Analytical modeling of swarm intelligence in wireless sensor networks through Markovian Agents

Dario Bruneo
Dipartimento di Matematica
Università di Messina
Messina, Italy
dbruneo@unime.it

Marco Scarpa
Dipartimento di Matematica
Università di Messina
Messina, Italy
mscarpa@unime.it

Andrea Bobbio
Dipartimento di Informatica
Università del
Piemonte Orientale
Alessandria, Italy
bobbio@mfn.unipmn.it

Davide Cerotti
Dipartimento di Informatica
Università di Torino
Torino, Italy
cerotti@di.unito.it

Marco Gribaudo
Dipartimento di Informatica
Università di Torino
Torino, Italy
marcog@di.unito.it

ABSTRACT

Wireless Sensor Networks (WSN) consist of a large number of tiny sensor nodes that are usually randomly distributed over a geographical region. In order to reduce power consumption, battery operated sensors undergo cycles of *sleeping - active* periods; furthermore, sensors may be located in hostile environments increasing their attitude to failure. As a result, the topology of the WSN may be varying in time in an unpredictable manner. For this reason multi-hop routing algorithms to carry messages from a sensor node to a sink should be rapidly adaptable to the changing topology. Swarm intelligence has been proposed for this purpose, since it allows to emerge a single global behavior from the interaction of many simple local agents. Swarm intelligent routing has been traditionally studied by resorting to simulation. The present paper is aimed to show that the recently proposed modeling technique, known as *Markovian Agents*, is suited to implement swarm intelligent algorithms for large networks of interacting sensors. Various experimental results and quantitative performance indices are evaluated to support the previous claim.

Keywords

Wireless Sensor Networks, Markovian Agents, Swarm intelligence, Gradient-based routing, Performance evaluation.

1. INTRODUCTION

Wireless Sensor Networks (WSN) are application-specific networks composed by a multitude of tiny sensor nodes with limited computation, communication, and power capabilities. Sensor nodes collect measures of physical parameters and transmit them to a sink node. Sensors may be scattered randomly over a geographical region and in order to save battery energy they may undergo cycles of *sleeping - active* periods [3]. Furthermore, nodes deployed in real fields might get damaged, or just fail at any time. As a result, the topology of the network may be varying in time in an unpredictable manner. For this reason routing algorithms to carry messages from a sensor node to a sink in a multi-hop fashion should rapidly adapt to the changing topology. A survey of routing algorithms is in [2].

Swarm intelligence (SI) techniques [9] are population-based stochastic methods in which the collective behavior of relatively simple individuals arises from their local interactions to produce global patterns. Through the adoption of the swarm intelligence concept, it is possible to design distributed, self-organizing, and fault tolerant routing protocols able to self-adapt to the environmental changes. The main properties of swarm intelligence are that: *i)* Single nodes are assumed to be simple with low computational intelligence and communication capabilities; *ii)* Nodes communicate indirectly, i.e., messages are not directed to any particular node; *iii)* The range of the messages may be very short, nevertheless a robust global behavior emerges from the interaction of the nodes; *iv)* The global behavior adapts to the environmental changes.

SI in WSN is inspired from the observation on how ant colonies forage for food. Ants tend to move along paths of high pheromone intensity and release pheromone during their passage thus reinforcing the pheromone trail. However, pheromone evaporates allowing the system to forget old information and randomly search for new solutions. In this way, large groups of simple agents, interacting only locally with neighboring agents, work together to coordinate their actions toward fulfilling a common goal. In such systems, modeling the state of the entire system as a cross-product of the states of individual nodes results in the well-known state explosion problem. In fact, the usual way to study these systems is through simulation [12, 16]. An attempt to tackle the problem analytically is in [14], but the analysis is limited to the asymptotic behavior of a two-nodes two-links system.

This paper describes how the performance analysis of large

VALUETOOLS 2009, October 20-22, Pisa, Italy

DOI 10.4108/ICST.VALUETOOLS2009.7672

SI systems composed by interacting agents can be modeled and investigated by resorting to a recently defined new entity called *Markovian Agent (MA)* [5, 6]. An MA is a discrete-state continuous-time Markov chain (CTMC) governed by a local transition rate matrix but able to send messages to other agents, and whose behavior can be modified by the messages received from other agents. Furthermore, agents are located in a geographical space, and their interaction depends on the relative positions and is governed by a suitable *perception function*.

This paper is aimed to show that a large system of interacting MAs can be analytically solved to generate a pheromone gradient around the sink(s) that can be successively used to create the routing table along the steepest gradient in order to minimize the number of hops from each node to the sink(s). In particular, we provide a stochastic model to analyze a swarm-based routing protocol that gets inspiration from the one presented in [12]. According to it, pheromone information is stored at each node, and the algorithm starts with the sink agent(s) emitting a message with the highest pheromone level; node agents that receive the pheromone message update their pheromone level and transmit it to the neighbors; at the same time nodes are subject to an evaporation process that reduces their stored pheromone intensity. In analogy with the biological systems, here data packets move around the network carrying the pheromone (like ants) and leaving it on the way. We assume in this paper that sensors are spread over a regular mesh, with at most one sensor in each cell. Even if the transmission range limits the activity of the pheromone messages to the first neighboring cells, the pheromone gradient rapidly forms over the entire region.

An analytical model for the system is presented and the numerical analysis is provided. Several examples illustrate how the proposed solution reacts to characteristic parameters (like emission and evaporation rates) and to different topological configurations. A detailed study is presented to show that increasing the range of activity of the pheromone messages the time to reach a stabilized gradient decreases but, at the same time, the energy consumption increases. Finally, a scenario composed of thousand of nodes has been studied demonstrating how the proposed analytical technique is suitable in the performance evaluation of very large networking systems.

This paper is organized as follows: In Section 2, we briefly revise routing protocols in WSN describing the swarm-based algorithm we want to model. In Section 3, we introduce the Markovian Agents and we present the proposed SI model. In Section 4, we describe the adopted performance indices and we show the results of several experiments. Finally, in Section 5 we provide concluding remarks and future works.

2. GRADIENT-BASED ROUTING PROTOCOLS IN WSN

WSN's specific characteristics make routing different from traditional wireless ad-hoc networks [17]. The modest processing power, the absence of global information, the high directionality of the flow (from multiple nodes to few sinks), the data redundancy, give rise to the definition of several routing algorithms [2]. Different routing metrics can be exploited with respect to different goals: hop count, energy consumption, Quality of Service, throughput, network lifetime [15, 11, 1]. However, the main philosophy is that what is important is the information that nodes contain, not the nodes themselves. Such assumption leads routing in WSN to become data-centric as opposed to node-centric [17]. In this context, gradient-based routings allow to establish routes to the sinks following such paths that respect criteria related to data typologies, network topology, and to nodes' status. In *Directed Diffusion* [10], sinks generate information requests and diffuse them through the network. During the diffusion phase, nodes build gradients associated with each interest request that are used to direct the information flow back to the sink. *GBR* [15] extends the Directed Diffusion allowing nodes to keep the number of hops to the sink (*height* of a node) thus realizing a gradient of heights. Packets are then forwarded through the link with the shortest gradient. In order to take into account energy consumption aspects, the height of each node can be also correlated to the battery level.

From an autonomic perspective, the need to set-up flexible, adaptive, and scalable networks calls for routing algorithms easy to manage and able to react to the environmental changes. In the last years, SI has been applied to autonomic networking systems [4, 8] demonstrating its feasibility and efficiency in adapting to highly dynamic distributed systems. In analogy to the biological process of pheromone emission, in [12] each node sends a signalling routing packet containing its pheromone level and updates such value based on the level of its neighbors, thus creating a pheromone gradient toward the sink. The routing task is driven by the pheromone level of the network: data packets are forwarded toward the highest pheromone density zone and reach the sink following the pheromone gradient. Any change on the network condition will be reflected by an update of the pheromone level of the involved nodes; changes on the pheromone gradient will automatically drive the routing decisions toward the new optimal solution. In this way, the network can self-organize its topology and adapt itself to environmental changes. Differently from Directed Diffusion and GBR, when a link failure occurs, the network reorganization task has not to be accomplished by the sink but only by those nodes near to the broken link, thus resulting in a robust and self-organized architecture.

In the following, we will describe a swarm-based algorithm that gets inspiration from the one presented in [12]. Since our purpose is to study the gradient construction process, we will focus on the signalling component of the routing protocol avoiding details about data forwarding. Routing paths toward the sink are established through the exchange of pheromone packets containing the pheromone level p of each node. We assume to have P discrete different pheromone levels, ranging from 0 to $P-1$. The gradient construction is triggered by sinks that, during the data collection period, maintain the highest level of pheromone $(P-1)$. The gradient construction protocol is described by the Algorithms 1 and 2, differentiated with respect to the node type: sink or sensor.

Sink nodes, once activated, set their internal pheromone level p to the highest value, (Algorithm 1: line 1). Then, they periodically send to their neighbors a pheromone packet containing p (Algorithm 1: lines 5-7). The time period is defined by the timer $T1$.

The pheromone level of a sensor node is initially set to the lowest value (Algorithm 2: line 1) and then it is up-

Digital Object Identifier: 10.4108/ICST.VALUETOOLS2009.7672
http://dx.doi.org/10.4108/ICST.VALUETOOLS2009.7672

dated following an *excitation - evaporation* process. Sensor nodes periodically send a pheromone packet containing their internal pheromone level. This activity is scheduled at fixed interval times by appropriately setting the timer $T1$ (Algorithm 2: line 2) and then, when the timer expires, a packet is sent to all the neighbors (Algorithm 2: line 12-14).

Excitation is triggered by the reception of the pheromone level from a neighbor (Algorithm 2: lines 6-10). As a consequence, the node updates its own level when a greater value is received; the new value is computed as a function of the current and the received pheromone level $update(p, p_n)$. In the Algorithm 2, we use $update(p, p_n) = round((p + p_n)/2)$ (line 9).

The evaporation mechanism is triggered at the expiration of the timer $T2$ (Algorithm 2: lines 15), and it simply decreases the value of p (Algorithm 2: lines 15-16), assuring it maintains a value greater or equal to 0.

Algorithm 1 Sink nodes

```
1: p ← P − 1
2: setTimer(T1)
3: loop
4:    e ← waitForEvent()
5:    if e = TIMER_EXPIRED then
6:       send(p)
7:    end if
8: end loop
```

Algorithm 2 Sensor nodes

```
1: p ← 0
2: setTimer(T1)
3: setTimer(T2)
4: loop
5:    e ← waitForEvent()
6:    if e = DATA_RECEIVED then
7:       p_n ← getDataReceived()
8:       if p_n > p then
9:          p ← update(p, p_n)
10:      end if
11:   else if e = TIMER_EXPIRED then
12:      t ← getTimer()
13:      if t = T1 then
14:         send(p)
15:      else {t = T2}
16:         p ← max(0, p − 1)
17:      end if
18:   end if
19: end loop
```

3. THE MARKOVIAN AGENT MODEL

Markovian Agents Models (MAMs) [5] represent systems as a collection of agents spread over a geographical space. Each agent is described by a finite state machine where two types of transitions can happen: *local transitions* and *induced transitions*. A local transition occurs whenever an agent changes its state due to information perceived on its environment, whereas induced transitions occur as a consequence of the interactions with other agents. Interactions are possible through message exchanging: when a local transition occurs, the agent can send a message and the receiving agent can ignore or accept it. In the second case, the accepting agent changes its state performing an induced transition.

MAs are spread over a finite geographical area $\mathcal{V}$ that can be either continuous or discrete. In case of a continuous space we have that $\mathcal{V} \subset \mathbb{R}^d$, where d is an integer number representing the dimension of the space. We denote by $\rho(\mathbf{v}) : \mathcal{V} \rightarrow \mathbb{R}^+$ the spatial density function of the agents. In particular, it is defined such that for every d-dimensional volume A in $\mathcal{V}$ (with $A \subseteq \mathcal{V}$) the number of agents in A is distributed according to a Poisson distribution with mean $\int_A \rho(\mathbf{v})d\mathbf{v}$. In this paper we will focus on the 2-dimensional case, and consider $d = 2$.

Message exchanging is governed by a function that takes into account the agent distribution over the space, the message routing policy, and the transmittance properties of the medium. This function is called the *perception function* and it is denoted by $u(\cdot)$. The definition of the perception function is quite general, and allows to model several message routing strategies and *MA* interdependencies. In particular, the receiving agent can be aware of the state in which the agent that issued the message was, and use this information to take an appropriate action.

In order to model an heterogeneous system such as a WSN with several sensor and sink nodes, we need to extend the MAM adding the capability to represent different types of exchanged messages (as in [6]) and to consider several classes of agents. Formally a *Multiple Agent Class, Multiple Message Type* Markovian Agents Model (M^3AM) is defined by the tuple:

$$M^3AM = \{\mathcal{C}, \mathcal{M}, \mathcal{V}, \mathcal{U}, \mathcal{R}\} \tag{1}$$

where:

$\mathcal{C} = \{1 \dots C\}$ is a set of classes of agents. We denote with MA^c an agent of class $c \in \mathcal{C}$.

$\mathcal{M} = \{1 \dots M\}$ is a set of types of messages. Each agent (independently of its class) can send or receive messages of type $m \in \mathcal{M}$.

$\mathcal{V}$ is the finite space over which Markovian Agents are spread.

$\mathcal{U} = \{u_1(\cdot) \dots u_M(\cdot)\}$ is a set of M perception functions (one for each message type), and $\mathcal{R} = \{\rho^1(\cdot) \dots \rho^C(\cdot)\}$ is a set of C agent density functions (one for each agent class).

Each agent MA^c of class c is characterized by n_c states, and it is defined by the tuple:

$$MA^c = \{\mathbf{Q^c}, \mathbf{\Lambda}^c, \mathbf{G}^c(m), \mathbf{A}^c(m), \boldsymbol{\pi}_0^c\} \tag{2}$$

Where:

$\mathbf{Q^c} = |q_{ij}^c|$ is the $n_c \times n_c$ infinitesimal generator matrix of the continuous time Markov chain that describes the behavior of the agent. Its entry q_{ij}^c, with $i \neq j$, represents the transition rate from state i to state j of an agent of class c. We define $q_{ii}^c = -\sum_{j \neq i}^{n_c} q_{ij}^c$.

$\mathbf{\Lambda}^c = |\lambda_i^c|$, is a vector of size n_c whose components represent the rate of *self-jumps* for an agent of class c, that is the rate at which the corresponding Markov chain reenters the same state. Self-jumps are required to allow an agent to send messages without changing the state.

$\mathbf{G}^c(m) = |g_{ij}^c(m)|$ is a $n_c \times n_c$ matrix, that describes the probability that an agent of class c that jumps from state i to state j generates a message of type m. The elements of $G^c(m)$ must respect the restriction $\sum_{m=1}^{M} g_{ij}^c(m) \leq 1$, $\forall c, i, j$ to ensure that during a transition an agent can generate at most one message.

Digital Object Identifier: 10.4108/ICST.VALUETOOLS2009.7672
http://dx.doi.org/10.4108/ICST.VALUETOOLS2009.7672

$\mathbf{A}^c(m) = |a_{ij}^c(m)|$ is a $n_c \times n_c$ matrix, that describes the acceptance probability of messages of type m for an agent of class c. A message is dropped with probability $a_{ii}^c(m)$, and it is accepted with probability $1 - a_{ii}^c(m)$. In the latter case, the agent immediately jumps to state j ($j \neq i$) with probability $a_{ij}^c(m)$, and $\sum_{j \neq i} a_{ij}^c(m) = 1 - a_{ii}^c(m)$, $\forall c, i, m$. This implies that rows of matrix $\mathbf{A}$ sum to 1.

$\boldsymbol{\pi}_0^c$, is a probability vector of size n_c which represents the initial state distribution of an agent of class c.

The perception function: $u_m : \mathcal{V} \times \mathcal{C} \times \mathbb{N} \times \mathcal{V} \times \mathcal{C} \times \mathbb{N} \rightarrow \mathbb{R}^+$ is defined such that the values of $u_m(\mathbf{v}, c, i, \mathbf{v}', c', i')$ represent the probability that an agent of class c, in position $\mathbf{v}$, and in state i, perceives a message m generated by an agent of class c' in position $\mathbf{v}'$ in state i'. As pointed out, this general definition of $u(\cdot)$ allows to model several message routing strategies and MA interdependencies.

3.1 Analysis

A M^3AM model can be analyzed solving a set of coupled ordinary differential equations. Let us call $\rho_i^c(t, \mathbf{v})$ the density of agents of class c, in state i, and located in position $\mathbf{v}$ at time t. We have that $\sum_{i=1}^{n_c} \rho_i^c(t, \mathbf{v}) = \rho^c(\mathbf{v})$, $\forall t$ (that is, we consider that the total density of agents in a position $\mathbf{v}$ remains constant over the time). We collect the state densities into a vector $\boldsymbol{\rho}^c(t, \mathbf{v}) = |\rho_i^c(t, \mathbf{v})|$. We will be interested in computing the transient evolution of $\boldsymbol{\rho}^c(t, \mathbf{v})$.

We start by defining $\beta_i^c(m)$ as the total rate at which messages of type m are generated by an agent of class c in state i:

$$\beta_i^c(m) = \sum_{j \neq i} q_{ij}^c \, g_{ij}^c(m) + \lambda_i^c \, g_{ii}^c(m) \tag{3}$$

This rate can be used to compute $\gamma_{ii}^c(t, \mathbf{v}, m)$, the total rate of messages of type m received by an agent of class c in state i at position $\mathbf{v}$ at time t:

$$\gamma_{ii}^c(t, \mathbf{v}, m) = \int_{\mathcal{V}} \sum_{c'=1}^{C} \sum_{i'=1}^{n_{c'}} \rho_{i'}^{c'}(t, \mathbf{v}') u_m(\mathbf{v}, c, i, \mathbf{v}', c', i') \beta_{i'}^{c'}(m) d\mathbf{v}' \tag{4}$$

We can collect them into a diagonal matrix $\boldsymbol{\Gamma}^c(t, \mathbf{v}, m) = \text{diag}(\gamma_{ii}^c(t, \mathbf{v}, m))$. This matrix can be used to compute $\mathbf{K}^c(t, \mathbf{v})$, the infinitesimal generator of an agent of class c in position $\mathbf{v}$ at time t:

$$\mathbf{K}^c(t, \mathbf{v}) = \mathbf{Q}^c + \sum_m \boldsymbol{\Gamma}^c(t, \mathbf{v}, m) \left[\mathbf{A}^c(m) - \mathbf{I}\right] \tag{5}$$

The evolution of the entire model, can be studied by solving $\forall \mathbf{v}, c$ the following ordinary differential equations:

$$\begin{cases} \boldsymbol{\rho}^c(0, \mathbf{v}) & = & \rho^c(\mathbf{v}) \boldsymbol{\pi}_0^c \\ \dfrac{d\boldsymbol{\rho}^c(t, \mathbf{v})}{dt} & = & \boldsymbol{\rho}^c(t, \mathbf{v}) \mathbf{K}^c(t, \mathbf{v}) \end{cases} \tag{6}$$

From the density of agents in each state, we can compute the probability of finding an agent of class c at time t in state i as:

$$\pi_i^c(t, \mathbf{v}) = \frac{\rho_i^c(t, \mathbf{v})}{\rho^c(\mathbf{v})} \tag{7}$$

and we collect all the terms in a vector $\boldsymbol{\pi}^c(t, \mathbf{v}) = |\pi_i^c(t, \mathbf{v})|$.

Equation (6) can be solved using conventional discretization techniques. In this paper we used an implicit method.

3.2 Model description

We model the protocol described in Section 2 with two Markovian Agent classes: one addressed as *sink* (Figure 1(a)) representing sink nodes, and the other called *node* (Figure 1(b)) representing sensor nodes. Each state of an agent is drawn with a circle. Each local transition is represented by a solid arrow and it is labelled with the corresponding transition rate. A dashed arrow starting from a solid arc denotes the generation of a message during the corresponding transition. The label of the dashed arrow specifies the type of message generated. Induced transitions are represented with dashed arcs. Each induced transition arc is labelled with a message type. The transition occurs only if the agent receives a message of the specified type.

We consider a model composed of N sensors, distributed over a rectangular grid of $n_h \times n_w$ square cells, with $N \leq n_h \cdot n_w$ and at most one sensor per cell. We call d_s the length of the cell side: all the sensors are spaced at a distance of d_s. We discretize the space $\mathcal{V}$ of the Markovian Agent Model exactly as the $n_h \times n_w$ sensor grid. The pheromone intensity is discretized into P levels (ranging from 0 to $P-1$) that identify also the maximum number of message types ($M = P$). We use a different message type for each possible pheromone level, and define $\mathcal{M} = \{0, 1, \ldots, P-1\}$.

The *sink* class has a very simple behavior, characterized by a single state. At a constant rate λ, sink nodes emit a message representing the maximum pheromone intensity, that is message $P-1$. Rate $\lambda = \frac{1}{T1}$ reflects the duration of timer $T1$ of the algorithms presented in Section 2.

Agents belonging to the *node* class, use their state space to represent their current pheromone level. They are thus characterized by P states, each corresponding to a different pheromone level $0, \ldots, P-1$. All the nodes starts in state 0. In each state a self-loop of rate $\lambda = \frac{1}{T1}$ models the firing of timer $T1$, and the transmission of the pheromone level to the neighboring nodes. The pheromone level decreases at rate $\mu = \frac{1}{T2}$, which models the firing of timer $T2$. In Figure 1(b), this is represented by the solid arc that connects each state with the previous one. The key part of the algorithm is implemented in the arcs that model the reception of the messages, and the corresponding definition of the acceptance matrix $\mathbf{A}^{node}(m)$. In particular, when a node in state i receives a message of type m, it immediately jumps to state j if $m \in M(i, j)$, with:

$$\begin{aligned} M(i, j) = \{m \in [0 \cdots P-1] : round((m + i)/2) = j\} \\ \forall i, j \in [0 \cdots P-1] : i > j. \end{aligned} \tag{8}$$

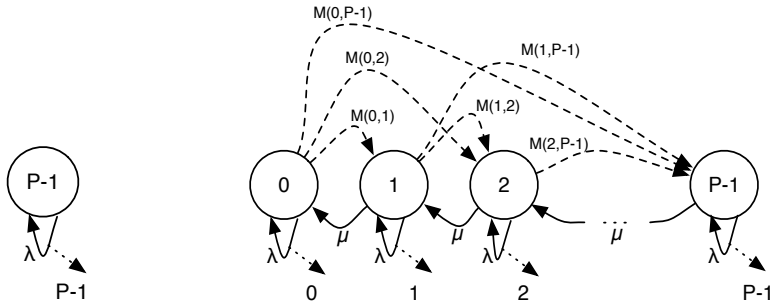
that is, an agent jumps to a state that represents the pheromone level that is the mean between the current level and the one encoded in the message. In the acceptance matrix we then have that $a_{ij}^{node}(k) = 1$, $\forall k \in M(i, j)$ and that $a_{ij}^{node}(h) = 0$, $\forall h \notin M(i, j)$.

Each sensor is characterized by a transmission range t_r. The definition of function $u(\cdot)$ is such that it allows only the nodes in transmission range to receive the pheromone update message. In particular we have that $\forall m, c, i, \mathbf{v}$:

$$u_m(\mathbf{v}, c, i, \mathbf{v}', c', i') = \begin{cases} 0 & \text{dist}(\mathbf{v}, \mathbf{v}') > t_r \\ 1 & \text{dist}(\mathbf{v}, \mathbf{v}') \leq t_r \end{cases} \tag{9}$$

where $\text{dist}(\mathbf{v}, \mathbf{v}')$ represents the distance between two sensors in position $\mathbf{v}$ and $\mathbf{v}'$. The definition of the perception function in equation (9) models a broadcast of a message in

Digital Object Identifier: 10.4108/ICST.VALUETOOLS2009.7672
http://dx.doi.org/10.4108/ICST.VALUETOOLS2009.7672



(a) Agent class = "sink". (b) Agent class = "node".

Figure 1: Markovian agent models.

a radius of t_r around the transmitting sensor. We can use the definition of the perception function to determine the mean number of neighbor nodes $\eta(\mathbf{v})$ of a node in position $\mathbf{v}$ as:

$$\eta(\mathbf{v}) = \sum_{v' \in \mathcal{V}} \sum_{c' \in \mathcal{C}} \left(\rho^{c'}(\mathbf{v}') u_m(\mathbf{v}, c, i, \mathbf{v}', c', i') - \rho^{c'}(\mathbf{v}) \right) d_s^2 \tag{10}$$

Note that since the perception function $u(\cdot)$ depends only on $\mathbf{v}$ and $\mathbf{v}'$, the choice of c, i and i' in equation (10) can be arbitrary. Remember also that the space has been discretized in such a way to have at most one sensor in each cell of the grid. This allows us to "count" the sensors by simply summing $\rho^c(\mathbf{v}) d_s^2$. We can define the maximum mean number of neighbors $\bar{\eta}$ as:

$$\bar{\eta} = \max_{v \in \mathcal{V}} \eta(\mathbf{v}) \tag{11}$$

Note that $\bar{\eta}$ is a global property of the network that is a function of the transmission range t_r and of the sensor distance (i.e., the length of the cell side d_s).

4. PERFORMANCE EVALUATION

In this section we show how the proposed model is able to represent the algorithm for the gradient creation, and how it can be used to evaluate appropriate measures that can be exploited to set up the algorithm parameters in a real WSN.

4.1 Measures of interest

As explained in Section 3, each MA represents a sensor node and it is able to sense other nodes when their radio signals have an adequate power. This physical property is modeled through the perception function $u()$. In the following, we will refer to sensed nodes from position $\mathbf{v}$ as *neighbors*, and we define $\mathcal{N}(\mathbf{v}) \subseteq \mathcal{V}$ as the set of their positions.

The algorithm behavior depends on both the pheromone emission rate λ and the pheromone evaporation rate μ; the excitation - evaporation process strictly depends on the number of agents sensed in a given position because the number of perceived messages increase with them, strengthening the level of pheromone into a node. To take into account this physical phenomenon, we refers to the following quantity

$$r = \frac{\lambda \cdot \overline{\eta}}{\mu} \tag{12}$$

that gives the relative speed of global emission and evaporation processes.

We estimate the pheromone level p of an agent of class *node* in position $\mathbf{v}$ at time t as its expected state

$$\phi(t, \mathbf{v}) = \sum_{i=0}^{P-1} i \cdot \pi_i^{node}(t, \mathbf{v}) \tag{13}$$

where $\pi_i^{node}(t, \mathbf{v})$ are computed by solving equations (6) and (7). The value of $\phi(t, \mathbf{v})$ over $\mathcal{V}$ represents the distribution of pheromone on the considered area of interest.

Since we want to evaluate whether the swarm-based algorithm correctly operates in creating a "well formed" pheromone distribution over $\mathcal{V}$, we need to define an estimator of the gradient quality. Our estimator is based on the fact that a gradient-based routing algorithm forwards the packets of node in position $\mathbf{v}$ toward the neighbor with the greatest pheromone level, whether it exists, as far as the sink is reached. We denote with $\langle \mathbf{v}, t \rangle$ the agent in position $\mathbf{v}$ sending a data packet to the sink at time t. To do that, it selects the neighbor with the greatest pheromone level greater than its own, whether it exists. If such node does not exist, data is not sent. More formally, let $l(\mathbf{v}, \mathbf{v}', t) = \phi(\mathbf{v}', t) - \phi(\mathbf{v}, t)$ be the amount of pheromone [1] of agent $\langle \mathbf{v}', t \rangle$ with respect to agent $\langle \mathbf{v}, t \rangle$; we define

$$l_m(\mathbf{v}, t) = \begin{cases} \max_{\mathbf{v}' \in \mathcal{N}(\mathbf{v})} l(\mathbf{v}, \mathbf{v}', t) & \phi(\mathbf{v}', t) > \phi(\mathbf{v}, t) \\ 0 & otherwise \end{cases}$$

as the increment of the pheromone level along a path toward the sink when a packet is sent from the node in position $\mathbf{v}$ at time t. The mean increased pheromone value when a node sends a packet to the sink is the considered estimator, and can be computed as:

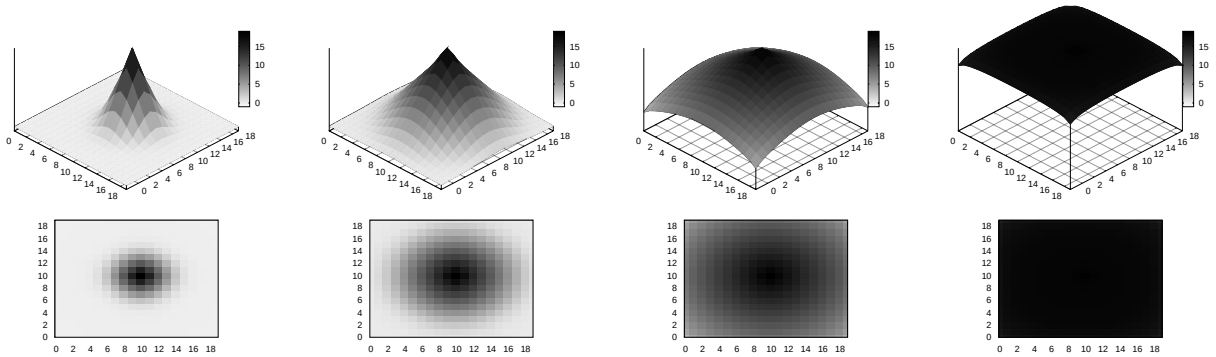
$$\overline{l}_m(t) = \frac{1}{N} \sum_{\mathbf{v} \in \mathcal{V}} l_m(\mathbf{v}, t), \tag{14}$$

where N is the number of sensor nodes in the area of interest. We use $\overline{l}_m(t)$ as the index measuring the goodness of the pheromone gradient distribution; a high value of $\overline{l}_m(t)$ means a high increase of the mean level of pheromone along the path towards the sink owing the maximum level, and as a consequence, low number of hops.

Another practical performance index in a real WSN is the time the pheromone gradient can be considered established. As before, let us consider the agent $\langle \mathbf{v}, t \rangle$; we say that it

[1] $l(\mathbf{v}, \mathbf{v}', t)$ could be a negative quantity, meaning that $\langle \mathbf{v}', t \rangle$ has a lower level than $\langle \mathbf{v}, t \rangle$.

Digital Object Identifier: 10.4108/ICST.VALUETOOLS2009.7672
http://dx.doi.org/10.4108/ICST.VALUETOOLS2009.7672



(a) r=1.0. (b) r=1.5. (c) r=2.0. (d) r=2.5.

Figure 2: Pheromone gradient over $\mathcal{V}$ under different values of r.

is in a *stable state* when its level of pheromone $\phi(t, \mathbf{v})$ does not vary any more; since $\phi(t, \mathbf{v})$ depends on $\boldsymbol{\pi}^{node}(t, \mathbf{v})$, we estimate the *stable state* as the first time where

$$\left\| \frac{\partial \boldsymbol{\pi}^{node}(t, \mathbf{v})}{\partial t} \right\| \leq \varepsilon \tag{15}$$

To numerically compute the relation (15), we evaluate the discrete derivative

$$\left\| \frac{\Delta \boldsymbol{\pi}^{node}(t, \mathbf{v})}{\Delta t} \right\| = \left\| \frac{\boldsymbol{\pi}^{node}(t, \mathbf{v}) - \boldsymbol{\pi}^{node}(t - \Delta t, \mathbf{v})}{\Delta t} \right\|, \tag{16}$$

where Δt is the discretization step, and evaluating the set

$$T_s(\mathbf{v}) = \left\{ t \in [0, +\infty] : \left\| \frac{\Delta \boldsymbol{\pi}^{node}(t, \mathbf{v})}{\Delta t} \right\| \leq \varepsilon \right\} \tag{17}$$

that defines the time interval when $\langle \mathbf{v}, t \rangle$ is in the stable state.

The first time the node in position $\mathbf{v}$ will reach a stable condition with respect to the pheromone level variation is given by:

$$t_s(\mathbf{v}) = \inf T_s(\mathbf{v}) \tag{18}$$

Since the overall network reaches the stability when all the nodes are into a stable state, the time for stability will be:

$$\widetilde{t} = \max_{\mathbf{v} \in \mathcal{V}} t_s(\mathbf{v}) \tag{19}$$

4.2 Performed measures

The above mentioned performance indices have been computed under different conditions in order to test the swarm intelligent algorithm and to provide insights into the settings of the algorithm parameters. First of all, we are interested in the evaluation of the system dependency on r that represents the most critical parameter. The model was solved using a grid of sizes $n_h = 20$ and $n_w = 20$, where sensors are uniformly distributed with a spatial density equal to 1, resulting in a grand total of $N = 400$ sensors. We also placed a sink node in the center of the area (i.e., in position $\mathbf{v} = (10, 10)$) fixing $\lambda = 4.0$, $P = 20$, and $\overline{\eta} = 4$. The numerical solution is computed fixing $\Delta t = 0.001 sec$ and $\varepsilon = 0.005$.

The first system property we want to analyze is the quality of the pheromone gradient. Figure 2 shows the pheromone

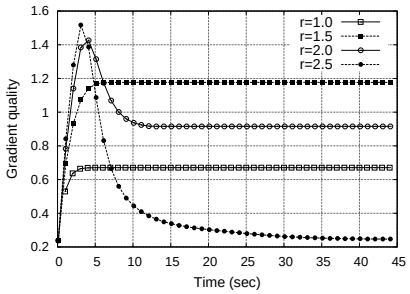
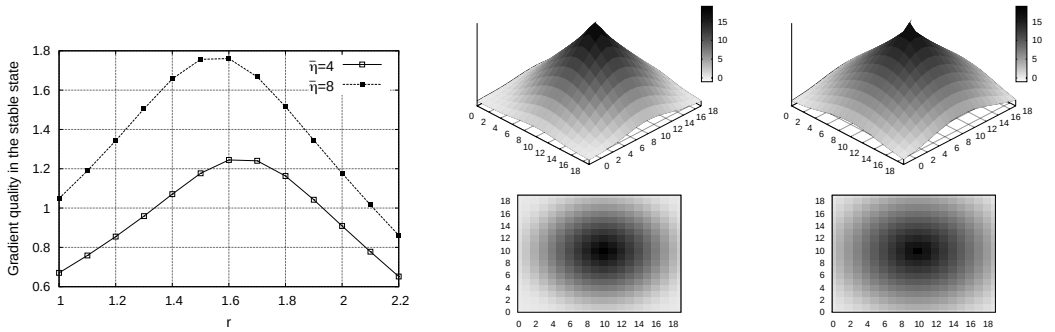


Figure 3: Transient analysis of the gradient quality estimator $\overline{l}_m(t)$ under different values of r.

distribution over $\mathcal{V}$ with respect to r in the stable state, i.e., at time $\widetilde{t}$. To improve the graph readability, each map is plotted both in 3D and 2D views. It can be noticed that the parameter r has a direct impact on the shape of the pheromone gradient. In particular, if r is too small ($r = 1.0$) or too high ($r = 2.5$), the quality of the gradient is poor. In fact, high values of the evaporation rate (Figure 2(a)) do not allow the pheromone diffusion thus reducing the area covered by the sink. On the contrary, low values of the evaporation rate (Figure 2(d)) give rise to a saturation of the pheromone level in the network that makes the routing strategy useless. Intermediate values, although giving rise to different gradient shapes, generate well formed pheromone gradients able to cover the whole area.

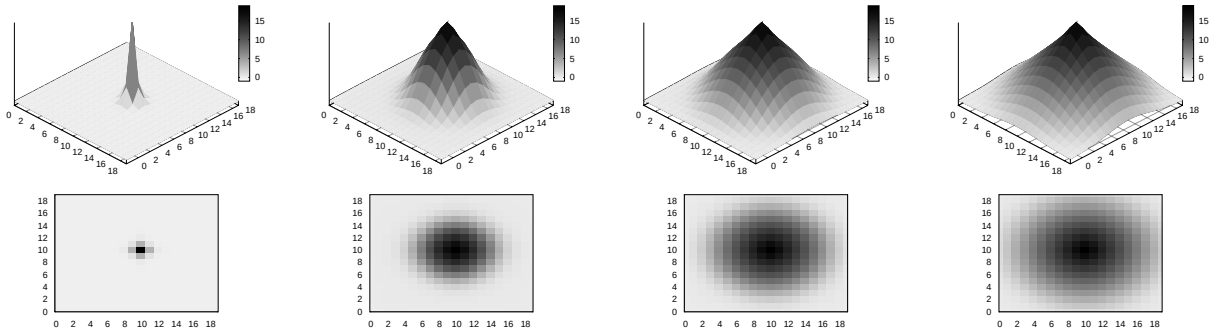
In order to provide a formal validation of the pheromone gradient construction process, we compute the gradient quality estimator $\overline{l}_m(t)$ for different values of r. Figure 3 shows that, for low values of r, $\overline{l}_m(t)$ exhibits a monotonic behavior and the values reached at the stable state increase with the increase of r (curve $r = 1.0$ and $r = 1.5$). Such trend is inverted for high values of r, due to the saturation phenomenon that can be observed as a rapid decrease of $\overline{l}_m(t)$ over time (curve $r = 2.0$ and $r = 2.5$) that produces decreasing values of $\overline{l}_m(t)$ at the stable state. In order to evaluate the value r^* of r that maximizes $\overline{l}_m(t)$, in Figure 4(a), we plot the value of the gradient quality estimator in the stable condition ($\overline{l}_m(\widetilde{t})$), varying r from 1.0 to 2.2. As expected, observing the curve $\overline{\eta} = 4$, it is possible to identify a maximum value of $\overline{l}_m(\widetilde{t})$, obtaining $r^* = 1.6$. The corresponding

Digital Object Identifier: 10.4108/ICST.VALUETOOLS2009.7672
http://dx.doi.org/10.4108/ICST.VALUETOOLS2009.7672



(a) (b) (c)

Figure 4: (a) Gradient quality estimator in the stable condition ($\overline{l}_m(\widetilde{t})$) with respect to r varying $\overline{\eta}$. (b) Pheromone gradient over $\mathcal{V}$ when $r = r^* = 1.6$ ($\overline{\eta} = 4$). (c) Pheromone gradient over $\mathcal{V}$ when $r = r^* = 1.6$ ($\overline{\eta} = 8$).



(a) t=0.1 sec. (b) t=2.0 sec. (c) t=4.0 sec. (d) t=10.0 sec.

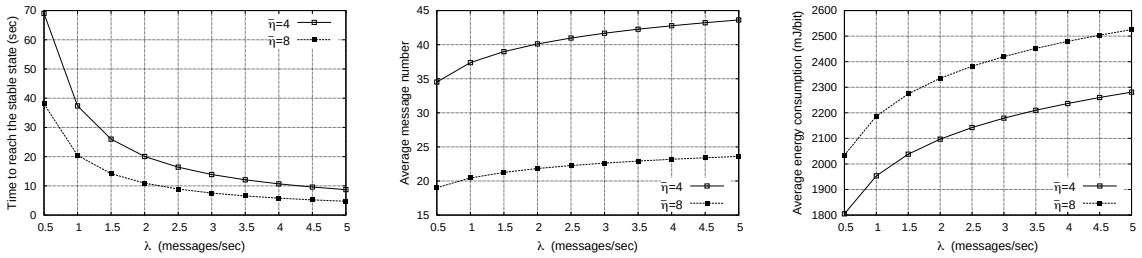
Figure 5: Growth of the pheromone gradient over $\mathcal{V}$ at different time intervals until the stable condition is reached ($r = r^* = 1.6$).

pheromone distribution over $\mathcal{V}$ is plotted in Figure 4(b). To better analyze the transient evolution of the pheromone distribution over $\mathcal{V}$, in Figure 5 we plot four graphs, taken at different time intervals, that highlight the gradient growth until the stable state is reached.

After computing the optimal value of r under the above assumptions, we are interested in evaluating the influence of network conditions on the protocol setting phase. In the next experiment, we increased the transmission range, obtaining a maximum mean neighbor number $\overline{\eta} = 8$ (instead of $\overline{\eta} = 4$), and we applied the same methodology to compute r^*. In the following, we assume sensor and sink nodes to have the same transmission range. Curve $\overline{\eta} = 8$ in Figure 4(a) shows that it is possible to identify a maximum value of $\overline{l}_m(\widetilde{t})$ even if the neighbor number has changed. The above graph also demonstrates that the qualitative trend of $\overline{l}_m(\widetilde{t})$ has not significantly changed, giving also in this case a value of r^* equal to 1.6. The different quantitative results obtained, can be explained by pointing out that a greater value of $\overline{\eta}$ gives rise to a decrease of the average hop number needed to reach the sink; this results in greater values of the pheromone level increment in one step and then in an increase of the value of $\overline{l}_m(\widetilde{t})$. In Figure 4(c), we plot the pheromone distribution over $\mathcal{V}$ setting $\overline{\eta} = 8$ and $r = r^* = 1,6$. Comparing Figures 4(b) and 4(c) we can observe that the produced gradients are quite similar. Such result is obtained thanks to the balancing of the excitation - evaporation process expressed by the parameter r that allows to easily adapt the swarm-based protocol in environments where the node density is not uniform.

Once the parameter r has been set, the pheromone gradient quality is not influenced by the pheromone emission rate λ. However, changes of λ can have a strong influence on other system properties, such as network timing. For this reason, it is important to carefully estimates such dependencies in order to properly set the value of λ. In Figure 6(a), the time $\widetilde{t}$ to reach a stable state is plotted as a function of the pheromone emission rate, fixing $r = r^*$ in both $\overline{\eta} = 4$ and $\overline{\eta} = 8$ conditions. It can be observed that increasing either λ or $\overline{\eta}$, shorter values of $\widetilde{t}$ are obtained. This trend is due to an improvement on the information distribution in the network that allows to propagate the pheromone signal faster; it is correlated to the following causes: *i)* An increase of λ results in an increase of the number of messages per second received by each node, allowing them to reach their stable pheromone level faster; *ii)* An increase of the neighbor number increments the number of nodes that are able to receive a message, thus producing a more pervasive infor-

Digital Object Identifier: 10.4108/ICST.VALUETOOLS2009.7672
http://dx.doi.org/10.4108/ICST.VALUETOOLS2009.7672



(a) (b) (c)

Figure 6: (a) Time to reach the stable condition ($\widetilde{t}$) versus λ varying $\overline{\eta}$, (b) corresponding average signalling overhead per node ($\overline{m}$), and (c) corresponding average energy consumption per node ($\overline{e}$).

mation dissemination. However, as λ assumes high values, the gap between the two curves is reduced, demonstrating that when the number of messages exchanged per second is high, the above mentioned second cause is mitigated.

An important aspect related to the time to reach the stable state is the average number $\overline{m}$ of signalling messages sent by each node. We can define such parameter starting from the pheromone emission rate as:

$$\overline{m} = \lambda \cdot \widetilde{t} \tag{20}$$

In Figure 6(b) the value of $\overline{m}$ is plotted as a function of the pheromone emission rate and varying the neighbor number. The graph shows that the average number of message needed to reach the stable state increases with λ. Even in this case, an increase of $\overline{\eta}$ from 4 to 8 gives rise to an improvement of the performance index, denoted by lower values of $\overline{m}$. In fact, a greater value of $\overline{\eta}$ allows to reach a greater number of nodes with the same number of sent messages, thus reducing the global number of messages needed to reach the stable state. However, since $\overline{\eta}$ is related to the transmission range of each node we need to take into account also energy consumptions aspects, in the computation of the cost needed to reach the stable condition. As described in [7], the energy cost per bit $E(t_r)$ required to exchange a message between a node and its $\overline{\eta}$ neighbors can be computed as:

$$E(t_r) = (1 + \overline{\eta})(E^{(ele)} + E^{(proc)}) + C_d \cdot t_r^{\alpha} \tag{21}$$

where $E^{(ele)}$ and $E^{(proc)}$ are the consumptions due to the transceiver electronics and the processing functions, C_d is a constant factor, t_r is the transmission range needed to cover the distance between the sender and the receivers, as defined in Section 3.1, and α is the exponential power decay factor. Assuming to have a regular grid where nodes are uniformly distributed, it is possible to express the value of $\overline{\eta}$ as a function of t_r and d_s. In fact, since d_s is the distance between any two adjacent nodes, the transmission range to obtain a value of $\overline{\eta} = 4$ will be $t_{r4} \geq d_s$, while to obtain a value of $\overline{\eta} = 8$ it will be $t_{r8} \geq d_s \cdot \sqrt{2}$. The average energy cost per node needed to reach the stable state can then be expressed as:

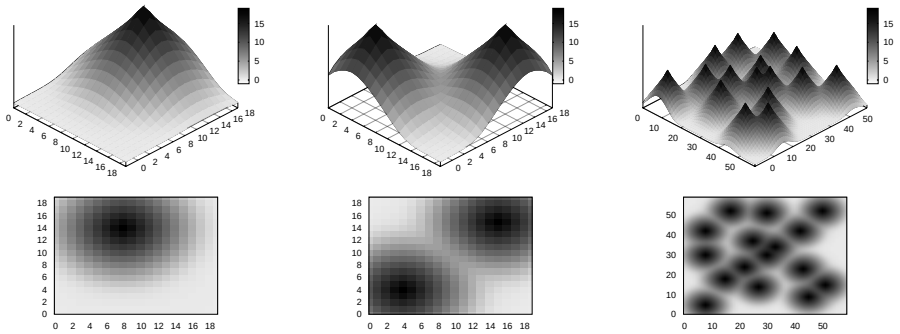
$$\overline{e} = E(t_r) * \overline{m} \tag{22}$$

In accordance with [13], we use the following values for setting the parameters in (21): $E^{(ele)} = E^{(proc)} = 0.15mJ/bit$, $C_d = 0.018mJ/(bit \cdot m^{\alpha})$, and $\alpha = 2.5$. Moreover, assuming to have a square area of $380 \times 380m^2$ we set $d_s = 20m$, $t_{r4} = 24m$, and $t_{r8} = 32m$. In Figure 6(c) we observe that, as expected, the average energy consumption needed to reach the stable state increases as λ increases. However, a comparison between Figures 6(b) and 6(c) shows that, notwithstanding the lower value of messages needed, when $\overline{\eta} = 8$ we obtain a greater energy cost than the case $\overline{\eta} = 4$. Using the proposed model, it is then possible to estimate the cost associated to the time needed to reach the stable state and, in order to respond to particular application-specific requirements, a trade-off between $\widetilde{t}$ and $\overline{e}$ can be opportunely found during the setting phase of the network. In the following, when not explicitly expressed, we will refer to a value of $\overline{\eta}$ equal to 4.

Next experiments aim at analyze the pheromone gradient construction process by leaving the regular network topology conditions assumed so far. First of all, we focus our attention on the sink position. In Figure 7 we plot the pheromone distribution over $\mathcal{V}$ in the stable state placing the sink in a decentralized position (Figure 7(a)) and placing two sinks in the area (7(b)). Through such diagrams, we can assess that the pheromone gradient is reached also when no symmetries are present in the network and that the proposed model is able to capture the behavior of the protocol in generating a correct pheromone gradient also in presence of different maximums. In order to test the model in more complex scenarios, in Figure 7(c),we considered a larger uniformly distributed grid of $1180 \times 1180m^2$, that corresponds to $n_h = 60$ and $n_w = 60$, with a grand total of $N = 3600$ sensors, with 16 sinks placed in random locations. Figure 7(c) validates the gradient construction process. In fact, using the same protocol configurations found for a simple scenario, the SI algorithm is able to create a well formed pheromone gradient also in a completely different situation, making such routing technique suitable in non predictable scenarios. Such scenario also demonstrates the scalability of the proposed analytical technique that can be easily adopted in the analysis of very large networks.

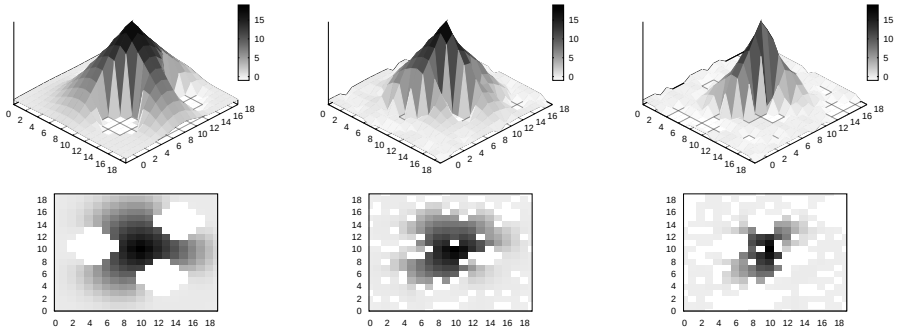
The robustness of the algorithm when the neighbor number is not constant has been analyzed through the study of the pheromone gradient in scenarios where some sensor nodes are removed from the network, as shown in Figure 8. We analyzed two different situations. In the first one (Figure 8(a)), contiguous areas of nodes are removed from the network, reproducing scenarios where nodes fail due to conditions strictly related to the geographic position. In the second one (Figures 8(b) and 8(c)), a percentage of nodes is randomly removed from the network, thus producing ir-

Digital Object Identifier: 10.4108/ICST.VALUETOOLS2009.7672
http://dx.doi.org/10.4108/ICST.VALUETOOLS2009.7672



(a) (b) (c)

Figure 7: Pheromone gradient over $\mathcal{V}$ in the stable condition when (a) the sink is decentralized, (b) two sinks are present in the network, and (c) the network is composed by a grid of 3600 sensor nodes with 16 sinks.



(a) (b) (c)

Figure 8: Pheromone gradient over $\mathcal{V}$ when some nodes are removed from the network. (a) Contiguous areas, (b) 20% of nodes, and (c) 35% of nodes.

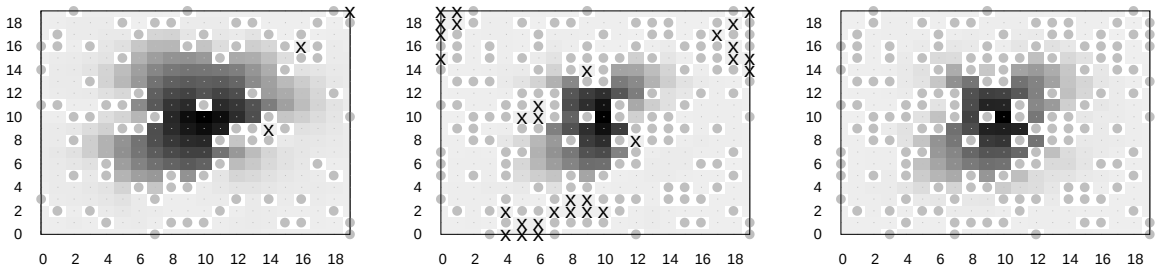
regular network topologies. Node failures can be observed as white spots in the graphs. From the analysis of the left plot, we observe that the SI behavior allows to create a useful gradient by diffusing the pheromone level around the areas where node failures occurred, without degrading its performance. When the network topology becomes very irregular, as shown in the center and right plots, the gradient construction process still works forcing the obtained gradient to follow the network topology. However, increasing the number of nodes removed from the network (Figure 8(c)), the gradient quality rapidly decreases, as can be noticed from the enlargement of the white areas in the graph. Such phenomenon can be explained as follow. Figure 9 shows the pheromone distribution over $\mathcal{V}$ in the above conditions (20 % and 35% of nodes removed from the network) highlighting the positions of broken nodes (gray circles). When nodes are removed from the network, other nodes happen to be "isolated" due to the break of paths toward the sink. Such nodes (highlighted in the graphs through "x" symbols) increase when the topology becomes very irregular giving rise to the enlargement of the areas not covered by the sink and thus reducing the gradient quality. Notwithstanding such side effect, the swarm-based protocol diffuses the pheromone level around the broken nodes allowing to carry out routing decisions among those nodes still connected to the sink.

To reduce the rise of disconnected areas, it is possible to increase the transmission range of each node. In Figure 9(c), we plot the pheromone distribution when exactly the same nodes of Figure 9(b) are removed from the network but with the maximum mean neighbor number $\overline{\eta}$ fixed to 8. It is possible to observe that the increased range of activity of each node allows to reduce the presence of isolated nodes thus increasing the pheromone gradient quality. In this way, we are able to improve the network reliability to the detriment of the power consumption and then of the network lifetime.

5. CONCLUSIONS

The analytical study of the gradient formation in large WSN with up to many thousands of nodes has been carried out resorting to systems of interacting Markovian Agents. Swarm intelligence mechanisms in which the global behavior is build up starting from very short range interactions (each agent is able to interact only with its first neighbors) have proved to be particularly suited to be analyzed with the MA technique proposed in this paper. Various examples show the adaptability of the algorithm to the changing conditions of the WSN, in terms of parameter values and topology.

Digital Object Identifier: 10.4108/ICST.VALUETOOLS2009.7672
http://dx.doi.org/10.4108/ICST.VALUETOOLS2009.7672



(a) 20% of nodes, $\overline{\eta}$=4. (b) 35% of nodes, $\overline{\eta}$=4. (c) 35% of nodes, $\overline{\eta}$=8.

Figure 9: Removed nodes (circles), isolated nodes (X), and the pheromone distribution over $\mathcal{V}$ in the stable condition.

6. ACKNOWLEDGMENT

This work has been partially supported by MIUR fund through the "Programma di Ricerca Scientifica di Rilevante Interesse Nazionale 2007" (PRIN 2007) under grant 2007J4SKYP.

7. REFERENCES

[1] K. Akkaya and M. Younis. Energy and QoS Aware Routing in Wireless Sensor Networks. *Cluster Computing*, 8:179–188, 2005.

[2] K. Akkaya and M. Younis. A survey on routing protocols for wireless sensor networks. *Elsevier Ad Hoc Networks*, 3:325–349, 2005.

[3] G. Anastasi, M. Conti, M. D. Francesco, and A. Passarella. How to prolong the lifetime of wireless sensor networks. *Chapter 6 in Mobile Ad Hoc and Pervasive Communications*.

[4] G. D. Caro and M. Dorigo. Ant Colonies for Adaptive Routing in Packet-Switched Communications Networks. *Lecture Notes in Computer Science - Parallel Problem Solving from Nature*, 1498:673–682, 1998.

[5] D. Cerotti, M. Gribaudo, and A. Bobbio. Analysis of on-off policies in sensor network using markovian agents. In *4-th Int. Workshop PerSens*, pages 300–305, 2008.

[6] D. Cerotti, M. Gribaudo, and A. Bobbio. Disaster Propagation in Heterogeneous Media via Markovian Agents. In *3rd International Workshop on Critical Information Infrastructures Security*, 2008.

[7] C.-F. Chiasserini, R. Gaeta, M. Garetto, M. Gribaudo, D. Manini, and M. Sereno. Fluid models for large-scale wireless sensor networks. *Performance Evaluation*, 64(7-8):715 – 736, 2007.

[8] X. Cui, T. Hardin, R. K. Ragade, and A. S. Elmaghraby. A swarm-based fuzzy logic control mobile sensor network for hazardous contaminants localization. In *IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, pages 194–203, 2004.

[9] M. Hinchey, R. Sterritt, and C. Rouff. Swarms and swarm intelligence. *IEEE Computer*, pages 111–113, April 2007.

[10] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM 00) Boston*, pages 56–67, 2000.

[11] L. Lin, N. B. Shroff, and R. Srikant. Energy-aware routing in sensor networks: A large system approach. *Elsevier Ad Hoc Networks*, 5(6):818–831, 2007.

[12] M. Paone, L. Paladina, D. Bruneo, and A. Puliafito. A Swarm-based Routing Protocol for Wireless Sensor Networks. In *6th IEEE International Symposium on Network Computing and Applications (NCA '07)*, pages 265–268, 2007.

[13] T. S. Rappaport and T. Rappaport. *Wireless Communications: Principles and Practice (2nd Edition)*. Prentice Hall PTR, December 2001.

[14] M. Roth and S. Wicker. Asymptotic pheromone behavior in swarm intelligent manets: An analytical analysis of routing behavior. In *Sixth IFIP IEEE International Conference on Mobile and Wireless Communications Networks (MWCN)*, 2004.

[15] C. Schurgers and M. Srivastava. Energy efficient routing in wireless sensor networks. In *Communications for network-centric operations: creating the information force (MILCOM 01)*, 2001.

[16] I. Stojmenovic. Simulations in Wireless Sensor and Ad Hoc Networks: Matching and Advancing Models, Metrics, and Solutions. *IEEE Communications Magazine*, 46(12):102–107, 2008.

[17] F. Zhao and L. Guibas. *Wireless sensor networks - An information processing approach*. Morgan Kaufmann, 2004.

Digital Object Identifier: 10.4108/ICST.VALUETOOLS2009.7672
http://dx.doi.org/10.4108/ICST.VALUETOOLS2009.7672