A biologically-inspired clustering protocol for wireless sensor networks

S. Selvakennedy[a*], S. Sinnappan[b], Yi Shang[c]

[a]School of Information Technologies J12, University of Sydney, NSW 2006, Australia.
[b]School of Economics & Information Systems, University of Wollongong, Wollongong NSW 2522 Australia.
[c]Dept. of Computer Science, University of Missouri-Columbia, Columbia, MO 65211, USA.

Abstract

Lately, wireless sensor networks are garnering a lot of interests, as it is feasible to deploy them in many ad hoc scenarios such as for earthquake monitoring, tsunami monitoring and battlefield surveillance. As sensor nodes may be deployed in hostile areas, these battery-powered nodes are mostly expected to operate for a relatively long period. Clustering is an approach actively pursued by many groups in realizing more scalable data gathering and routing. However, it is rather challenging to form an appropriate number of clusters with well-balanced memberships. To this end, we propose a novel application of collective social agents to guide the formation of these clusters. In order to counter the usual problems of such meta-heuristics, we propose a novel atypical application that allows our protocol to converge fast with very limited overhead. An analysis is performed to determine the optimal number of clusters necessary to achieve the highest energy efficiency. In order to allow for a realistic evaluation, a comprehensive simulator involving critical components of the communication stack is used. Our protocol is found to ensure a good distribution of clusterheads through a totally distributed approach. To quantify certain clustering properties, we also introduced two fitness metrics that could be used to benchmark different clustering algorithms.

Keywords: Clustering, Routing, Biologically Inspired, Meta-Heuristic, Simulations.

1. Introduction

Wireless sensor technology is evolving rather rapidly in various aspects, as it draws significant interests from different research communities such as wireless networking, embedded system, database, data mining and distributed computing groups. For instance, in terms of the sensor node hardware, the Mica2 mote has roughly eight times the memory and communication bandwidth as its predecessor, the Rene mote, developed in 1999 for the same power budget [1]. These sensor nodes coupled with wireless communication capability have found use in many applications such as earthquake monitoring, target tracking and surveillance, structural monitoring and precision agriculture. The nodes are typically stationary due to their unique application needs, substantially more resource constrained and more densely deployed than mobile ad hoc networks (MANETs). Even though, there have been significant advances in recent years to improve wireless sensor nodes capabilities, more energy-efficient solutions are required within the communication stack and middleware for the conservation of the battery power. As the entire wireless sensor network (WSN) utility relies on its useful *lifetime*, these solutions however might come at the tradeoff against the traditional performance measures such as packet delivery ratio, mean latency and throughput. Within the communication stack, an approach that is likely to succeed in this regard is the use of a hierarchical structure for routing [2, 3].

Clustering with data aggregation is an important technique in this direction, and it makes the tradeoff between energy efficiency and data resolution. Even though many protocols have been proposed in the literature to minimize energy dissipation on the forwarding paths, some nodes may still be drained quickly. By employing a dynamic clustering technique, these nodes could lose their popularity as certain roles are rotated dynamically

[*] Corresponding author. Tel.: +61-2-9351 4494.
E-mail address: skennedy@it.usyd.edu.au.

[3]. Various clustering techniques in different context have been proposed. Most algorithms aim at generating the minimum number of clusters and transmission distance. These algorithms also distinguish themselves by how the clusterheads (CHs) are elected.

Another important design issue to consider is network reliability or robustness. Since the wireless medium and the sensor nodes themselves are considered less reliable relative to the wired technology and its tethered nodes, adopting a deterministic approach in the solution design, and assuming only average case performance would be disastrous in practice. To this end, social insect swarm behavior may provide an ideal model for the design of such less controllable systems. To our knowledge, very few researchers have considered or adopted such biologically-inspired approaches for the WSN design. However, a number of recent works has been based on different swarm behaviors in the design of routing protocols for MANETs. As there are many important similarities between these two ad hoc technologies, we believe building on these knowledge may be useful for WSNs. Most of these swarm-based routing algorithms are simple yet robust as well as adaptive to topological changes. However, such algorithms cannot establish the shortest or appropriate paths before a sufficient number of agents is flooded [4].

In this paper, we propose to biologically inspire the clustering approach, whereby the network is clustered around certain nodes deemed biologically fit. A preliminary version of this algorithm is introduced in [5]. According to this approach, when a node has a *special agent*, it self-elects itself to become a CH. Such election obviates the need to maintain many state variables. A fixed number of such agents (i.e. in biological term, the swarm size) are used to ensure that a certain number of clusters are formed throughout the network useful life. This number is derived analytically to minimize energy dissipation through data aggregation. We derive the optimal number of clusters assuming a Voronoi tessellation incorporating both the influence of intra- and inter-traffic. Based on this optimal number and our approach of clustering, we could guarantee that the network always operates close to optimal once system convergence is achieved. In addition, we could ensure that the CHs are well distributed across the network to promote load balancing. Thus, using this approach, we address a multi-objective optimization problem of energy efficiency and load balancing. To quantify the clustering properties, we also introduce two fitness metrics that will be useful for comparing any clustering algorithms.

To the best of our knowledge, this is the first attempt to adapt a biologically-inspired technique to the resource-starved device class in a unique approach. To counter the general problems of such biologically-inspired meta-heuristics whereby they requires significant number of agents and a long time for convergence, we only adopt its certain useful features, and modify the mechanics of the agents. The main features that we still adopt are:

- The notion of biological fitness of a solution;
- The use of a collection of simple agents *interacting* locally. Interaction is not achieved through explicit message exchanges, but implicitly by leaving some simple state at the nodes.

In a typical swarm intelligence use, an agent has to be present at an object (here, a node) to affect its biological fitness. In our approach, an agent is allowed to influence even from *remote*. We demonstrate later how this unique adaptation has reduced the need for a large agent population and also reduced the convergence time in our atypical application. Also, our dynamic clustering protocol exhibits fault-tolerance in the face of node failures compared against fixed or deterministic clustering approaches in the literature. Even when a node fails, this protocol could still achieve the desired properties of good CH distribution and energy efficiency.

The rest of the paper is organized as follows. Section 2 presents the perspective of this area of research. Various clustering and biologically-inspired algorithms proposed in the literature are discussed there. The basic swarm principles are presented in Section 3. In Section 4, the details of our protocol are described. The comprehensive simulator used to experiment with this algorithm is described in Section 5. Many different experiments comparing our protocol against well-known proposals and their corresponding results are presented and analyzed in Section 6. The main findings of this work are summarized in the final section with some directions for further work.

2. Related Work

Intense research in the field of sensor network technology in recent years has fueled further development in micro-sensor technology and low-power analog/digital electronics. Some of the ongoing challenging issues facing researchers are low-power signal processing architecture, routing protocol and data gathering strategy, data storage and indexing, data aggregation, low-power security protocol, and localization. It is realized that the approach that is likely to succeed to provide a scalable and energy-efficient solution is of a hierarchical structure. To this end, various clustering algorithms have been proposed in different context. Initially, these algorithms focused on the connectivity problem [6-8] but later energy-efficiency was more of interest in wireless ad hoc and sensor networks [9-15]. However, almost all focuses on reducing the number of clusters formed, which may not necessarily entail minimum energy dissipation.

Generally, clustering algorithms segment a network into non-overlapping clusters comprising a CH each. Non-CH nodes transmit sensed data to CHs, where the sensed data could be aggregated as these signals could be sufficiently correlated due to the nodes spatial proximity, and transmitted to the base-station (aka the *sink*). Clustering algorithms may be distinguished by the way the CHs are elected. The Linked Cluster Algorithm (LCA) [8] selects a CH based on the highest id among all nodes within one-hop. This is enhanced by LCA2 [16] that selects the node with the lowest id among all nodes that is neither a CH nor is one-hop of the previously selected CHs. In [17], the authors developed a similar distributed algorithm to LCA2, which identifies the CH by choosing the node with the highest degree. Other algorithms such as the Distributed Cluster Algorithm (DCA) [18] and Weighted Clustering Algorithm (WCA) [19] rely on weights to select CHs.

Load balancing heuristics are proposed in [10] and [20]. In [20], the proposed clustering algorithm is designed to ensure each cluster has equal number of nodes while keeping minimal distance between nodes and their CH. These "load-balanced" algorithms focus mainly on balancing the intra-cluster traffic load without due consideration to the external traffic. The max-min d-cluster algorithm is proposed to achieve better load balancing among CHs as well as to reduce the number of CHs as compared to LCA or LCA2 [11]. It generates d-hop clusters with a run-time of $O(d)$ rounds. In [12], some clustering algorithms are proposed to maximize the network lifetime by varying the cluster size and the duration of a node being nominated as a CH based on the assumption that the locations are known a priori. These algorithms ignore inter-cluster traffic, and also need the recognition of the whole network topology, which may not be feasible in many cases. The proposal in [21] incorporates the effect of inter-cluster traffic in the determination of the optimal sink placement that maximizes the network topological lifetime. In [22], a fixed clustering algorithm that performs energy load balancing to improve network lifetime is proposed. It also takes into consideration the interaction between clustering and routing. Two schemes were introduced. The first scheme finds the optimal cluster size and location, whereas the second allows a CH to probabilistically choose to either relay the traffic to the next-hop or to deliver it directly to the sink. It assumes a heterogeneous network, where the CH nodes have larger resources than the others.

The Low-Energy Adaptive Clustering Hierarchy (LEACH) algorithm [13] and its related extensions [14, 15, 23] use probabilistic self-election, where each sensor node has a probability p of becoming a CH in each round. It guarantees that every node becomes a CH only once in $1/p$ rounds. Such role rotation aims to distribute the energy usage for a more load-balanced operation. However, LEACH assumes all nodes are able to reach the sink directly, and requires position knowledge to perform a precise transceiver power control. Some of these algorithms were designed to generate stable clusters in environments with mobile nodes. In a typical WSN, the sensor nodes are stationary, and the instability of clusters due to mobility of these nodes may not be an issue. Drawing from the wealth of clustering proposals for both MANETs and WSNs, The Hybrid Energy-Efficient Distributed (HEED) clustering is proposed in [3] to adopt real-valued weight-based clustering. It uses residual energy as the primary clustering parameter to probabilistically elect a number of *tentative* CHs, and then advertises to their neighbors of their intentions to become CHs. Such messages include a secondary cost measure that is a function of neighbor proximity or cluster density. This secondary cost is mainly used to avoid elected CHs to be in range of each other, and to guide the regular nodes in choosing the best cluster to join. This proposal is generally able to achieve a good CH distribution across the network.

Another crucial design aspect of WSNs to consider is the network reliability and fault-tolerance. It has been demonstrated in different context that the collective behavior of social insects has many attractive features, not the least robustness and reliability through redundancy. However, only a handful WSN proposals have been inspired by nature. Due to some parallels to MANETs, we will consider some biologically-inspired algorithms proposed for this domain. The first MANET routing algorithm based on the ant colony principles is the Ant-colony-based Routing Algorithm (ARA) [24] derived from AntNet [25], an algorithm for wired networks. It exploited the *pheromone* laying behavior of ants.

Pheromone is a quality metric indicating the goodness of a path. Although pheromone evaporates over time, subsequent ants leave additional pheromone and thus reinforce the path. Ants gradually establish the shortest path between food and their nest in a fully distributed and autonomous manner. Ants are flooded towards destinations while establishing the reverse paths to the source. The fact of gradual decay of pheromone introduces a form of a negative feedback to prevent old routes from remaining in the forwarding tables when routes fall out of favor with ants. As the number of ants that completes journeys to food in a given time is larger on a shorter path than on a longer path, a shorter path accumulates more pheromone and attracts more ants. The shortest paths become preferable, and most ants use them. However, longer paths are not entirely lost as some ants may still maintain such routes. Routing schemes based on such ant colony behavior is both robust and adaptable. When the shortest route is lost due to some event, the longer routes provide alternative options.

Another MANET routing algorithm that inspired by termite is *Termite* [26]. In this approach, no exclusive agents are used though, as the agents are piggybacked onto data packets. These packets follow the pheromone trail for their destinations, and leave new pheromone for their sources. In [27], routing is established based on a swarm of different types of honeybees. This algorithm, termed *BeeAdHoc*, is a hybrid routing approach where it could either be proactive or reactive. It is shown that BeeAdHoc achieves better energy savings while still maintaining performance in terms of the traditional metrics against DSR, AODV and DSDV.

As the number of nodes grows, the number of agents required to establish the routing infrastructure may explode. A way to overcome the overhead explosion and attain scalability is by using the hierarchical routing approach. For MANETs, an adaptive Swarm-based Distributed Routing (adaptive-SDR) [28] and Mobile Ants Based Routing (MABR) [29] are proposed, whereby both schemes consider dividing the network into small clusters/zones, and then perform intra- and inter-routing. To our knowledge, the only work to have considered swarm intelligence in the wireless sensor context is [30]. They implemented a distributed network of mobile sensor nodes and controlled the nodes physical movements using swarm principles. It was demonstrated that the swarm behavior could be used to ensure *safe separation* between the agents and *coverage efficiency* while enforcing a level of *cohesion* that maintains a level of connectivity between the mobile agents. These nodes however are less resource-constrained than typical mote-class devices, and the overhead due the swarm intelligence meta-heuristic was not a major concern there.

We propose the use of both a hierarchical structure and the biological inspiration to realize a scalable and robust data routing strategy. To appreciate this integration, the basic principles of swarm intelligence are presented next, and then followed by our proposed protocol.

3. Swarm Principles

Swarms are useful in many optimization problems. A swarm of agents is used in a stochastic algorithm to obtain near optimum solutions to complex, non-linear optimization problems [31, 32]. In this work, we consider the use of a swarm of agents and its control aspects. Swarm control issues are important as the swarm behavior could be used to establish logical network topology. In the description of his *boids* model, Reynolds [33] presents the classic swarm control theory. There are three basic controlling behaviors that govern movements of agents within the swarm. Kadrovach and Lamont [30] have summarized these behaviors as shown in Table 1.

Table 1
Agent behaviors in a swarm

Behavior	Description
Separation	Avoid collisions with *nearby* agents.
Alignment	Attempt to match velocity with *nearby* agents.
Cohesion	Attempt to stay close to *nearby* agents.

When agents migrate in a swarm, every agent must ensure that they do not collide with one another. Also, in order to ensure the best survival of an agent, it should stay as close as possible to the others. This implies that an agent should match its velocity with neighboring agents to keep abreast. Any swarm behavior is solely based on locally observable phenomena, and is reflected in Table 1 by the adjective *nearby*. The integration of these behaviors results in a stable swarm formation, where every agent is at least some minimum distance from others. In social insects, many sophisticated community behaviors emerge from the interaction of individuals where each insect carries out simple tasks. Some known collective behaviors are foraging, nest construction, thermoregulation, brood sorting and cemetery formation [34]. These collective behaviors of social insects have inspired computer scientists to replicate them as they exhibits many attractive features, such as robustness and reliability through redundancy [4, 34].

In this paper, we propose a biologically-inspired clustering protocol that uses a collection of agents. We exploit on the first two swarm principles, namely separation and alignment, through pheromone control to achieve a stable and near uniform distribution of the CH nodes. Moreover, it is observed in [35] that the algorithm converges faster to an optimal or near optimal solution when pheromone is also reduced drastically from those elements that make up the worst solution in each iteration. Thus, subsequent ants are discouraged from returning to poorer solutions seen in the past. This constitutes the simplest way to implement *anti-pheromone* where the evaporation of pheromone is simulated by a significant reduction in existing pheromone levels. The integrated use of these separate components as part of the proposal is described in-depth further.

4. The T-ANT Protocol

Our main design goal is to consistently form an optimal number of clusters with a good CH distribution for a load-balanced and energy efficient network operation. Each regular node is mapped to exactly one cluster. Accordingly, we enumerate the requirements for the proposed design strategy:

1. *The protocol should be completely distributed. Each node only makes a local decision on its role.*
2. *The CH election should be completed in constant time.*
3. *The protocol should be efficient in terms of processing complexity and message exchange.*
4. *The elected CHs should be well distributed.*

4.1 Overview of the T-ANT protocol

As T-ANT is a dynamic clustering protocol, its operation is divided into rounds. Each round comprises a cluster setup phase and a steady state phase. A number of timers are utilized to control the operations for optimal performance. However, if time synchronization is absent, this protocol will still function albeit at sub-optimal performance. During the cluster setup phase, CHs are elected and clusters are formed around them. To avoid the maintenance of many state variables as in many current clustering proposals, we utilize social agents in a unique way to control CH election. (Just for simplicity of reference, we term them ants; these ants however do not behave as ants in the ant-colony optimization technique.) A node with an ant becomes a CH, whereas the others would choose to join the best cluster in range. As we reveal further, our approach represents an atypical use of the swarm intelligence approach. During the steady state phase, the cyclic process of data collection from members, data aggregation and transfer to the sink occurs at fixed intervals. Since ants play a critical role in this protocol, we need to determine an appropriate swarm size and the way they are introduced into the network. This issue is addressed through the ant release component used during network initialization.

The details of all aspects of the protocol are individually described further with appropriate analyses where necessary.

4.2 The ant release component

During network initialization, the sink releases a fixed number of ants (i.e. control messages) into the network. When the sink releases an ant, it chooses one of its neighbors at random. The ant makes a random walk into the network as deep as restricted by its time-to-live (TTL) value. Assuming the terrain is a square $M \times M$ and nodes with radio range r, TTL is initially fixed at $\lceil M / \sqrt{2}r \rceil$. This represents half of the diagonal terrain length (in hops) to facilitate the ants to wander away from each other. To avoid *attraction* between ants, they do not leave any pheromone trail at this stage. Before releasing the next ant, the sink waits for a time proportional to one-hop delay (τ) and a random offset to ensure its subsequent transmission does not self-interfere. When an ant arrives at a node, the next stop is again randomly chosen (excluding the sender) if TTL has not expired. If TTL expires, the ant remains at this node. However, if the final ant location overlaps with another ant, the latter ant must find another location. Algorithm 1 summarizes this initialization phase. Beyond this initialization phase, ants only migrate in single-hops at the start of each round (i.e. TTL = 1).

Algorithm 1. The Ant Release Protocol

```
Sink:
Repeat
  Choose a random neighbor (node i) to release an ant
  Send the ant to node i
  Wait for τ(1 + rand), where 0 < rand < 1
Until all ants are released

Other nodes:
If an ant wanders to node i
  Decrement its TTL
  If TTL > 0
    Choose a random neighbor, j
    Send the ant to node j
  ElseIf TTL == 0 and node i already has an ant
    Pick a random neighbor
    Send the ant to it
  Else
    Store the ant
  End
End
```

Deciding on the appropriate swarm size is crucial for T-ANT's optimal performance. The analysis goal here is to organize nodes into clusters for optimal data aggregation to minimize overall energy dissipation. Towards this end, an analysis to determine the appropriate swarm size is performed. The following assumptions are made for this analysis:

- The nodes are randomly scattered in a two-dimensional plane and follows a homogeneous spatial Poisson process with λ intensity [15].
- All nodes in the network are homogeneous with radio range r.
- The communication from each node follows isotropic propagation model.
- The energy needed for the transmission of one bit of data from node u to node v is the same as to transmit from v to u (i.e. a symmetric channel).
- The sink is located the center of the terrain (an alternate derivation is indicated for corner locations).

The overall idea of the derivation of this optimal system parameter value is to define a function for the energy used in the network to disseminate information to the sink.

As per the assumptions, the nodes are distributed according to a homogeneous spatial Poisson process. The number of nodes in a square area of side M is a Poisson random variable, N with mean λA where $A = M \times M$. Lets assume that for a particular realization of the process, there are n nodes in this area. If a node requires an ant to become a CH (the details of CH election is explained in the next subsection), and there are p% ants as to the number of nodes, there will be np nodes elected as CHs. Also, the CHs and regular nodes are distributed as per independent homogeneous spatial Poisson processes P1 and P0 of intensity $\lambda_1 = p\lambda$ and $\lambda_0 = (1\text{-}p)\lambda$, respectively.

Using the ideas in stochastic geometry, each node joins the cluster of the closest CH to form a Voronoi tessellation [36]. The plane divides into zones called Voronoi cells, with each cell corresponding to a P1 process point termed its nucleus. If N_v is the random variable representing the number of P0 process points in each Voronoi cell and L_v is the total length of all segments connecting the P0 process points to the nucleus in a Voronoi cell, then based on the results in [23]:

$$\mathrm{E}[\mathrm{N_v} \mid N = n] \approx \mathrm{E}[\mathrm{N_v}] = \frac{\lambda_0}{\lambda_1} = \frac{1-p}{p} \tag{1}$$

$$\mathrm{E}[\mathrm{L_v} \mid N = n] \approx \mathrm{E}[\mathrm{L_v}] = \frac{\lambda_0}{2\lambda_1^{\frac{3}{2}}} = \frac{1-p}{2p^{\frac{3}{2}}\sqrt{\lambda}} \tag{2}$$

Now, to derive the energy dissipation, the free space (d^2 path loss) channel model is used. [For other path loss exponents (i.e. 3-6), it follows a similar derivation, and has a constant effect on the optimal number of CHs.] To transmit an l-bit packet a distance r (i.e. its radio range), the radio expends:

$$\mathrm{E_{Tx}}(l,d) = l\mathrm{E_{elec}} + l\varepsilon_{fs}r^2 \tag{3}$$

where $\mathrm{E_{elec}}$ is the electronic energy that depends on factors like digital coding, modulation, filtering and spreading of the signal, and $\varepsilon_{fs}r^2$ is the amplifier energy that depends on the distance and the acceptable bit-error rate. As to receive a packet, the radio expends:

$$\mathrm{E_{Rx}}(l,d) = l\mathrm{E_{elec}} \tag{4}$$

The T-ANT protocol guarantees that the number of clusters per round is *always np*. The dissipated energy by the nodes can be analytically estimated using the computation and communication energy models. Each CH dissipates energy receiving signals from its members, aggregating the signals and transmitting the aggregate signal to the sink. Since a CH could be located at any (x,y) point on the terrain with uniform intensity, the probability density function of its location is constant ($1/M^2$). The transmission to the sink may also be multihop. We could estimate the average distance to the sink (d_{ToS}) by integrating the distance function over the area as follows:

$$\mathrm{E}[d_{\mathrm{ToS}}] = \int_y \int_x \sqrt{x^2 + y^2}\, \frac{1}{M^2}\, dx\, dy$$

$$= \sqrt{1/6}\, M \tag{5}$$

Limits of the definite integral are [-M/2,M/2] for both x and y. If we however assume the sink is located at a corner (say bottom left origin), range of this integral is changed to [0,M] giving $E[d_{ToS}] = \sqrt{2/3}M$. Accordingly, the average hop to the sink is $E[d_{ToS}]/r$. Now, let C_1 represent the energy spent by a CH node during a single round:

$$E[C_1 \mid N{=}n] = (E[N_v])lE_{elec} + (E[N_v]+1)lE_{DA} + \frac{\sqrt{1/6}M}{r}\left(2lE_{elec} + l\varepsilon_{fs}r^2\right) \tag{6}$$

We assume lossy data aggregation is performed at the CH with the energy for aggregation is E_{DA}.

As for each non-CH node, it only needs to transmit its data to the CH once during a data interval. Let C_2 represent the energy used by each non-CH node:

$$E[C_2 \mid N{=}n] = lE_{elec} + l\varepsilon_{fs}\left(\frac{E[L_v]}{E[N_v]}\right)^2 \tag{7}$$

This allows us to determine the energy spent in a cluster (C_3) during each round as:

$$E[C_3 \mid N{=}n] = E[C_1 \mid N{=}n] + E[C_2 \mid N{=}n] \times E[N_v] \tag{8}$$

Since there are np clusters in the network, we can now derive the total energy usage. Let C represent the total energy spent in the system, then:

$$E[C \mid N{=}n] = np\, E[C_3 \mid N{=}n]$$

$$= nl[2(1-p)E_{elec} + E_{DA} + \frac{\sqrt{1/6}Mp}{r}\left(2E_{elec} + \varepsilon_{fs}r^2\right) + \frac{\varepsilon_{fs}(1-p)}{4p\lambda}] \tag{9}$$

Removing the conditioning on N yields:

$$E[C] = E[E[C \mid N = n]] = E[N]\; l[2(1-p)E_{elec} + E_{DA} + \frac{\sqrt{1/6}Mp}{r}\left(2E_{elec} + \varepsilon_{fs}r^2\right) + \frac{\varepsilon_{fs}(1-p)}{4p\lambda}]$$

$$= \lambda M^2 l[2(1-p)E_{elec} + E_{DA} + \frac{\sqrt{1/6}Mp}{r}\left(2E_{elec} + \varepsilon_{fs}r^2\right) + \frac{\varepsilon_{fs}(1-p)}{4p\lambda}] \tag{10}$$

E[C] is minimized by a value of p that is a solution of:

$$\frac{\varepsilon_{fs}}{4p^2\lambda} + 2E_{elec} - \frac{\sqrt{1/6}M}{r}\left(2E_{elec} + \varepsilon_{fs}r^2\right) = 0 \tag{11}$$

Equation (11) has two roots where only one is positive. The second derivative of the above equation is also positive for this root, hence minimizing the total energy spent. This only positive root of (11) is given by:

$$p_{opt} = \frac{1}{2}\sqrt{\frac{\varepsilon_{fs}}{\lambda\left[\frac{\sqrt{1/6}M}{r}(2E_{elec} + \varepsilon_{fs}r^2) - 2E_{elec}\right]}} \tag{12}$$

The above expression gives the optimal percentage of ants required to achieve the optimal clustering performance in terms of energy dissipation.

4.3 The clustering algorithm

Any clustering algorithm comprises two components, namely CH election and cluster formation. Both components are invoked during the cluster setup phase of T-ANT, whereby the phase is activated through the CS_Timer. When this timer expires, a node checks to see whether it possesses an ant. If the node has an ant, it is guaranteed to become a CH. As there are only a fixed number of ants in the network, a fixed number of CHs are formed. Since there is no looping statement as a function of the number of nodes, it is trivially evident that the election terminates in constant time. This CH election approach has a very small constant time complexity as opposed to other proposals in the literature.

Observation 1. The CH election process terminates in $O(1)$ time (c.f. requirement 2).

The cluster formation is triggered by these CHs, whereby they advertise to the neighbors by broadcasting an advertisement (ADV) message with their node id. Upon receiving such message, a regular node records the CH id and the number of ADV messages received thus far. At this stage, the most critical difference between our biologically-inspired approach against other typical swarm intelligence applications occurs. When regular nodes receive an ADV message from a CH, we allow this message to affect the pheromone level at these nodes. Here, an ant does not need to be at a node to leave a pheromone trail, which is contrary to its typical application. As such, the influence of an ant is broader and quicker than their typical implementation. This atypical property of our approach has contributed to a significant reduction in the required number of ants for the system behavioral convergence, and thus overcomes a common problem highlighted in many biologically-inspired proposals.

The actual clustering process happens once another timer expires. A node decides to join a cluster when its JOIN_Timer expires. The node computes its pheromone level based on the number of CHs (n_c) in its neighborhood and its normalized residual energy. The pheromone function (p_i) is based on the forwarding probability formula used in the uniform ant routing algorithm [37], but expanded as:

$$p_i = \frac{p_{i-1} + \Delta p}{1 + \Delta p} \tag{13}$$

where Δp is given by:

$$\Delta p = k \times \frac{E_{resi}}{E_{max}} \times \frac{1}{n_c^2} \tag{14}$$

E_{resi} is a node's residual energy, E_{max} is the reference maximum battery energy and k is the learning rate of the algorithm (= 0.1). This pheromone function is adopted to achieve dual clustering objectives, namely load balancing as well as energy efficiency. The effect of quadratic drop in pheromone with the number of CHs in range not only promotes a good distribution of CHs, but also indirectly affects the load balancing objective. A regular node chooses the nearest cluster to join by sending a JOIN message with its id, the selected CH id and its pheromone level. When a CH receives JOIN messages, it finds the member with the highest pheromone level to attract its ant for the next round. It is obvious that the amount of state required for this clustering is small compared against the stochastic or weight-based clustering algorithms proposed in the literature.

Observation 2. T-ANT is completely distributed (c.f. requirement 1). A node locally decides to become a CH if an ant wanders to it or may join a cluster.

Lemma 1. T-ANT has a worst-case processing time complexity of $O(n)$ per node, where n is the number of nodes in the network (c.f. requirement 3).

Proof. For a homogeneous spatial Poisson process, the process intensity is uniform. Thus, the average node degree is $\pi r^2 n/M^2$, where r is the radio range and M is the side of the square region. Generally $r << M$. In the worst-case, a CH has to process JOIN messages from the potential members in the order of n to determine the neighbor with the highest pheromone level. It also has to receive and aggregate data messages linear in n during the steady state phase. Therefore, the worst-case processing time is $O(n)$. □

Lemma 2. T-ANT clustering has a worst-case message exchange complexity of $O(n)$ in the network (c.f. requirement 3).

Proof. During the cluster setup phase, a fixed number of CHs are elected as bounded by the number of ants. Each CH only broadcasts one ADV message. All *covered* regular nodes (i.e. nodes that receive at least one ADV) reply by sending a JOIN message giving $O(n)$ messages in the network. □

Before the next CS_Timer expires, the ants wander to the nodes with the highest pheromone level among their neighbors, and these nodes will become the next CHs. Before an ant leaves its current node, an amount of anti-pheromone is laid to mimic a rapid decay of pheromone level. It was shown in [38] that a rate of 0.1 is suitable to force an agent not to revisit a visited city in the *Traveling Salesman Problem*. This is also necessary in our context to ensure that the ants do not return to the same node too soon. This decay rate also indirectly promotes energy load balancing.

In the steady state phase, each regular node sends its sensory data message to its CH. To ensure that transmissions from the members face minimal collision, we adopt the radial coordination strategy proposed in [39] for *convergecast* transmissions. The details are given later as it is implemented within the MAC protocol. Due to the spatial proximity of the nodes in a cluster, the CH will perform a lossy aggregation to exploit of the possible high spatial correlations. The details on how to capture any correlation structure present in the network data is beyond the scope of this paper. For specific discussion on this issue, interested readers are referred to [40, 41] and references therein. Algorithm 2 presents overall T-ANT clustering with the setup and the steady state phases.

Algorithm 2. The T-ANT Clustering Algorithm

```
Cluster Setup:
When CS_Timer expires
  If node i has an ant
    CH status = true
    Create an ADV message (CH_id, TTL)
    Broadcast the ADV message to neighbors
  Else
    Node i sets its JOIN_Timer and waits for ADVs
  End
End
If node i receives an ADV message from CH k
  Increment number of ADVs received
  If node i is not a CH and this CH k is nearer
    Select CH k as the best CH
  End
End
/* Only for nodes with ADV */
```

```
When JOIN_Timer expires at node i
  Compute the pheromone level, p
  Create a JOIN message (CH_id, p)
  Send JOIN to the selected CH
End

Steady State:
When DATA_Timer expires at node i
  Capture sensory value, val
  If node i is a CH
    Wait for data messages from all members
    Aggregate data signals
    Create a data message (sink_id, val)
    Send to sink
  Else
    If node i belongs to a cluster
       /* for covered nodes */
      Create a data message (CH_id, val)
      Send to CH
    Else
      /* for uncovered nodes */
      Create a data message (sink_id, val)
      Forward to sink
    End
  End
End
/* only for CH nodes */
When ANT_Timer expires
  Pick the best neighbor, j
  Send ant to node j
End
```

Lemma 3. The probability of two CH nodes in each other's radio range is at least inverse of *n*, implying the CHs are mostly well distributed (c.f. requirement 4).

Proof. Without loss of generality, lets assume that there are two ants in a network of *n* nodes. In the worst-case, two CHs may be formed in almost complete overlapping ranges as shown in Fig. 1, especially during network initialization when the ants just follow a random walk. At this stage, the normalized residual energy is fairly uniform across the network (it does not need to be exactly same). Thus, the only factor that influences the pheromone level of the cluster members is the number of CHs (n_c) in range. The nodes in the shaded area will have a quadratic in n_c lower pheromone than the rest. This is due to the quadratic drop introduced into the pheromone function [see eqn. (14)]:

$$\Delta p \sim \frac{1}{n_c^2}$$

$$p_i \sim \frac{p_{i-1} + {}^{1}\!/\!{}_{n_c^2}}{1 + {}^{1}\!/\!{}_{n_c^2}} \sim \frac{1}{n_c^2}$$

At the next cluster setup phase, the ants would only wander to *any* of their neighbors outside the shaded area (e.g. ant *a* wanders to any node in A ∩ B'), as their pheromone level is higher. These ants would only return to any nodes in the shaded area only when *all* other nodes (in the order of *n*) are visited. This implies that in a cycle of n visits, only once an ant comes in range of another ant giving a probability $1/n$. By induction, a network with more ants will exhibit a similar behavior. Moreover, when more than two ants come into range, the repulsion effect is even greater. Thus, CHs are mostly well distributed. □

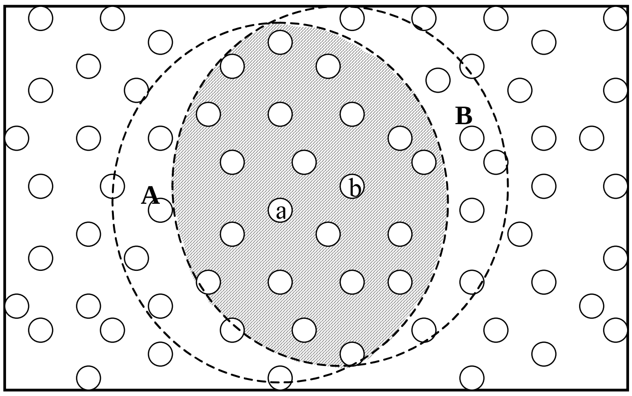


Fig. 1. An example to illustrate the worst-case scenario of closely overlapping CH nodes. Sets A and B represent nodes in range of CH *a* and *b*, respectively. The shaded area represents nodes with significantly lower pheromone level than the rest.

Theorem 1. The T-ANT protocol promotes load balancing.

Proof. In this dynamic protocol, the role of CH is continuously shared among the nodes. With Lemma 3, the elected CHs are mostly well distributed throughout the network lifetime. As each regular node chooses to associate to its nearest CH, this results in similar cluster sizes as well. Therefore, T-ANT achieves load balancing. □

It is possible that the wandering ants may die due to the environment uncertainty or any node failure. When there are lesser ants than the optimal number, the network operation would still proceed as before albeit suboptimally. Due to the robustness of the proposed protocol, the remaining CHs would strive to share the aggregation task. If there is any *uncovered* node, it would to resort to *direct* (multihop) transmissions towards the sink. The event of most ants dying simultaneously has a probability in the order of all nodes dying at about the same time, as the ants are mostly well distributed. Thus, such an event is only likely when the network is nearing the end of its lifetime. To limit the impact of isolated node failures, we could introduce a timer on the ants. When this timer expires, current ants are flushed out and the sink re-releases the same optimal number of ants to restart the above process. Further robustness and reliability study on this issue is left for future study.

4.4 The clustering properties

The use of a specific swarm size as derived earlier guarantees the network mostly operates with the optimal number of clusters compared to a stochastic approach that may have fewer or more than the optimal number at any time. The given pheromone expression guides the evolution of the swarm to achieve the *separation* behavior among ants in the swarm, which is one the swarm principles highlighted in Section 3. Separation may not be achieved immediately upon the random release of these ants by the sink. However, it is found empirically that separation is attained rather quickly within a small constant number of rounds. To quantify this phenomenon, the *CH election fitness* metric is introduced. It is defined as follows:

$$S = \frac{1}{n_c}\sum_{i=1}^{n_c} n_i \qquad (15)$$

where n_c is the number of CH nodes and n_i is the number of ADVs seen by CH_i. By normalizing against number of CHs, it captures the average number CHs in the range of another. This metric will have a small value for a network with well distributed CHs. For instance, a value of two represents on average two CHs in the range of another CH. Another desirable swarm behavior is *alignment*. In our context, the number of members served by a CH is used to represent alignment. When the swarm evolves to achieve separation, alignment is also achieved as a side-benefit. The *clustering fitness* metric A is defined to represent alignment as follows:

$$A = \sqrt{\frac{1}{n_c}\sum_{i=1}^{n_c} (m_i - \mu^*)^2} \qquad (16)$$

where m_i is the number of members of CH_i and μ^* is the ideal average number of nodes served by a CH in a perfect uniform CH distribution. μ^* is computed as n/n_c. Thus, this clustering fitness metric is the standard deviation of the clusters membership against the ideal average. A smaller metric value implies that the membership variance among the clusters is small thereby the clusters are well balanced.

5. The Simulation Framework

We evaluated the performance of the T-ANT clustering using a discrete-event simulator. To enable a comprehensive study, the effects of both routing and MAC protocols are also integrated. In the description of the simulator, we assume that each sensor node is aware of:

1. its neighbors (even if it changes) due to the occasional *beaconing* by the sink and the cluster setup phase; and
2. the network is synchronized by means of any time synchronization protocols [42].

The simulator is developed in C++ and adopts the object-oriented approach to allow natural mapping to a real sensor network. The radio model is currently assumed to follow an isotropic propagation. As for the MAC choice, we adopted the CSMA protocol due to its simplicity and its promise of scalability [39, 43]. However, a straightforward application of this protocol in a convergecast scenario is a recipe for failure. In the periodic monitoring type application, when the sensor DATA_Timer expires, all nodes capture their sensory value and convert to digital via an analog-to-digital converter (ADC) linked to the sensing hardware. Assuming a time-synchronized clustered network, all nodes simultaneously generate the sensory message for transmission towards their CHs. If no precautions were taken, their transmissions would interfere resulting in many collisions and retransmissions. This many-to-one transmission is known as convergecast. In order to reduce collisions, Huang and Zhang [39] proposed that a node should delay its transmission relative to its distance to the hop destination (h) and the node density. As there is likely to be many nodes at each hop, an additional random offset is also included to further reduce the collision probability. We chose to adopt a similar temporal coordination with respect to a node's CH. However, it was modified to be less conservative to reflect the smaller scope of a cluster rather than an entire network as in [39]. Accordingly, the random wait time function (T) is given as:

$\mathrm{T}(h) = (1 + rand).\tau h$ (17)

where *rand* is a uniformly distributed random number from 0 to 1 and τ is the average one-hop delay.

At the network layer, we adopted a simple routing mechanism in Greedy Routing Scheme (GRS) [44] to control the network's forwarding behavior. The forwarding objective is to minimize the number of hops between the sink and the other nodes. To establish this minimum hop routing tree, the sink occasionally broadcasts a *beacon* message with a hop count, which is initialized to zero. To overcome the problem of asymmetrical links that may be prevalent in WSNs [45], the sink could broadcast at a power level lower than the maximum level of a regular node, a move similar to He's proposal [46]. This is possible for a sensor node with tunable transmit power as in Mica2 motes. Upon receiving the beacon, each node records the sender id, increments the hop count by one, and then rebroadcasts at a power level below its maximum level. A node only rebroadcasts if the new hop count is smaller than its stored value. Since we are focusing on a stationary scenario, the sink node only needs to perform occasional beaconing to avoid significant overhead. This forwarding rule establishes a minimum hop tree rooted at the sink. Finally, the T-ANT protocol is implemented between the application and the network layer, and thus, the overall system framework is as shown in Fig. 2.

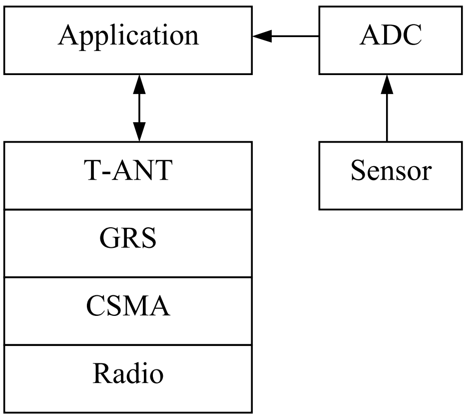


Fig. 2. Simulation framework involving the core components of a communication stack.

Based on the given simulation framework, we investigate T-ANT's performance against LEACH, HEED and a flat strategy (i.e. the application sits directly on GRS). Since LEACH can't be applied directly to a multihop network, we modified this algorithm to use the GRS routing protocol to forward messages whenever the destination is not within the radio range. As for HEED, even though this algorithm is able to achieve a good CH distribution in general compared to others in the literature, the use of tentative CHs that do not finally become CHs leaves some regular nodes *uncovered* (i.e. no CH in range to join). According to the proposed HEED implementation [3], such uncovered nodes are forced to become CHs. In our view, this move goes against its main design goal of achieving a good CH distribution. Such forced CHs may be in range of other CHs, and they are mostly single-node clusters. The issue of having many single-node clusters was addressed by increasing the radio range of the nodes in [3]. We, however, feel that instead of forcing these nodes to become CHs, they should just be left uncovered. Only the nodes that did not receive any messages at all should be made CHs. Such a move has a notional structure change, but it is operationally the same. Thus, for our comparative study here, we adopt this suggested implementation.

The results from the comparison and other evaluations are presented further.

6. Results and Discussions

For these simulation experiments, we assumed that there are 100 sensor nodes distributed randomly in a square $M \times M$ region with M = 100 m. The transceiver energy parameters are set as: E_{elec} = 50 nJ/bit and ε_{fs} = 10 pJ/bit/m^2. The energy for data aggregation is set to E_{DA} = 5 nJ/bit per signal [13]. The control and data message sizes are fixed at 30 bytes, and sensory data is generated at 2-second interval. Each CH node retains its CH status for 20 seconds. The anti-pheromone rate is fixed at 0.1. Unless otherwise stated, all the following investigations adopt these values as their system parameters as summarized in Table 2. For all simulation results in this paper, each experiment is repeated 10 times and a 95% confidence interval is obtained.

Table 2
The System Parameters

Parameter	**Value**
Field dimension, M	100 m
Message size	30 bytes
Sensor DATA_Timer	2 sec
CS_Timer	20 sec
Anti-pheromone rate, β	0.1
Electronics energy, E_{elec}	50 nJ/bit
Amplifier energy, ε_{fs}	10 pJ/bit/m^2
Data aggregation energy, E_{DA}	5 nJ/bit/signal
Node radio range	30 m

For the purpose of benchmarking our protocol against others in the literature, we chose to compare against two well known proposals, LEACH and a more recent one, HEED, as indicated in the previous section. These two protocols belong to two different approaches of clustering; the former is a stochastic clustering scheme and makes a clever use of dynamic thresholding, whereas the latter adopts a widely used approach in the MANET environment, namely weight-based clustering for ensuring good CH distribution. The key design goals of these protocols are the same as our proposal. Both protocols are good baseline for comparison due to the following features:

- Clustering is distributed and entirely based on local information.
- Clusters are disjoint.
- They are designed to achieve energy efficiency through load balancing.
- Each employs a different mode of CH election.

For the HEED protocol implementation, we used the values suggested in [3] for its parameters, and chose to use node degree as the secondary parameter for clustering, as it results in the best load-balanced performance. In this section, a comprehensive evaluation of T-ANT is provided. The followings are the different aspects of study performed:

- The number of clusters required for optimal T-ANT performance.
- The clustering properties of the clustered formations.
- The distribution of residual energy of nodes.
- Topology formation as the system evolves towards convergence.
- Network lifetime.

These are the performance metrics utilized in our investigations:

- *Clustering fitness*: This is based on the fitness metric given as eqn. (16). It represents the fitness of the cluster formation in terms of membership.
- *CH election fitness*: This is based on the fitness metric given in eqn. (15). It represents the fitness of all the elected CH nodes in terms of their distribution.
- *Energy per round*: This metric represents the energy dissipated by all nodes in a round of data collection.
- *Network lifetime*: This metric represents the time period from the instant the network is deployed to the moment when the first node runs out of energy.

6.1 T-ANT characteristics

Since T-ANT's performance depends on the number of ants used to form the clusters, the analysis results are used to determine this optimal number. These results are verified using simulation. Fig. 3 depicts the comparison between those results. Even though the absolute energy values are different, it is obvious that both curves suggest a very similar pattern. The discrepancy is mainly due to the analysis considering the energy usage of T-ANT alone, whereas the simulator considers the overhead of the entire stack. Both results concur that the optimal number of ants for this scenario is nine. This optimal number of ants, which determines the number of clusters, is attainable using eqn. (12). When there are fewer or more clusters in the network, the energy dissipation worsens. When there are fewer clusters, there will be many cases of uncovered nodes. These nodes will resort to sending their messages, possibly multihop, to the sink. However, when there is more than the optimal number, the overhead of managing these clusters will outweighs their benefit. Thus, it is necessary to identify the optimal number of ants to achieve T-ANT's best performance based on the deployment parameters.

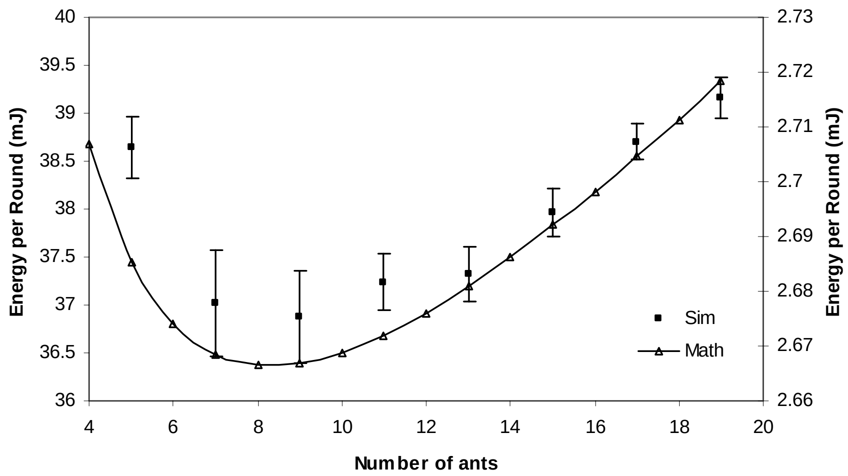


Fig. 3. Average energy dissipated against the number of ants for T-ANT. The simulation results (i.e. dots) are plotted against the left y-axis, and the analysis results (i.e. line) are against right y-axis.

In order to observe the generality of the above finding, we used eqn. (12) to determine the optimal number of clusters for different terrain areas as well as number of nodes. In Fig. 4, a wireframe 3D plot is given to depict how the optimal number varies with the two factors. For a fixed area, the optimal number of clusters varies in

$O\left(\sqrt{n}\right)$, where n is the number of nodes. As for a fixed number of nodes, the optimal number of clusters is linear of terrain area.

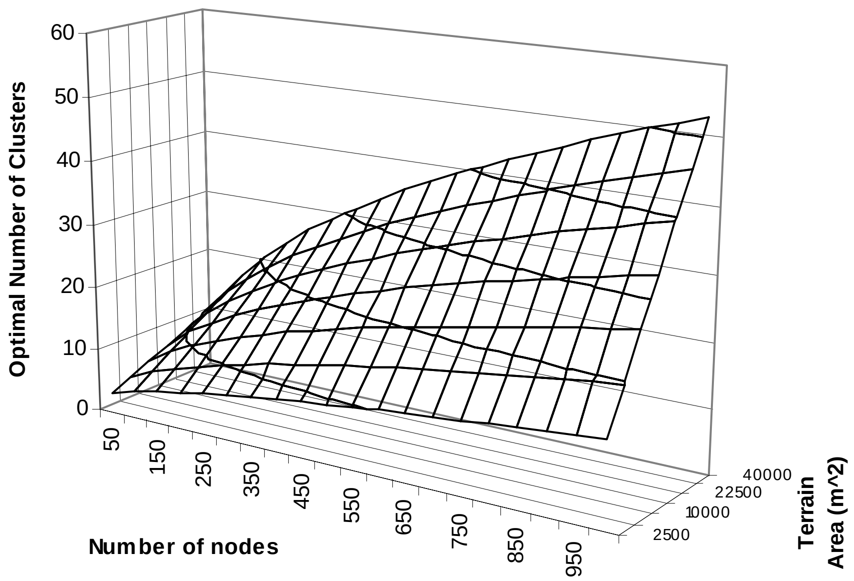


Fig. 4. The optimal number of clusters [using eqn. (12)] against the number of nodes and the terrain area. The number of nodes is in range [50,1000], whereas area size is in [2500,40000].

6.2 Clustering properties

In this study, we characterize the clustering properties of T-ANT against LEACH and HEED in terms of clustering and CH election fitness. Fig. 5 depicts clustering fitness of these protocols at different simulation time. This fitness metric represents the alignment property, which indicates how well aligned the cluster sizes are. Both T-ANT and HEED mostly exhibit small standard deviation values. This implies each cluster has similar number of members. This behavior is only achievable if the CHs are always well distributed. As claimed in [3], we verified that HEED is able to achieve this property with our proposed modification. As for LEACH, the fitness value varies quite unpredictably. This is mainly due to its stochastic election nature, where the number of clusters formed at each round changes quite significantly. Moreover, it is also possible that the elected CHs may even be clumped. When the CHs are clumped, the disparity among clusters is large in terms of their membership. At other times, the number of clusters may be substantially lesser than necessary to achieve complete network coverage. Many nodes will not be covered and the few clusters may have too many members far from the ideal mean. The two outlying LEACH points on Fig. 5 represent such a scenario.

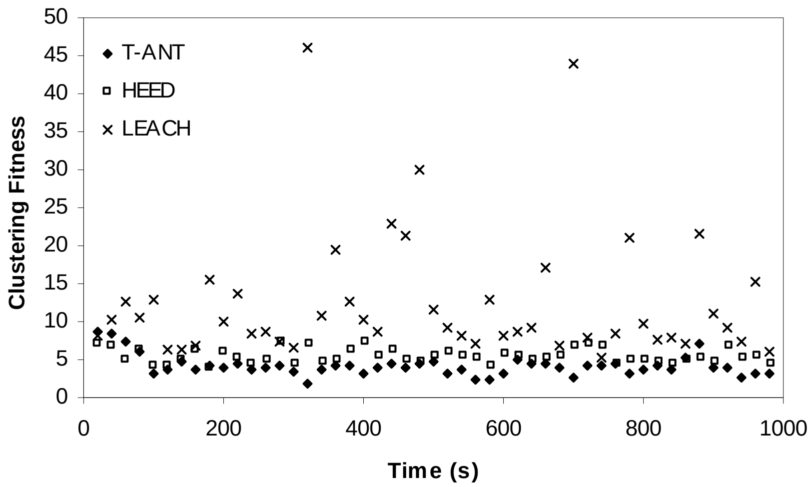


Fig. 5. The clustering fitness at different simulation time for T-ANT, LEACH and HEED.

In Fig. 6, the CH election fitness is depicted for these protocols. This fitness metric captures how well the elected CHs are distributed. It represents the average number of CHs in the range of another. A rather consistent behavior as above is obtained. Both T-ANT and HEED have smaller values with HEED showing the least CH overlaps with the proposed modification. Due to its biological inspiration, T-ANT has a higher value initially before the system achieves convergence. As the swarm evolves, the ants are able to move to better locations, and within five rounds, the swarm is able to achieve separation. Even after a good CH distribution is achieved, the fitness value keeps changing as the ants are forced to wander to achieve load balancing. It is also found that HEED mostly forms lesser number of CHs than T-ANT. The number of clusters formed using T-ANT always follows the suggested optimal number as per our analysis. Thus, in terms of energy efficiency, HEED will not be able to perform as well as T-ANT due to its clustering process. As for LEACH, its probabilistic CH election forces the value to vary significantly. At some rounds, there are no CHs in range of another but at other rounds, there are more than a few CHs in range.

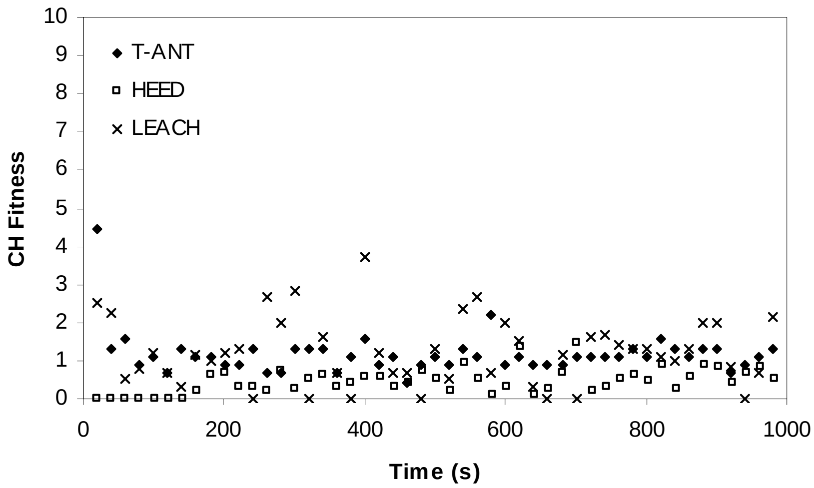
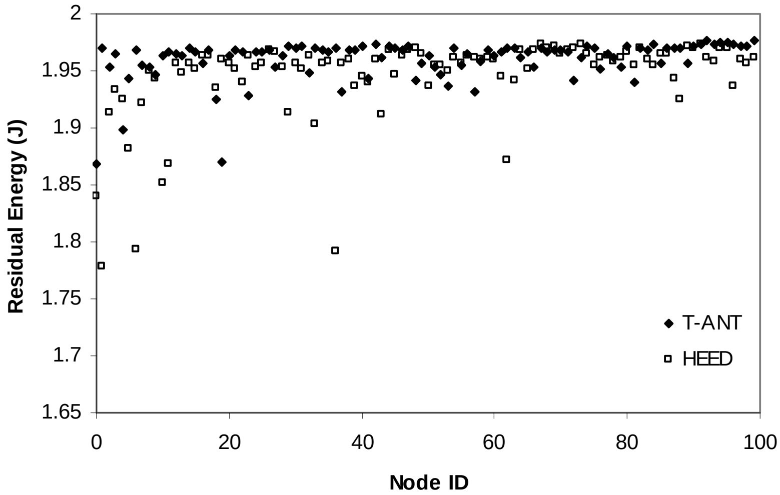


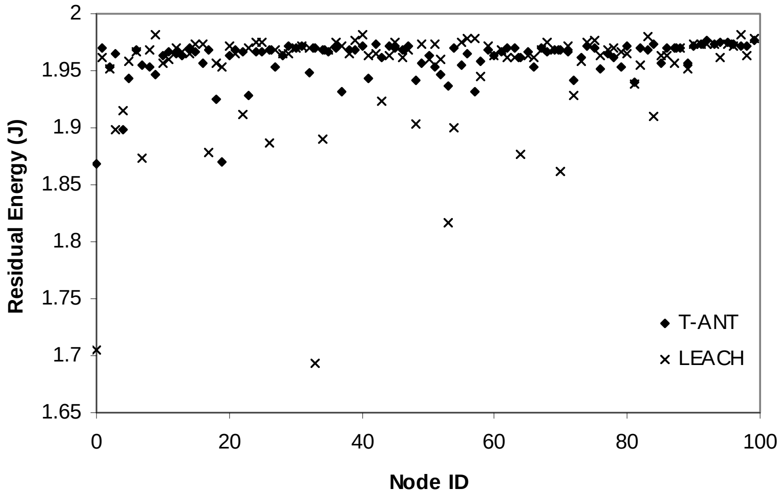
Fig. 6. The CH election fitness at different simulation time for T-ANT, LEACH and HEED.

6.3 Load distribution

In order to observe how well T-ANT promotes load balancing among the nodes, we ran a simulation on periodic data collection at 2-sec interval for 2000 sec. At the outset, each node had 2-J battery energy. Figs. 7(a) and (b) shows residual energy across the nodes at the end of simulation comparing three protocols. The plots are separated to avoid clutter. It is obvious that T-ANT achieved the best performance by maintaining a near uniform battery discharge among the nodes. HEED did not perform as well mainly due to its use of tentative CHs that do not ultimately become CHs. This causes the affected nodes to become uncovered, which in turn forces them to send their messages directly, possibly multihop, to the sink. As expected, LEACH also did not perform as well. Since LEACH does not guarantee an optimal clustering throughout its operation, the number of clusters formed and their sizes vary greatly. When a suboptimal clustered topology is formed and operated for certain interval, this causes some nodes to become more energy drained than the others.



(a)

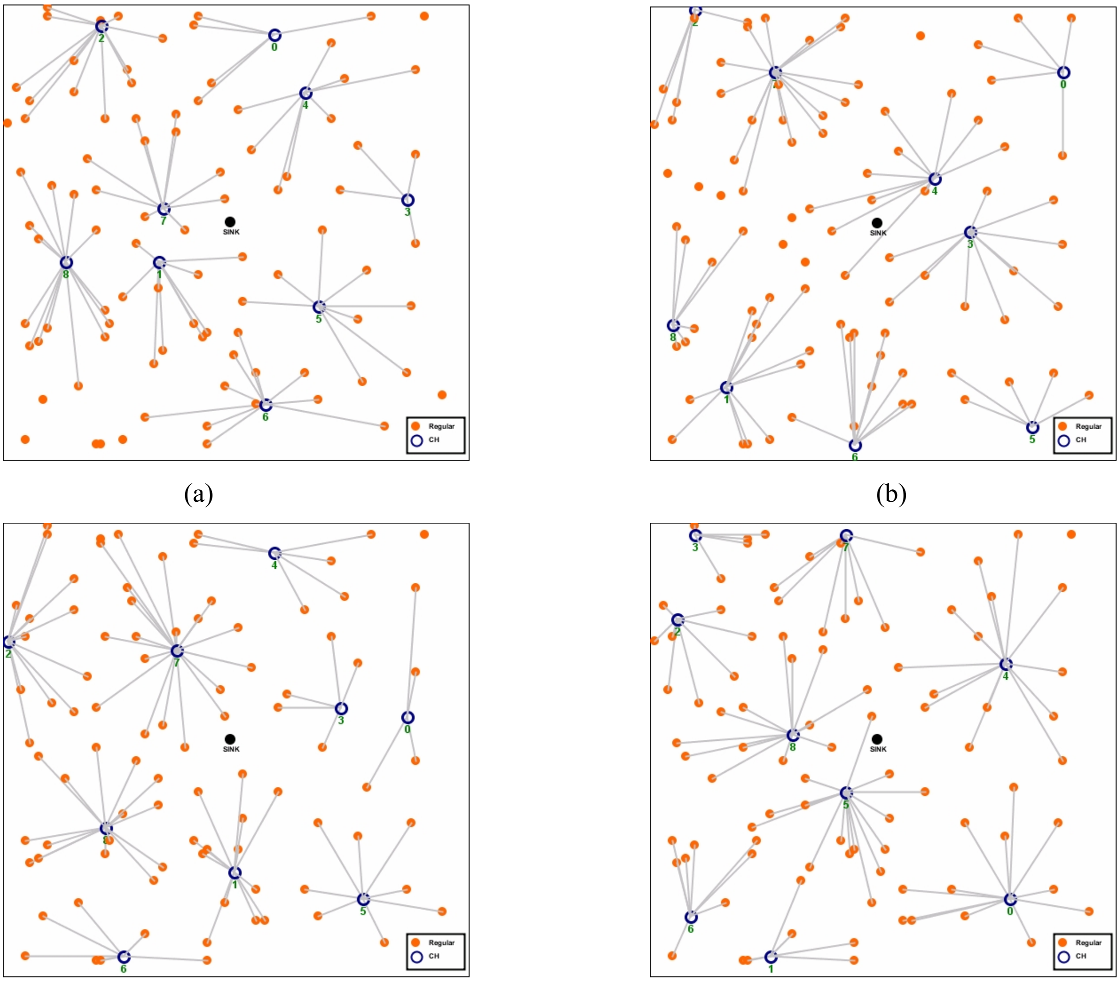


(b)

Fig. 7. Residual energy distribution of nodes after 2000-sec simulation of T-ANT, HEED and LEACH. The nodes are initially equipped with 2-J battery energy. (a) T-ANT vs. HEED; (b) T-ANT vs. LEACH.

6.4 Dynamics of clustered topology

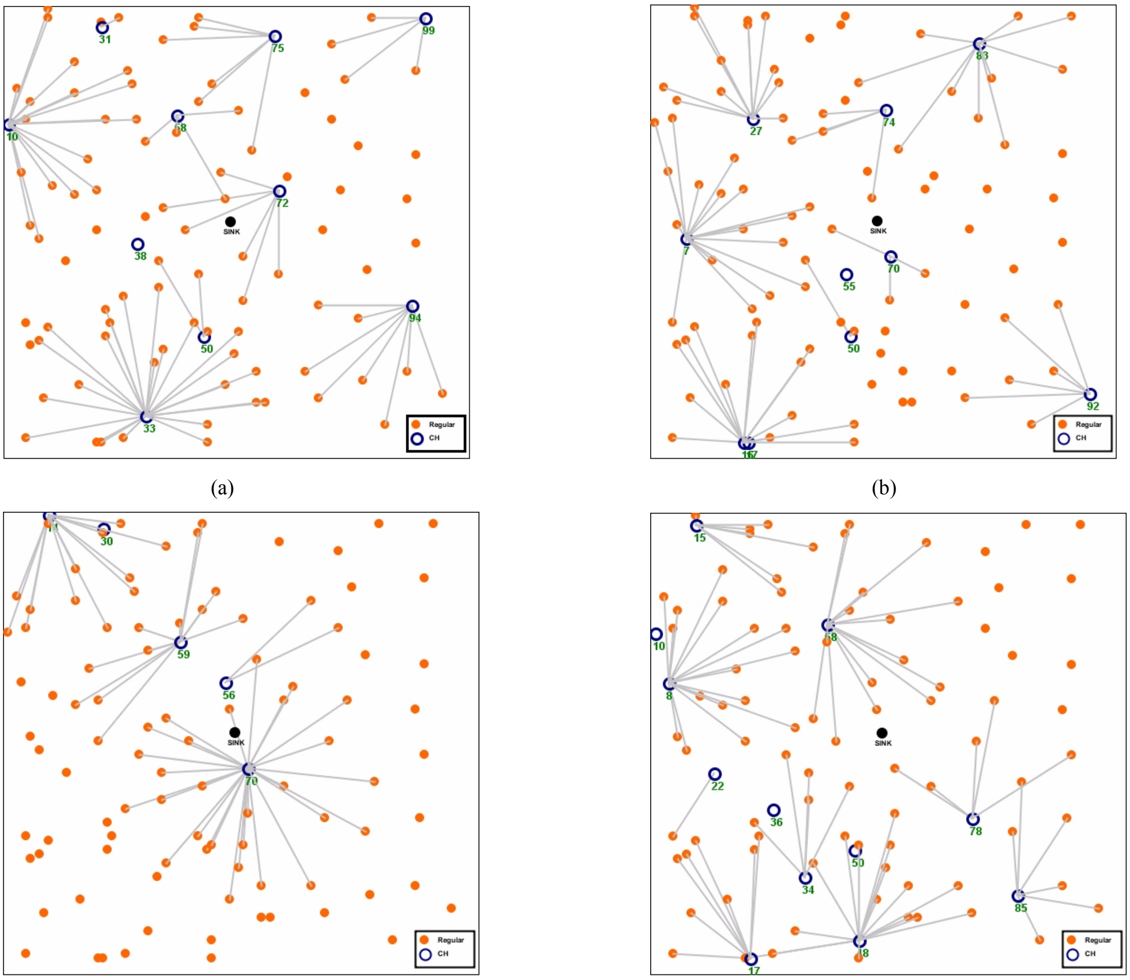
The following figures visually demonstrate how T-ANT promotes good distribution of CHs in the network compared to HEED and LEACH. Figs. 8 and 9 depict the clustered topology formation at different rounds. For T-ANT, we chose to use the optimal number of ants based on the analysis. In the following figures, a ring represents a CH, a filled circle represents a regular node, and a line segment links a member to its CH. Any unlinked regular nodes are uncovered. For T-ANT, it is visually evident that when the sink randomly releases the ants, they could still be forced to neighboring nodes, which results in some regular nodes left uncovered as shown in Fig. 8(a). However, as the swarm evolves, the ants are able to achieve some separation by round three [Fig. 8(b)]. When the system converges by round five, the network achieves good CH distribution.



(a) (b) (c) (d)

Fig. 8. The dynamics of T-ANT clustered topology at different rounds as the swarm evolves. During earlier rounds, quite a few nodes are uncovered as the swarm behavior has not converged. However, the coverage is almost total when convergence is achieved. Topology formed at round: (a) 1 (b) 3 (c) 10 and (d) 20.

HEED is also able to achieve a fair CH distribution. However, it is not able to achieve complete network coverage resulting in many regular nodes left un-clustered. There are two issues that might have contributed to this shortcoming. Firstly, the issue of tentative CHs that do not finally become CHs, as highlighted earlier. The other issue is with the protocol's *Finalize* phase, its last phase of cluster setup. During this phase, any node that did not receive a CH message is forced to become a CH. This causes some neighboring nodes to become CHs, as depicted in Fig. 9(b). LEACH exhibits significant variations to its clustered topology, as expected. For certain rounds, there are very few clusters formed with many nodes left uncovered, as shown in Fig. 9(c). At other rounds, there are many more clusters than necessary, which force many of them to be in range of quite a few CHs [Fig. 9(d)]. Such a formation results in significant disparity in their cluster membership, leading to imbalance in energy dissipation when the network has to operate under this sub-optimal structure for a certain interval.



(a) (b) (c) (d)

Fig. 9. The dynamics of HEED and LEACH clustered topologies at different rounds. The coverage of both protocols is not total. HEED at round (a) 10 (b) 20; LEACH at round (c) 10 and (d) 20.

6.5 Network lifetime

In Fig. 10, the improvement gained through T-ANT is further exemplified by the network lifetime graph. For this investigation, we set initial battery energy at only 0.1J to let the nodes to die sooner. This however does not change the pattern of behavior of these protocols. It is evident that T-ANT exhibits the longest lifetime with all nodes remaining fully functional. It is found that T-ANT achieves more than twice the lifetime of HEED and LEACH. It also achieves as much as five times the lifetime of the flat routing approach. There is a distinctive stair-case effect across all the curves. This is caused by the chosen routing protocol. Whenever certain critical nodes on the routing tree die, a part of the network is partitioned resulting in a number of nodes simultaneously losing connectivity to the sink. Thus, there are sudden plunges in the number of active nodes throughout the curves. T-ANT avoids the use of such critical nodes as long as possible with its ants. This obviously indicates that T-ANT promotes good load balancing across the entire network to sustain the network for its longest possible use. Fig. 10 also indicates the utility of a clustering algorithm against a flat routing approach. All the clustering protocols with data aggregation are able to sustain network lifetime at least twice the lifetime of the minimum hop routing protocol.

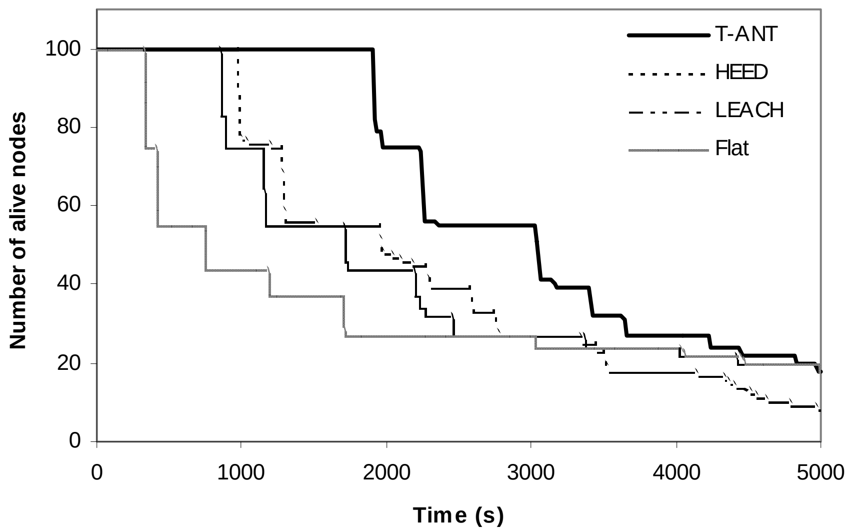


Fig. 10. Network lifetime against simulation time of T-ANT, HEED and LEACH.

7. Conclusions

In this work, we proposed the first novel application of the swarm intelligence technique to the resource-starved wireless sensor nodes domain. Due to the unique challenges of this domain, we devised an approach that allows the algorithm to converge very quickly with only limited overhead. Unlike the traditional applications of such biologically-inspired technique, we allow an agent to influence all nodes within the radio range of the host node, and not just this node alone.

The T-ANT clustering protocol utilizes these social agents to guide the election of clusterheads in a totally distributed manner. An analysis was performed to determine the number of clusterheads necessary to achieve the optimal performance in terms of energy dissipation. Accordingly, we employed this fixed number of agents to elect clusterheads. This unique approach allowed T-ANT to form and maintain the optimal number of clusters throughout the network operation. Due to the robustness of any biologically-inspired algorithm, this protocol could handle unforeseen circumstances in the environment and node failures. We have also introduced two fitness metrics that are useful for comparison of clustering properties between algorithms. It is demonstrated here that T-ANT is able to achieve two desirables swarm behaviors, namely separation and alignment, in a fully distributed way. Due to the separation behavior, the elected clusterheads are mostly well distributed across the network. Moreover, as a side-benefit of this behavior and the alignment behavior, T-ANT also achieved an even distribution of members among the clusters. It is also found that T-ANT maintain substantially lesser state overhead in memory compared to LEACH or HEED.

Such a biologically-inspired approach may also be useful in applications that require an in-network actuation, to assist in the sensor-actuator coordination. The feasibility of this use is left for our future work.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions. This work was supported in part by grants from ARC Discovery Project DP0664782 and USyd R&D L2844 U3230.

References

[1] J. Hill, M. Horton, R. Kling, L. Krishnamurthy, The platforms enabling wireless sensor networks, Communications of the ACM 47 6 (2004) 41 - 46.
[2] K. Akkaya, M. Younis, A survey on routing protocols for wireless sensor networks, Elsevier Ad Hoc Networks 3 3 (2005) 325-349.
[3] O. Younis, S. Fahmy, Heed: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks, IEEE Transactions on Mobile Computing 3 4 (2004) 366 - 379.
[4] Y. Ohtaki, N. Wakamiya, M. Murata, M. Imase, Scalable ant-based routing algorithm for ad-hoc networks, in: 3rd IASTED International Conference on Communications, Internet, and Information Technology, 2004, pp.
[5] S. Selvakennedy, S. Sinnappan, Y. Shang, Data dissemination based on ant swarms for wireless sensor networks, in: IEEE Consumer Communications and Networking Conference, 2006, pp. 132-136.
[6] M. Gerla, J. T.-C. Tsai, Multicluster, mobile, multimedia radio network, Wireless Networks 1 3 (1995) 255-265.
[7] F. Kuhn, T. Moscibroda, R. Wattenhofer, Initializing newly deployed ad hoc and sensor networks, in: 10th annual international conference on Mobile computing and networking, 2004, pp. 260-274.
[8] D. Baker, A. Ephremides, The architectural organization of a mobile radio network via a distributed algorithm, IEEE Transactions on Communications 29 11 (1981) 1694-1701.
[9] G. Gupta, M. Younis, Load-balanced clustering of wireless sensor networks, in: IEEE International Conference on Communications, May 2003, pp. 1848-1852.
[10] A. D. Amis, R. Prakash, Load-balancing clusters in wireless ad hoc networks, in: 3rd IEEE Symposium on Application-Specific Systems and Software Engineering Technology, Mar 2000, pp. 25-32.
[11] A. D. Amis, R. Prakash, T. H. P. Vuong, D. T. Huynh, Max-min d-cluster formation in wireless ad hoc networks, in: 19th Joint Conference of the IEEE Computer and Communications, Mar 2000, pp. 32-41.
[12] C.-F. Chiasserini, I. Chlamtac, P. Monti, A. Nucci, An energy-efficient method for nodes assignment in cluster-based ad hoc networks, Wireless Networks 10 3 (2004) 223-231.
[13] W. B. Heinzelman, A. P. Chandrakasan, H. Balakrishnan, An application-specific protocol architecture for wireless microsensor networks, IEEE Transactions on Wireless Communications 1 4 (2002) 660 - 670.
[14] S. Selvakennedy, S. Sinnappan, A configurable time-controlled clustering algorithm for wireless sensor networks, in: 1st International Workshop on Heterogenous Wireles Sensor Networks, IEEE Computer Society, 2005, pp. 368-372.
[15] S. Bandyopadhyay, E. J. Coyle, An energy efficient hierarchical clustering algorithm for wireless sensor networks, in: 22nd IEEE Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), 2003, pp. 1713 - 1723.
[16] A. Ephremides, J. Wieselthier, D. Baker, A design concept for reliable mobile radio networks with frequency hopping signaling, Proceedings of the IEEE 75 1 (Jan 1987) 56-73.
[17] C. R. Lin, M. Gerla, Adaptive clustering for mobile wireless networks, IEEE Journal on Selected Areas in Communications 15 7 (1997) 1265-1275.
[18] S. Basagni, Distributed clustering for ad hoc networks, in: 4th International Symposium on Parallel Architectures, Algorithms, and Networks, Jun 1999, pp. 310-315.
[19] M. Chatterjee, S. K. Sas, D. Turgut, An on-demand weighted clustering algorithm (wca) for ad hoc networks, in: IEEE Global Telecommunications Conference, Dec 2000, pp. 1697-1701.
[20] S. Ghiasi, A. Srivastava, X. Yang, M. Sarrafzadeh, Optimal energy aware clustering in sensor networks, Sensors 2 (2002) 258-269.
[21] J. Pan, Y. T. Hou, L. Cai et al., Topology control for wireless sensor networks, in: 9th international conference on mobile computing and networking, Sept 2003, pp. 286-299.
[22] T. Shu, M. Krunz, S. Vrudhula, Power balanced coverage-time optimization for clustered wireless sensor networks, in: 6th ACM international symposium on Mobile ad hoc networking and computing, May 2005, pp. 111-120.
[23] M. J. Handy, M. Haase, D. Timmermann, Low energy adaptive clustering hierarchy with deterministic cluster-head selection, in: 4th International Workshop on Mobile and Wireless Communications Network, 2002, pp. 368 - 372.
[24] M. Genes, U. Sorges, I. Bouazizi, Ara - the ant-colony based routing algorithm for manets, in: ICPP Workshop on Ad Hoc Networks, Aug 2002, pp. 79 - 85.

[25] G. D. Caro, M. Dorigo, Antnet: Distributed stigmergetic control for communication networks, Journal of Artificial Intelligence 9 (1998) 317-365.
[26] M. Roth, S. Wicker, Termite: Emergent ad-hoc networking, in: Second Mediterranean Workshop on Ad-Hoc Networks, 2003, pp.
[27] H. F. Wedde, M. Farooq, T. Pannenbaecker et al., Beeadhoc: An energy efficient routing algorithm for mobile ad hoc networks inspired by bee behavior, in: 2005 conference on Genetic and evolutionary computation, June 2005, pp. 153-160.
[28] I. Kassabalidis, M. A. El-Sharkawi, R. J. Marks et al., Adaptive-sdr: Adaptive swarm-based distributed routing, in: International Joint Conference on Neural Networks, May 2002, pp. 351-354.
[29] M. Heissenbuttel, T. Braun, Ants-based routing in large scale mobile ad-hoc networks, in: Kommunikation in verteilten Systemen, Feb 2003, pp. 91-99.
[30] B. A. Kadrovach, G. B. Lamont, A particle swarm model for swarm-based networked sensor systems, in: ACM symposium on Applied computing, Mar 2002, pp. 918-924.
[31] M. J. Mataric, Issues and approaches in the design of collective autonomous agents, Robotics and Autonomous Systems 16 2-4 (1995) 321-331.
[32] M. Dorigo, G. D. Caro, L. M. Gambardella, Ant algorithms for discrete optimization, Artificial Life 5 2 (1999) 137-172.
[33] C. W. Reynolds, Flocks, herds and schools: A distributed behavioral model, ACM SIGGRAPH Computer Graphics 21 4 (1987) 25-34.
[34] V. Hartmann, Evolving agent swarms for clustering and sorting, in: conference on Genetic and evolutionary computation, Jun 2005, pp. 217-224.
[35] R. Schoonderwoerd, O. E. Holland, J. L. Bruten, L. J. M. Rothkrantz, Ant-based load balancing in telecommunications networks, Adaptive Behavior 2 (1996) 169–207.
[36] S. G. Foss, S. A. Zuyez, On a voronoi aggregative process related to a bivariate poisson process, Advances in Applied Probability 28 4 (1996) 965-981.
[37] D. Subramanian, P. Druschel, J. Chen, Ants and reinforcement learning: A case study in routing in dynamic networks, in: International Joint Conference on Arti cial Intelligence, Morgan Kaufmann, 1997, pp. 832-838.
[38] J. Montgomery, M. Randall, Anti-pheromone as a tool for better exploration of search space, in: Third International Workshop on Ant Algorithms, Springer-Verlag, Sept 2002, pp. 100-110.
[39] Q. Huang, Y. Zhang, Radial coordination for convergecast in wireless sensor networks, in: 29th IEEE International Conference on Local Computer Networks, Nov 2004, pp. 542-549.
[40] H. Gupta, V. Navda, S. R. Das, V. Chowdhary, Efficient gathering of correlated data in sensor networks, in: MobiHoc '05, ACM Press, 2005, pp. 402-413.
[41] J. Chou, D. Petrovic, K. Ramachandran, A distributed and adaptive signal processing approach to reducing energy consumption in sensor networks, in: INFOCOM, 2003, pp. 1054-1062.
[42] B. Sundararaman, U. Buy, A. D. Kshemkalyani, Clock synchronization for wireless sensor networks: A survey, Elsevier Ad Hoc Networks 3 3 (2005) 281-323.
[43] A. Woo, D. Culler, A transmission control scheme for media access in sensor networks, in: 7th annual international conference on Mobile computing and networking, Jul 2001, pp. 221-235.
[44] G. Finn, "Routing and addressing problems in large metropolitan-scale internetworks," ISI, Mar 1987.
[45] D. Ganesan, B. Krishnamachari, A. Woo et al., "Complex behavior at scale: An experimental study of low-power wireless sensor networks," UCLA, LA, 2002, pp. 1-11.
[46] T. He, S. Krishnamurthy, J. A. Stankovic et al., Wide-area monitoring of mobile objects: Energy-efficient surveillance system using wireless sensor networks, in: 2nd ACM International conference on Mobile systems, applications, and services, ACM, 2004, pp. 270 - 283.