mathrm{O}(1)$ in n because a small, fixed number of steps are executed during each epoch; there are no loops or other recursive constructs.

DCAP, as defined in section III-B, 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 $\mathrm{O}(a)$ in cell adjacency degree, a. The algorithmic complexity is $\mathrm{O}(1)$ in a because a small, fixed number of steps are executed during each epoch; again, there are no loops or other recursive constructs.

V. RESULTS AND EVALUATION

We now implement and measure DCAP through simulation experiments. Simulation offers perfectly repeatable and perfectly controllable conditions, and is *clearly a better choice than [physical] experiments when considering large networks, as controlling large testbeds is very hard* [20]. We evaluate DCAP under non-ideal conditions including normally distributed clock drift and release jitter, and probabilistic packet loss, to ensure that our findings mirror those that would be observed in a real world deployment context.

A. Network configuration and assumptions

Cellular sensornets divide the physical deployment region into discrete, non-overlapping cells with well-defined boundaries. The spatial configuration implicitly defines intercellular relationships. In an infinite hexagonal planar tiling each hexagonal cell is surrounded by 6 neighbouring cells.

Network management packets compete with user application packets; some will be delayed or lost. We assume nodes fail to hear intracellular transmissions with uniform probability ρ_α, and intercellular transmissions with probability ρ_β. Typically, $\rho_\alpha > \rho_\beta$. We set $\rho_\alpha = 0.01$, and $\rho_\beta = 0.05$; these values can be changed to model specific environments.

Local node clocks are prone to *clock drift*, with perceived time scaling factor $\eta > 0$. Perfect clocks have $\eta = 1$. We assume each node clock has constant η [21], distributed normally as $\eta \sim N(\mu_\eta, \sigma_\eta^2)$; both μ_η and σ_η are measured in *seconds per second*. We set $\mu_\eta = 1$ to model clocks equally likely to run fast as to run slow. We set $\sigma_\eta = 1 \times 10^{-3}$ to model drift orders of magnitude greater than the 1×10^{-6} drift typical of commodity quartz crystal timers [6].

The *jitter* [22] of packet release time, ι, is the transmission time irregularity of periodic synchronisation packets, as measured against a reference clock. We assume ι is distributed normally as $\iota \sim N(\mu_\iota, \sigma_\iota^2)$; both μ_ι and σ_ι are measured in *seconds*. We set $\mu_\iota = 0$ as early transmission is as likely as late transmission. We set $\sigma_\iota = 1 \times 10^{-3}$, orders of magnitude greater than typical protocol stack propagation latency [23].

B. 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 pulses 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.

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 $U_1 = 0$.

C. Value of U_1 as a function of time

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 $n = 10$; this is an an energy-efficient cluster size for sensornets containing hundreds to thousands of nodes [4]. Trials were performed with differing cell population sizes, and no discernible difference in performance was observed.

We set the LISP feedback parameter $f_\alpha = 0.9$ and epoch length $e = 10s$, as [8] indicates these values are effective for cells similar to those considered here. Section IV-A 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. The cells approach *synchronised equilibrium*, as discussed in section III-B3; 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 $f_\beta = 0.005$ yield this behaviour, but more slowly.

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.

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.

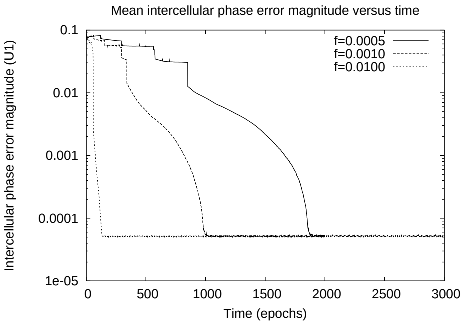


Figure 3. U_1 convergence

D. Time to synchrony for varying feedback parameter, f_β

In this section we consider the time t required for a system to reach the synchronised equilibrium condition, discussed in section III-B3, as a function of varying *feedback parameter*, f_β. We employ the same network considered in section V-C, 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 [8] to define the limit of convergence for LISP in similar network systems. We define the system as converged at the point U_1 becomes, and remains, smaller than 10^{-4} units.

In figure 4 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 V-C. Approximately 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 $f_\beta \approx 0.01$.

The near-linear relationship is useful for predicting the epochs required to reach equilibrium for an arbitrary f_β during system design. If a wide range of f_β values result in an acceptable network response, any value from this interval can be selected without incurring parameter tuning overhead.

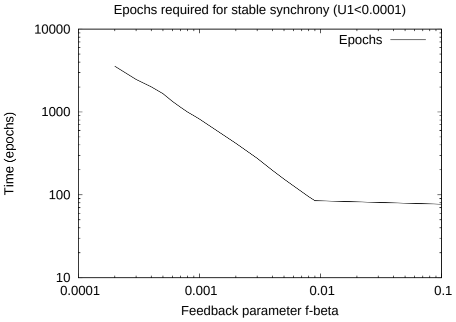


Figure 4. Epochs to equilibrium versus f_β

E. Time to synchrony for varying cell adjacency degree, a

We consider the time t required for a system to reach the synchronised equilibrium condition, discussed in section III-B3, as a function of *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 [24] in which $a = 6$, but also allowing for other more exotic configurations.

We reuse the cellular network considered in section V-C. However, 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 equilibrium condition as discussed in section V-D, but set $f_\beta = 0.01$ and vary a. We ignore the cases $a \in \{0, 1\}$ as these imply disconnected networks. 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 5 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 intercellular 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.

A greater number of adjacent cells implies a greater number of measured π values from which to calculate local phase adjustments. 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, reducing the time to reach synchronised equilibrium.

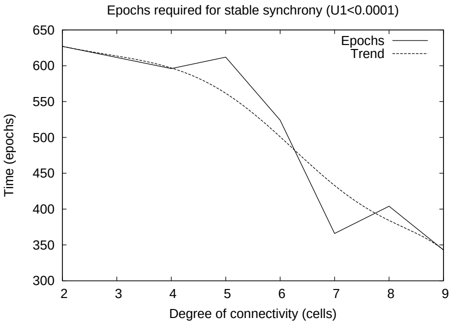


Figure 5. Epochs to equilibrium versus degree of cell connectivity, a

VI. CONCLUSIONS

The *Dynamic Cellular Accord Protocol* (DCAP) was defined in section III. The algorithm consists of two parts, each based on 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*.

Theoretical estimates of protocol performance are defined in sections III and IV, which are validated against empirical measurements in section V. The protocol achieves its aims, obtaining a globally coordinated sequence of periodic events with which local time-sensitive behaviours can be managed.

REFERENCES

[1] D. Estrin, L. Girod, G. Pottie, and M. Srivastava, "Instrumenting the world with wireless sensor networks," in *Proc. 11th IEEE International Conference on Acoustics, Speech, and Signal Processing*, 7-11 May 2001, pp. 2033–2036.

[2] G. Yu, "Cell-based coverage management and routing protocol for wireless sensor networks," in *Proc. 3rd International Conference on Mobile Technology, Applications, and Systems*, 10-12 September 2008, pp. 1–7.

[3] J. Chen, E. Shen, and Y. Sun, "The deployment algorithms in wireless sensor net works: A survey," *Information Technology Journal*, vol. 8, no. 3, pp. 293–301, 2009.

[4] D. Wang, "An energy-efficient clusterhead assignment scheme for hierarchical wireless sensor networks," *International Journal of Wireless Information Networks*, vol. 15, no. 2, pp. 61–71, June 2008.

[5] P. Costa, M. Cesana, S. Brambilla, L. Casartelli, and L. Pizziniaco, "A cooperative approach for topology control in Wireless Sensor Networks," in *Proc. 9th International Symposium on a World of Wireless, Mobile and Multimedia Networks*, 23-26 June 2008, pp. 1–10.

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

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

[8] 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*, 12-15 October 2009, pp. 448–457.

[9] 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*, July 2007, pp. 87–96.

[10] R. Mirollo and S. Strogatz, "Synchronization of pulse-coupled biological oscillators," *SIAM Journal Of Applied Mathematics*, vol. 50, no. 6, pp. 1645–1662, December 1990.

[11] D. Mills, "RFC 1305: Network Time Protocol (version 3)," http://www.ietf.org/rfc/rfc1305.txt, Newark, DE, 1992.

[12] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," in *Proc. 5th Symposium on Operating Systems Design and Implementation*, 9-11 December 2002, pp. 147–163.

[13] M. Caccamo, L. Zhang, and L. Sha, "An implicit prioritized access protocol for wireless sensor networks," in *Proc. 23rd Real-Time Systems Symposium*, December 2002, pp. 39–48.

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

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

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

[17] G. Werner-Allen, G. Tewari, A. Patel, and M. Welsh, "Firefly-inspired sensor network synchronicity with realistic radio effects," in *Proc. 3rd International Conference on Embedded Networked Sensor Systems*, 2-4 November 2005, pp. 142–153.

[18] B. Sundararaman, U. Buy, and A. Kshemkalyani, "Clock synchronization for wireless sensor networks: a survey," *Ad Hoc Networks*, vol. 3, no. 3, pp. 281–323, 2005.

[19] H. Frey, "Planar graph routing on geographical clusters," *Ad Hoc Networks*, vol. 3, no. 5, pp. 560–574, December 2005.

[20] G. Anastasi, E. Ancillotti, M. Conti, and A. Passarella, "Design and performance evaluation of a transport protocol for ad hoc networks," *The Computer Journal*, vol. 52, no. 2, pp. 186–209, March 2009.

[21] M. Sichitiu and C. Veerarittiphan, "Simple, accurate time synchronization for wireless sensor networks," in *Proc. 5th IEEE Wireless Communications and Networking Conference*, vol. 2, 16-20 March 2003, pp. 1266–1273.

[22] J. Liu, *Real-Time Systems*. Prentice Hall, 2000.

[23] C. Ee, R. Fonseca, S. Kim, D. Moon, A. Tavakoli, D. Culler, and S. Shenker, "A modular network layer for sensornets," in *Proc. 7th USENIX Symposium on Operating Systems Design and Implementation*, 6-8 November 2006, pp. 18–31.

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