A Proces-Algebraic Framework for Estimating the Energy Consumption in Ad-hoc Wireless Sensor Networks*

L. Gallina, A. Marin, S. Rossi
Università Ca' Foscari Venezia, Italy
{lgallina,marin,srossi}@dais.unive.it

T. Han, M. Kwiatkowska
University of Oxford, United Kingdom
{marta.kwiatkowska,tingting.han}@cs.ox.ac.uk

ABSTRACT

We present a framework for modelling ad-hoc Wireless Sensor Networks (WSNs) and studying both their connectivity properties and their performance in terms of energy consumption, throughput and other relevant indices. Our framework is based on a probabilistic process calculus where system executions are driven by Markovian probabilistic schedulers, allowing us to translate process terms into discrete time Markov chains (DTMCs) and use the probabilistic model checker PRISM to automatically evaluate/estimate the connectivity properties and the energy costs of the networks. To the best of our knowledge, this is the first work that proposes a unique framework for studying qualitative (e.g., by proving the equivalence of components or the correctness of a behaviour) and quantitative aspects of WSNs using a tool that allows both exact and approximate (via Monte Carlo simulation) analyses. We demonstrate our framework at work by considering different communication strategies based on gossip routing protocols, for a typical topology and a mobility scenario.

Categories and Subject Descriptors

C.4 [**Computer Systems Organization**]: Performance of systems—*modeling techniques*; C.2.1 [**Computer-Communication Networks**]: Network architecture and design—*Wireless Communication*

General Terms

Theory, reliability

Keywords

Process algebra, simulation, sensor networks

*Work partially supported by the MIUR Project CINA "Compositionality, Interaction, Negoziation, Autonomicity for the future ICT society" and ERC Advanced Grant VERIWARE.

1. INTRODUCTION

Wireless Sensor Networks (WSNs) [2] are collections of spatially distributed sensing devices equipped with limited computing and radio communication capabilities. They are employed in a variety of applications, ranging from military surveillance to health care or assisted living, and from smart cities to environmental monitoring.

A typical sensor network is characterized by a large number of sensor nodes, which are densely deployed, and have frequent topology changes due to the mobility of its devices. Nodes communicate using wireless transmission in a specified range, with communication between nodes implemented in terms of routing protocols.

A critical issue in wireless sensor networks is the limited availability of energy within the network devices. Therefore, judicious choice of routing protocols that can reduce the nodes' power consumption is crucial, not only for the performance of each single node, but also throughout the network lifetime.

In this paper we introduce a framework for the specification, modelling and automated analysis or simulation of connectivity properties and the evaluation or estimation of energy consumption in ad-hoc WSNs. The framework is based on a variant of the Probabilistic Energy-aware Broadcast Unicast and Multicast (PEBUM) calculus introduced in [5], and supports the automatic performance evaluation through (approximate) probabilistic model checking, e.g., using the PRISM model checker [9].

The advantage of using a process algebra is due to its compositional nature, which allows us to decompose both the model construction and the qualitative analysis. Concerning the quantitative analysis, we utilise the various kinds of properties which can be specified in terms of a temporal logic and verified using model checking.

The calculus we propose is built around nodes, representing the sensor devices of the systems, and locations, identifying the position cells across which each device may move inside the network. Node mobility is governed by probability distributions. By contrast, wireless synchronisations are controlled by sequential processes inside the nodes: each transmission broadcasts a message within a given transmission range. The semantics of our calculus is inspired by Segala's probabilistic automata [15] driven by schedulers to resolve the nondeterministic choice among the probability distributions over target states.

Differently from [5], in this work we assume that nodes are not equipped with a unique identifier and they all share the same transmission frequency. These choices reflect the fact

Networks		Processes	
M,N ::= $\mathbf{0}$	Empty network	P,Q ::= $\mathbf{0}$	Inactive process
$\mid [P]_l^{\mathbf{J}}$	Sensor Node	$\mid (\tilde{x}).P$	Input
$\mid M_1 \mid M_2$	Parallel composition	$\mid \langle \tilde{w} \rangle_{L,r}.P$	Output
		$\mid [w_1 = w_2]P, Q$	Matching
		$\mid A\langle \tilde{w} \rangle$	Recursion

Table 1: Syntax

that transmissions in ad-hoc sensor networks are directed to a geographical location rather than to a specific node and, due to the low-cost hardware of sensors, only one frequency is used at a given time [17]. Moreover, in contrast to [6, 5], in this paper we employ *Markovian probabilistic schedulers*, mapping the non-deterministic choices among the different actions a system may enable into probability distributions. As a consequence, the labelled transition system underlying process terms is a discrete time Markov chain (DTMC), which can be used for automatically performing a range of qualitative and quantitative analyses by means of probabilistic model checking, e.g., using PRISM. We define a probabilistic observational congruence in the style of [13] to equate networks exhibiting the same connectivity behaviour. As in [5], and in contrast to [12], the notion of observability is associated with specific locations in the network, reflecting the fact that in ad-hoc WSNs the transmissions are not addressed to specific nodes but to specific locations. We provide a coinductive characterization of the observational congruence based on a probabilistic labelled bisimilarity. Finally, we define an energy-aware preorder over networks, to contrast networks having the same behaviour, but different energy costs.

We use our framework for a comparative study of gossip based routing protocols for wireless ad-hoc sensor networks. We address the problem of state space explosion using statistical model checking implemented in PRISM in terms of Monte Carlo simulation. Specifically, we consider different scenarios obtained by varying both the protocol parameters and the network power strategies, in order to find the best solution that reduces the power consumption while maintaining the same connectivity.

The paper is organised as follows. Section 2 presents the calculus, its observational semantics expressed in terms of behavioural equivalences and a characterization based on a notion of probabilistic bisimilarity. In Section 3 an energy-aware preorder over networks is defined: it allows us to compare the average energy cost of different networks but exhibiting the same connectivity behaviour. Finally, in Section 4, gossip based routing protocols for different scenarios are considered, and we show the framework at work studying the sensitivity of the performance of these protocols to some configuration parameters. Finally, Section 5 discusses the related bibliography and concludes the paper.

2. A CALCULUS FOR WSN

We present a variant of the PEBUM calculus introduced in [5], which focuses on the main features of ad-hoc wireless sensor networks. Specifically, nodes are not equipped with a unique identifier and only one transmission frequency is used.

Syntax. We use letters l for *locations*, r for *transmission radii*, x and y for *variables*. *Closed values* contain locations, transmission radii and any basic value (booleans, integers, ...). *Values* include also variables. We use u and v for closed values and w for (open) values. We write $\tilde{v}$, $\tilde{w}$ for tuples of values. We write $\mathcal{N}$ for the set of all networks and **Loc** for the set of all locations. While movement may be assumed to be continuous, we identify locations as the countable set of cells that constitute the observing areas within the network.

The syntax of our calculus is shown in Table 1. This is defined in a two-level structure: the lower one for processes, the upper one for networks.

Networks are collections of sensor nodes running in parallel and communicating messages. As usual, $\mathbf{0}$ denotes the empty network and $M_1|M_2$ denotes the parallel composition of two networks. We denote by $\prod_{i\in I} M_i$ the parallel composition of the networks M_i, for $i \in I$. $[P]_l^{\mathbf{J}}$ denotes a sensor node located at the physical location l and executing the process P. $\mathbf{J}$ is the transition matrix of a discrete time Markov chain modelling node mobility: each entry $\mathbf{J}_{lk}$ is the probability that the sensor node located at l moves to the location k. Hence, $\sum_{k\in\mathbf{Loc}} \mathbf{J}_{lk} = 1$ for all locations $l \in \mathbf{Loc}$. Static nodes inside a network are associated with the identity Markov chain, i.e., the identity matrix $\mathbf{J}_{ll} = 1$ for all $l \in \mathbf{Loc}$ and $\mathbf{J}_{lk} = 0$ for all $l \neq k$.

Processes are sequential and live within the nodes: $\mathbf{0}$ is the inactive process, $(\tilde{x}).P$ is ready to listen to a transmission, while $\langle \tilde{w} \rangle_{L,r}.P$ is ready to transmit. In $(\tilde{x}).P$, the variables in $\tilde{x}$ are bound with scope in P. As to the output form, the tag r represents the transmission radius of the sender, while the tag L is used to maintain the set of physical locations of the intended recipients: $L = \mathbf{Loc}$ represents a broadcast transmission, while a finite set of locations L denotes a multicast communication (unicast if L is a singleton). As stated in the introduction, communication protocols for ad-hoc sensor networks are usually intended to reach a certain location, rather than a specific device, due to the absence of global identifiers associated with the sensor devices. The remaining syntactic forms are standard: $[w_1 = w_2]P, Q$ behaves as P if $w_1 = w_2$, and as Q otherwise. $A\langle \tilde{w} \rangle$ is the process defined via a (possibly recursive) definition $A(\tilde{x}) \stackrel{\text{def}}{=} P$, with $|\tilde{x}| = |\tilde{w}|$ where $\tilde{x}$ contains all variables appearing free in P.

Probability distributions for networks. We denote by $\mu_l^{\mathbf{J}}$ the probability distribution associated with a node located at l with transition matrix $\mathbf{J}$, i.e., the function over **Loc** such that $\mu_l^{\mathbf{J}}(k) = \mathbf{J}_{lk}$ for all $k \in \mathbf{Loc}$. We model the probabilistic evolution of the network according to these distributions.

Let M be a network. We denote by $M\{[P]_k^{\mathbf{J}}/[P]_l^{\mathbf{J}}\}$ the network obtained by replacing l with k inside the sensor node $[P]_l^{\mathbf{J}}$. We also denote by $[\![M]\!]_{\mu_l^{\mathbf{J}}}$ the probability distribution over networks induced by $\mu_l^{\mathbf{J}}$ and defined by: for all

$$\text{(R-Bcast)}\ \frac{}{[\langle\tilde{v}\rangle_{L,r}.P]^{\mathbf{J}}_{l} \mid \prod_{i\in I}[(\tilde{x}_i).P_i]^{\mathbf{J}^i}_{l_i} \rightarrow [\![[P]^{\mathbf{J}}_{l} \mid \prod_{i\in I}[P_i\{\tilde{v}/\tilde{x}_i\}]^{\mathbf{J}^i}_{l_i}]\!]_{\Delta}} \quad \text{where } \forall i \in I. d(l,l_i) \leq r \text{ and } |\tilde{x}_i| = |\tilde{v}|$$

$$\text{(R-Move)}\ \frac{}{[P]^{\mathbf{J}}_{l} \rightarrow [\![[P]^{\mathbf{J}}_{l}]\!]_{\mu^{\mathbf{J}}_{l}}} \qquad \text{(R-Par)}\ \frac{M \rightarrow [\![M']\!]_{\theta}}{M|N \rightarrow [\![M'|N]\!]_{\theta}} \qquad \text{(R-Struct)}\ \frac{N \equiv M \ \ M \rightarrow [\![M']\!]_{\theta} \ \ M' \equiv N'}{N \rightarrow [\![N']\!]_{\theta}}$$

Table 2: Reduction Semantics

networks M',

$$[\![M]\!]_{\mu^{\mathbf{J}}_{l}}(M') \ = \ \begin{cases} \mu^{\mathbf{J}}_{l}(k) & \text{if } M' = M\{[P]^{\mathbf{J}}_{k}/[P]^{\mathbf{J}}_{l}\} \\ 0 & \text{otherwise.} \end{cases}$$

Intuitively, $[\![M]\!]_{\mu^{\mathbf{J}}_{l}}(M')$ is the probability that the network M evolves to M' due to the movement of the sensor node $[P]^{\mathbf{J}}_{l}$. We say that M' is in the support of $[\![M]\!]_{\mu^{\mathbf{J}}_{l}}$ if $[\![M]\!]_{\mu^{\mathbf{J}}_{l}}(M') \neq 0$. We write $[\![M]\!]_{\Delta}$ for the Dirac distribution on the network M, i.e., the probability distribution defined as: $[\![M]\!]_{\Delta}(M) = 1$ and $[\![M]\!]_{\Delta}(M') = 0$ for all $M' \neq M$. Finally, we let θ range over $\{\mu^{\mathbf{J}}_{l} \mid \mathbf{J}$ is a transition matrix and $l \in \mathbf{Loc}\} \cup \{\Delta\}$.

Reduction semantics. The dynamics of the calculus is specified by the *probabilistic reduction relation* ($\rightarrow$) described in Table 2: it takes the form $M \rightarrow [\![M']\!]_{\theta}$ denoting a transition that leaves from M and leads to a probability distribution $[\![M']\!]_{\theta}$. As usual, reduction relies on structural congruence ($\equiv$), such that, e.g., $M|N \equiv N|M$, $(M|N)|M' \equiv M|(N|M')$ and $M|\mathbf{0} \equiv M$.

Nodes cannot be created or destroyed, and move autonomously. Node connectivity is verified by looking at the physical location and the transmission radius of the sender: a message broadcast by a node is received only by the nodes that lie in the area delimited by the transmission radius of the sender. We presuppose a function $d(\cdot,\cdot)$ which returns the distance between two locations.

Rule (R-Bcast) models the transmission of a tuple of messages $\tilde{v}$ by a sensor node located at l and using a radius r. The index set I may be empty, i.e., the rule can be applied even if no nodes are ready to receive. The radius r associated with the output action denotes the transmission radius of that communication which may depend on the energy consumption strategy adopted by the surrounding protocol. All the nodes that lie in the range of the sender (i.e., such that $d(l,l_i) \leq r$) will receive the messages. Rule (R-Move) deals with node mobility: a node $[P]^{\mathbf{J}}_{l}$ executing a move action will reach a location with a probability described by the distribution $\mu^{\mathbf{J}}_{l}$ that depends on the Markov chain $\mathbf{J}$ statically associated with the node. The remaining rules are standard.

Since we are dealing with a probabilistic reduction semantics, which maps networks into probability distributions, we need a way of representing the steps of each probabilistic evolution of a network. Formally, given a network M, we write $M \rightarrow_{\theta} N$ if $M \rightarrow [\![M']\!]_{\theta}$ and N is in the support of $[\![M']\!]_{\theta}$. Following [6], an execution for M is a (possibly infinite) sequence of steps $M \rightarrow_{\theta_1} M_1 \rightarrow_{\theta_2} M_2 \ldots$.

Observational semantics. According to standard practice, we formalise the observational semantics of our calculus in terms of a notion of *barb* that provides the basic unit of observation [13]. As in other calculi for wireless communication, the definition of barb is naturally expressed in terms of message transmission.

We denote by $behave(M) = \{[\![M']\!]_{\theta} \mid M \rightarrow [\![M']\!]_{\theta}\}$ the set of the possible behaviours of M. In order to solve the nondeterminism in a network execution, we consider each possible probabilistic transition $M \rightarrow [\![M']\!]_{\theta}$ as arising from a *probabilistic scheduler* defined as follows.

Definition 1 (Scheduler). *A* probabilistic scheduler *is a total function F assigning to a network M a distribution ϕ on the set $behave(M)$.*

We denote by *Sched* the set of all probabilistic schedulers. Given a network M and a scheduler F, we define the set of all executions starting from M and driven by F as:

$$\begin{aligned} Exec^F_M = \{ & e = M_0 \rightarrow_{p_1\theta_1} M_1 \rightarrow_{p_2\theta_2} M_2 \ldots \mid M_0 \equiv M \text{ and} \\ & \forall j > 0 :\ M_{j-1} \rightarrow [\![M'_j]\!]_{\theta_j},\ p_j = F(M_{j-1})([\![M'_j]\!]_{\theta_j}) \\ & \text{and } M_j \text{ is in the support of } [\![M'_j]\!]_{\theta_j} \}. \end{aligned}$$

For a finite execution $e = M \rightarrow_{p_1\theta_1} M_1 \ldots \rightarrow_{p_k\theta_k} M_k \in Exec^F_M$ starting from M and driven by a scheduler F we define

$$P^F_M(e) = p_1 \cdot [\![M'_1]\!]_{\theta_1}(M_1) \cdot \ldots \cdot p_k \cdot [\![M'_k]\!]_{\theta_k}(M_k)$$

where $\forall j \leq k,\ p_j = F(M_{j-1})([\![M'_j]\!]_{\theta_j})$. We denote by $last(e)$ the final state of a *finite* execution e, by e^j the prefix execution $M \rightarrow_{p_1\theta_1} M_1 \ldots \rightarrow_{p_j\theta_j} M_j$ of length j of the execution $e = M \rightarrow_{p_1\theta_1} M_1 \cdots \rightarrow_{p_j\theta_j} M_j \rightarrow_{p_{j+1}\theta_{j+1}} M_{j+1} \cdots$, and by $e\uparrow$ the set of $\bar{e}$ such that $e \leq_{prefix} \bar{e}$. We write $M \rightarrow^F_* M'$ if there exists a finite execution $e \in Exec^F_M$ with $last(e) = M'$.

We define the probability space on the executions starting from a given network M as follows. Given a scheduler F, $\sigma Field^F_M$ is the smallest sigma field on $Exec^F_M$ that contains the basic cylinders $e\uparrow$, where $e \in Exec^F_M$. The probability measure $Prob^F_M$ is the unique measure on $\sigma Field^F_M$ such that $Prob^F_M(e\uparrow) = P^F_M(e)$. Given a measurable set of networks H, we denote by $Exec^F_M(H)$ the set of executions starting from M and crossing a state in H. Formally $Exec^F_M(H) = \{e \in Exec^F_M \mid last(e^j) \in H \text{ for some } j\}$. We denote the probability for a network M to evolve into a network in H, according to the policy given by F, as $Prob^F_M(H) = Prob^F_M(Exec^F_M(H))$.

Note that the use of probabilistic schedulers allows us to model networks as discrete time Markov chains (DTMCs). This is the result of the application of a *two level* probability distribution: the reduction semantics maps a network M into a probability distribution in the set $behave(M)$ while, in turn, the probabilistic scheduler maps M into a probability distribution ϕ over the probability distributions in the set $behave(M)$, giving rise to a fully probabilistic model.

$$\text{(Output)}\ \frac{-}{\langle\tilde{v}\rangle_{L,r}.P \xrightarrow{\tilde{v}_{L,r}} P} \qquad \text{(Input)}\ \frac{-}{(\tilde{x}).P \xrightarrow{\tilde{v}} P\{\tilde{v}/\tilde{x}\}}$$

$$\text{(Then)}\ \frac{P \xrightarrow{\eta} P'}{[\tilde{v} = \tilde{v}]P, Q \xrightarrow{\eta} P'} \qquad \text{(Else)}\ \frac{Q \xrightarrow{\eta} Q' \quad \tilde{v}_1 \neq \tilde{v}_2}{[\tilde{v}_1 = \tilde{v}_2]P, Q \xrightarrow{\eta} Q'} \qquad \text{(Rec)}\ \frac{P\{\tilde{v}/\tilde{x}\} \xrightarrow{\eta} P'}{A\langle\tilde{v}\rangle \xrightarrow{\eta} P'} \quad A(\tilde{x}) \stackrel{\text{def}}{=} P$$

Table 3: LTS rules for Processes

EXAMPLE 1. *Consider the network $M \equiv [\langle\tilde{v}\rangle_{L,r}.P]_l^{\mathbf{J}}$ consisting of a single sensor node. The set of possible behaviours of M is $\{[\![[P]_l^{\mathbf{J}}]\!]_\Delta, [\![[\langle\tilde{v}\rangle_{L,r}.P]_l^{\mathbf{J}}]\!]_{\mu_l^{\mathbf{J}}}\}$, since the sensor node at the next step can either move or transmit. Then, for each $F \in Sched$, $\forall e \in Exec_N^F$ such that $N \in \mathcal{N}$ and $last(e) = M$ we get $F(e) = \phi$ such that there exist p_1 and p_2 with $p_1 + p_2 \leq 1$ and for all $M' \in \mathcal{N}$:*

$$\phi(M') = \begin{cases} p_1 & \text{if } M' \equiv [P]_l^{\mathbf{J}} \\ p_2 * q_i & \text{if } M' \equiv [\langle L, r\rangle_v.P]_k^{\mathbf{J}} \text{ and} \\ & [\![[\langle L, r\rangle_v.P]_l^{\mathbf{J}}]\!]_{\mu_l^{\mathbf{J}}}([\langle L, r\rangle_v.P]_k^{\mathbf{J}}) = q_i \\ 0 & \text{otherwise.} \end{cases}$$

The notion of barb introduced below denotes an observable transmission with a certain probability according to a fixed scheduler. We first introduce the notion of *strong barb*: for a network M, we write $M\downarrow_K$ when $M \equiv [\langle\tilde{v}\rangle_{L,r}.P]_l^{\mathbf{J}}|M'$ with $\emptyset \neq K \subseteq L$ and forall $k \in K$, $d(l,k) \leq r$. Roughly, a transmission is observable only if at least one location in the set of the intended recipients is able to receive the message. We say that a network M has a *barb* with probability p at the set K of locations, according to the scheduler F, written $M\Downarrow_p^F K$, if $Prob_M^F(\{M'|M \rightarrow_*^F M'\downarrow_K\}) = p$. Intuitively, for a given network M and scheduler F, if $M\Downarrow_p^F K$ then there is a positive probability that M, driven by F, performs a transmission and at least one of the intended recipients is able to correctly listen to it.

In the following, we introduce a notion of probabilistic observational congruence relative to a specific set of schedulers $\mathcal{F} \in Sched$. Since our semantics is contextual, we need to ensure that the set of schedulers we consider allows the specific networks we analyse to interact with any possible context. Hence, for a set $\mathcal{F}$ of schedulers, we define the *contextual superset* $\mathcal{F}_\mathcal{C}$ of $\mathcal{F}$ as the largest set of schedulers allowing networks to interact with any possible context even when driven by $\mathcal{F}$ (see [4] for a formal definition). It holds that $Sched_\mathcal{C} = Sched$. Hereafter, a context $C[\cdot]$ is a term with a hole defined by the grammar: $\mathcal{C}[\cdot] ::= [\cdot] \mid [\cdot]|M \mid M|[\cdot]$.

Our probabilistic observational congruence relative to a specific set of schedulers is defined as follows.

DEFINITION 2. *Given a set $\mathcal{F} \subseteq Sched$ and a relation $\mathcal{R}$ over networks:*
- $\mathcal{R}$ is barb preserving *relative to $\mathcal{F}$ if $M\mathcal{R}N$ and $M\Downarrow_p^F K$ for some $F \in \mathcal{F}_\mathcal{C}$ implies that there exists $F' \in \mathcal{F}_\mathcal{C}$ such that $N\Downarrow_p^{F'} K$.*
- $\mathcal{R}$ is reduction closed *relative to $\mathcal{F}$ if $M\mathcal{R}N$ implies that for all $F \in \mathcal{F}_\mathcal{C}$ there exists $F' \in \mathcal{F}_\mathcal{C}$ such that for all classes $\mathcal{C} \in \mathcal{N}/\mathcal{R}$, $Prob_M^F(\mathcal{C}) = Prob_N^{F'}(\mathcal{C})$.*
- $\mathcal{R}$ is contextual *if $M\mathcal{R}N$ implies that $\mathcal{C}[M]\,\mathcal{R}\,\mathcal{C}[N]$ for every context $\mathcal{C}[\cdot]$.*
- Probabilistic observational congruence *relative to $\mathcal{F}$, written $\cong_p^\mathcal{F}$, is the largest symmetric relation over networks which is reduction closed, barb preserving and contextual.*

Two networks are related by $\cong_p^\mathcal{F}$ if they exhibit the same probabilistic (connectivity) behaviour relative to $\mathcal{F}$.

In the next section a bisimulation-based proof technique for $\cong_p^\mathcal{F}$ is developed in order to provide an efficient method to check whether two networks are related by $\cong_p^\mathcal{F}$.

Deciding the Observational Congruence. We express the semantics of the calculus in terms of labelled transition systems (LTS) which are built upon two sets of rules: one for processes and one for networks. Table 3 presents the LTS rules for processes. Transitions are of the form $P \xrightarrow{\eta} P'$, where η ranges over input and output actions: $\eta ::= \tilde{v}\,|\,\tilde{v}_{L,r}$.

Table 4 presents the LTS rules for networks. Transitions are of the form $M \xrightarrow{\gamma} [\![M']\!]_\theta$, where M is a network and $[\![M']\!]_\theta$ is a distribution over networks. Probabilities are used to model the mobility of nodes. Tag γ ranges over the labels:

$$\gamma ::= L!\tilde{v}[l, r] \mid ?\tilde{v}@l \mid R!\tilde{v}@K \mid \tau.$$

Rule (Snd) models the sending of tuple $\tilde{v}$ to a specific set L of locations with transmission radius r, while rule (Rcv) models the reception of $\tilde{v}$ at l. Rule (Bcast) models the broadcast message propagation: all the nodes lying within the transmission cell of the sender may receive the message, regardless of the fact that they lie in one of the locations in L. Rule (Obs) models the observability of a transmission: every transmission may be detected (and hence *observed*) by any recipient lying in one of the observation locations within the transmission cell of the sender. The label $R!\tilde{v}@K$ represents the transmission of the tuple $\tilde{v}$ of messages: the set R is the set of all the locations receiving the message, while its subset K contains only the locations where the transmission is observed. Rule (Lose) models message loss. As usual, τ-transitions denote non-observable actions. Rule (Move) models node mobility according to the probability distribution $\mu_l^{\mathbf{J}}$. Finally, (Par) is standard.

Based on the LTS semantics, we define a probabilistic labelled bisimilarity that is a characterisation of our *probabilistic observational congruence*. It is built upon the actions:

$$\alpha ::= ?\tilde{v}@l \mid R!\tilde{v}@K \mid \tau.$$

We write $lbehave(M)$ for the set of all possible behaviors of M, that is, $lbehave(M) = \{(\alpha, [\![M']\!]_\theta) \mid M \xrightarrow{\alpha} [\![M']\!]_\theta\}$. Labelled executions arise by resolving the non-determinism of both α and $[\![M]\!]_\theta$. As a consequence, a scheduler[1] for the labelled semantics is a function F assigning a probability to each pair $(\alpha, [\![M]\!]_\theta) \in lbehave(M)$ with a network

[1]We abuse notation and still use F to denote a scheduler for the LTS semantics.

$$(\text{Snd})\ \frac{P \xrightarrow{\tilde{v}_{L,r}} P'}{[P]^{\mathbf{J}}_{l} \xrightarrow{L!\tilde{v}[l,r]} [\![[P']^{\mathbf{J}}_{l}]\!]_{\Delta}} \qquad (\text{Rcv})\ \frac{P \xrightarrow{\tilde{v}} P'}{[P]^{\mathbf{J}}_{l} \xrightarrow{?\tilde{v}@l} [\![[P']^{\mathbf{J}}_{l}]\!]_{\Delta}}$$

$$(\text{Bcast})\ \frac{M \xrightarrow{L!\tilde{v}[l,r]} [\![M']\!]_{\Delta} \quad N \xrightarrow{?\tilde{v}@l'} [\![N']\!]_{\Delta}}{M|N \xrightarrow{L!\tilde{v}[l,r]} [\![M'|N']\!]_{\Delta}} \quad d(l,l') \le r$$

$$(\text{Obs})\ \frac{M \xrightarrow{L!\tilde{v}[l,r]} [\![M']\!]_{\Delta}}{M \xrightarrow{R!\tilde{v}@K} [\![M']\!]_{\Delta}} \quad R \subseteq \{l' \in Loc:\ d(l,l') \le r\} \quad K = R \cap L,\ K \neq \emptyset$$

$$(\text{Lose})\ \frac{M \xrightarrow{L!\tilde{v}[l,r]} [\![M']\!]_{\Delta}}{M \xrightarrow{\tau} [\![M']\!]_{\Delta}} \qquad (\text{Move})\ \frac{}{[P]^{\mathbf{J}}_{l} \xrightarrow{\tau} [\![[P]^{\mathbf{J}}_{l}]\!]_{\mu^{\mathbf{J}}_{l}}} \qquad (\text{Par})\ \frac{M \xrightarrow{\gamma} [\![M']\!]_{\theta}}{M|N \xrightarrow{\gamma} [\![M'|N]\!]_{\theta}}$$

Table 4: LTS rules for Networks

M. We denote by *LSched* the set of schedulers for the LTS semantics. A labelled *execution* e of a network M driven by a scheduler F is a finite (or infinite) sequence of steps: $M \xrightarrow{\alpha_1}_{p_1\theta_1} M_1 \xrightarrow{\alpha_2}_{p_2\theta_2} M_2 \dots \xrightarrow{\alpha_k}_{p_k\theta_k} M_k$. By abuse of notation, we define $Exec^F_M$, $last(e)$, e^j and $e\uparrow$ as for unlabeled executions.

Since we are interested in weak observational equivalences, that abstract over τ-actions, we introduce the notion of *weak action* as follows: $\Longrightarrow$ is the transitive and reflexive closure of $\xrightarrow{\tau}$; $\stackrel{\alpha}{\Longrightarrow}$ denotes $\Longrightarrow\xrightarrow{\alpha}\Longrightarrow$ $\forall \alpha \neq \tau$. We write $\stackrel{\hat{\alpha}}{\Longrightarrow}$ for the weak action $\stackrel{\alpha}{\Longrightarrow}$ if $\alpha \neq \tau$, and $\Longrightarrow$ otherwise.

We denote by $Exec^F_M(\stackrel{\alpha}{\Longrightarrow}, H)$ the set of all executions that, starting from M, according to the scheduler F, lead to a network in the set H by performing $\stackrel{\alpha}{\Longrightarrow}$. We define the probability of reaching a network in H from M by performing $\stackrel{\alpha}{\Longrightarrow}$, according to a scheduler F as $Prob^F_M(\stackrel{\alpha}{\Longrightarrow}, H) = Prob^F_M(Exec^F_M(\stackrel{\alpha}{\Longrightarrow}, H))$.

For $\mathcal{F} \subseteq Sched$, we denote by $\hat{\mathcal{F}}_{\mathcal{C}} \subseteq LSched$ its *contextual superset* for the LTS semantics (see [4]).

Definition 3. *Let M and N be two networks. An equivalence relation $\mathcal{R}$ over networks is a* probabilistic labelled bisimulation *relative to a set $\mathcal{F}$ of schedulers, if $M\mathcal{R}N$ implies: for all schedulers $F \in \hat{\mathcal{F}}_{\mathcal{C}}$ there exists a scheduler $F' \in \hat{\mathcal{F}}_{\mathcal{C}}$ such that for all α and for all classes $\mathcal{C} \in \mathcal{N}/\mathcal{R}$:*
- if $\alpha \neq ?\tilde{v}@l$ then $Prob^F_M(\xrightarrow{\alpha}, \mathcal{C}) = Prob^{F'}_N(\stackrel{\hat{\alpha}}{\Longrightarrow} \mathcal{C})$;
- if $\alpha = ?\tilde{v}@l$ then either $Prob^F_M(\xrightarrow{\alpha}, \mathcal{C}) = Prob^{F'}_N(\stackrel{\alpha}{\Longrightarrow}, \mathcal{C})$ or $Prob^F_M(\xrightarrow{\alpha}, \mathcal{C}) = Prob^{F'}_N(\Longrightarrow, \mathcal{C})$.

Probabilistic labelled bisimilarity *relative to $\mathcal{F}$, written $\approx^{\mathcal{F}}_p$, is the largest probabilistic labelled bisimulation relative to $\mathcal{F}$ over networks.*

Probabilistic labelled bisimilarity is a characterization of our probabilistic observational congruence [4].

Theorem 1. *$M \cong^{\mathcal{F}}_p N$ if and only if $M \approx^{\mathcal{F}}_p N$.*

3. MEASURING ENERGY CONSUMPTION

In this section, based on the LTS semantics, we define a preorder over networks which allows us to study the performance, in terms of energy consumption, of different networks, but exhibiting the same (or similar) connectivity behaviour. For this purpose we associate an energy cost with labelled transitions as follows. For a transmission with radius r, let

$$\mathtt{En}(r) = \mathtt{En}_{elec} \times \mathtt{packet\_len} + \ \mathtt{En}_{ampl} \times \mathtt{packet\_len} \times r^2$$

where $\mathtt{En}_{elec}$ (nJ/b) is the energy dissipated to run the transmitter circuit, while $\mathtt{En}_{ampl}$ $(pJ/b/m^2)$ is the radio amplifier energy (see [11]). We define

$$\mathbf{Cost}(M, N) = \begin{cases} \mathtt{En}(r) & \text{if } M \xrightarrow{L!\tilde{v}[l,r]}_{\Delta} N \\ & \text{for some } L, \tilde{v}, l \text{ and } r \\ 0 & \text{otherwise} \end{cases}$$

For an execution $e = M_0 \xrightarrow{\alpha_1}_{\theta_1} M_1 \xrightarrow{\alpha_2}_{\theta_2} M_2 \dots \xrightarrow{\alpha_k}_{\theta_k} M_k$,

$$\mathbf{Cost}(e) = \sum\nolimits_{i=1}^{k} \mathbf{Cost}(M_{i-1}, M_i).$$

Let H be a set of networks; we denote by $Paths^F_M(H)$ the set of all executions from M ending in H and driven by F which are not prefix of any other execution ending in H. More formally, $Paths^F_M(H) = \{e \in Exec^F_M(H) \mid last(e) \in H$ and $\forall e'$ such that $e <_{prefix} e'$, $e' \notin Paths^F_M(H)\}$.

Now, we are ready to define the average cost of reaching a set of networks H from the initial network M according to the scheduler F.

Definition 4. *The average cost of reaching a set of networks H from an initial network M according to the scheduler F is*

$$\mathbf{Cost}^F_M(H) = \frac{\sum_{e \in Paths^F_M(H)} \mathbf{Cost}(e) \times P^F_M(e)}{\sum_{e \in Paths^F_M(H)} P^F_M(e)}.$$

The average cost is computed by weighting the cost of each execution by its probability according to F and normalized by the overall probability of reaching H.

The following definition provides an efficient method to perform both qualitative and quantitative analyses.

Definition 5. *Let $\mathcal{H}$ be a countable set of sets of networks and let $\mathcal{F} \subseteq LSched$ a set of schedulers. We write*

$$N \sqsubseteq^{\mathcal{F}}_{\mathcal{H}} M,$$

if $N \approx^{\mathcal{F}}_p M$ and, for all schedulers $F \in LSched$ and for all $H \in \mathcal{H}$, there exists a scheduler $F' \in LSched$ such that $\mathbf{Cost}^{F'}_N(H) \le \mathbf{Cost}^F_M(H)$.

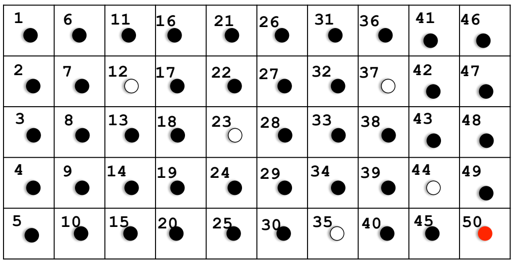

Figure 1: Topology of the Static Network (SN)

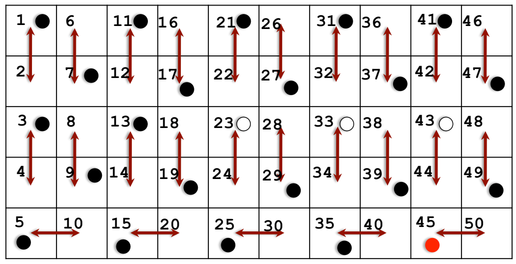

Figure 2: Topology of the Mobile Network (MN)

4. STUDYING GOSSIP PROTOCOLS

Gossip protocols are a family of communication protocols inspired by the way that gossiping disseminates information in social networks. A gossip protocol is a variant of the flooding algorithm, where each node forwards a message with some probability to reduce the overhead of the routing protocols. Gossiping-based routing protocols are commonly used in large-scale networks (see, e.g., [3, 10, 7]) to reduce the number of retransmissions and the energy cost.

In this section we show that our framework is suitable for providing an integrated automatic analysis of the gossip strategy in terms of both connectivity maintenance and energy consumption. In particular, we assume that, when a node receives a message, it forwards it with a fixed probability `psend` and discards it with probability $1 - \texttt{psend}$. Common values for `psend` range from 0.6 to 0.8: it is shown that, in practical scenarios, these values provide a reduction of more than 30% of the forwarding transmissions without deteriorating the network connectivity [7].

Sensor networks are usually characterised by a large number of small devices, densely distributed in the network area, sensing the environment and forwarding data. Here we consider two different network configurations on a rectangular area of 50×100m. We assume omnidirectional antenna and a fixed transmission power for each sensor node, which covers circular areas with a radius of 10m. In the following, we denote by $[P_i]^{\mathbf{J}}_l$ the sensor node i located at l, executing the process P_i and moving according to the transition matrix $\mathbf{J}$.

We study the behaviour of the networks by varying the value of the parameter `psend`.

The first network we consider consists of 50 static nodes, evenly distributed within the network area (see Figure 1). Node mobility is characterised by the identity matrix $\mathbf{I}$.

In our tests, we consider a fixed receiver $[P_{50}]^{\mathbf{I}}_{50}$, while the sender node's location varies in the set $\{12, 23, 35, 37, 44\}$, in order to study how the connectivity behaviour of the network changes, depending on the distance between the sender node and the receiver. The network is expressed by the term:

$$M_j \stackrel{\text{def}}{=} \sum\nolimits_{i=1}^{50} [P_i]^{\mathbf{I}}_i,$$

with $j \in \{12, 23, 35, 37, 44\}$, and
$P_i \stackrel{\text{def}}{=} (x_i).\langle x_i \rangle_{\{50\},10}.P_i, \;\; \forall i \notin \{j, 50\}$,
$P_j \stackrel{\text{def}}{=} \langle x_j \rangle_{\{50\},10}.P_j$,
$P_{50} \stackrel{\text{def}}{=} (x_{50}).P_{50}$,
modelling the communication between $[P_j]^{\mathbf{I}}_j$ and $[P_{50}]^{\mathbf{I}}_{50}$.
The second configuration consists of 25 mobile sensor nodes, again evenly distributed within the network area. Each sensor node can move between two adjacent locations, modelling the instability caused by, e.g., environmental conditions (see Figure 2). The probability distribution associated with node mobility can be captured by the transition matrix $\mathbf{J}$ such that: $\mathbf{J}_{l(l+5)} = \mathbf{J}_{(l+5)l} = \varepsilon \; \forall l \in \{5, 15, 25, 35, 45\}$, and $\mathbf{J}_{l(l+1)} = \mathbf{J}_{(l+1)l} = \varepsilon$ for all the other odd locations in the network area, and $\mathbf{J}_{ll} = 1 - \varepsilon$ for all the locations, with $0 < \varepsilon < 1$. Notice that the choice of ε and the definition of the scheduler allow us to model the relative speed between movements and transmissions. Henceforth we assume that $\varepsilon = 0.8$. This network is expressed by the term:

$$N_h \equiv \sum\nolimits_{i=1}^{25} [P_i]^{\mathbf{J}}_{(2i-1)},$$

with $h \in \{12, 17, 22\}$, where
$P_i \equiv (x_i).\langle x_i \rangle_{\{45,50\},10}.P_i, \;\; \forall i \notin \{h, 25\}$,
$P_h \equiv \langle x_h \rangle_{\{45,50\},10}.P_h$ and
$P_{25} \equiv (x_{25}).P_{25}$,
modelling the communication between $[P_h]^{\mathbf{J}}_{(2h-1)}$ and $[P_{25}]^{\mathbf{J}}_z$, where $z \in \{45, 50\}$ is the set of locations where we expect to find P_{25}.

We model several different gossip strategies by varying the value of `psend` in the interval $[0.6 - 1.0]$. In particular, for each value of `psend` we assume a set $\mathcal{F}_{\texttt{psend}}$ of schedulers such that, at each step, the probability for each node to perform a synchronisation or a movement is the same. Moreover, we do not consider message loss due to link failure or other environmental causes: a message can be lost only when a node discards it, consistently with the protocol.

The analysis is performed using the PRISM model checker [8] (see the Appendix for details). The first step of our methodology consists in translating the process-algebraic definition of our networks into the language supported by PRISM. This can be achieved in a purely algorithmic way. In general, the exact analysis of real WSNs' models is infeasible due to the explosion of the state space of the model. For this reason, we choose to perform an exact analysis to study problems of equivalence or performance in case of small components, e.g., to replace a network's node with a functionally equivalent one that has better performance in terms of throughput or energy consumption. Conversely, when studying the overall properties of wide WSNs we apply *approximate model checking* (also known as *statistical model checking*), that relies on a Monte Carlo simulation of the underlying DTMC. As a consequence, PRISM will compute estimates of the desired indices rather than exact results, whose precision is controlled by means of confidence interval specifications (absolute width and confidence). This approach is suitable for most practical purposes. When simulation is adopted, the estimates are obtained by sampling, i.e., generating a large number of random paths through the

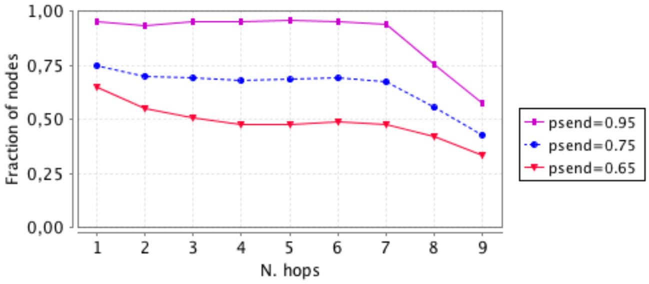

Figure 3: (SN) Fraction of nodes reached by a transmission

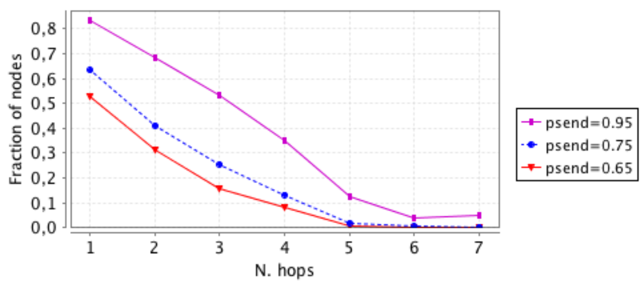

Figure 4: (MN) Fraction of nodes reached by a transmission

process underlying the model, hence avoiding the generation of whole DTMC.

In our case studies we assume that the sender node keeps retransmitting the same packet until the destination node receives it. The outcomes of this study allow us to determine the expected number of retransmissions of the same packet that are needed to reach the intended recipient or, in more detail, the number of retransmissions needed in order to reach the destination with a probability higher than a given threshold. Our goal is the comparison between the different network configurations, according to the definition of the energy-aware preorder introduced in Section 3. The following proposition is important for the termination condition of the simulations. It states that, by varying the sender location, the packet eventually reaches the intended recipient.

PROPOSITION 1.

(i) $\forall \mathcal{F}_{\texttt{psend}}$, $\texttt{psend} \in [0.6{-}1.0]$ *and* $\forall j_1, j_2 \in \{12, 23, 35, 37, 44\}$

$$M_{j_1} \approx_p^{\mathcal{F}_{\texttt{psend}}} M_{j_2}.$$

(ii) $\forall \mathcal{F}_{\texttt{psend}}$, $\texttt{psend} \in [0.6 - 1.0]$ *and* $\forall h_1, h_2 \in \{12, 17, 22\}$

$$N_{h_1} \approx_p^{\mathcal{F}_{\texttt{psend}}} N_{h_2}.$$

Proof. The proof can be formally done by observing that the sender keeps retransmitting the packet and that there is a non-null probability for a packet to reach the destination given any WSN configuration. Nevertheless, we can use the PRISM model checker in order to automatically verify the bisimulation among the different networks within a certain confidence. Usually, since this platform supports several temporal logics, this can be done by constructing characteristic formulas for bisimulation. Since in our case the movements and the forwarding transmissions are silent actions, while the only observable action is the transmission of the packet to the location 50 (resp. $\{45, 50\}$ for the mobile network), and our bisimulation does not take into account silent actions, we can simply verify that the probability for $[P_{50}]^{\mathbf{I}}_{50}$ ($[P_{25}]^{\mathbf{J}}_{45}$) to receive the message is always 1 (with the specified confidence).

We use the PCTL (Probabilistic Computation Tree Logic) **P** operator (for *reachability properties*). In particular, we verify $\mathbf{P}{=}?$ [**F** `goal`], i.e., what is the probability to eventually perform a transmission at location 50 ($\{45, 50\}$). **goal** is the formula indicating that P_{50} (P_{25}) has correctly received the message, and **F** means that the goal state will be eventually reached in a finite number of steps. For each network the probability to eventually reach the successful state turns out to be 1, where the confidence interval width is 0.01 based on 95% confidence level. □

Once we proved that the networks we are considering have the same connectivity, we are ready to compare their energy costs, by changing the value of **psend** and the distance among the sender and the receiver. Using the PRISM model checker, we exploit the possibility of defining reward measures to compute the energy cost function defined in Section 3. Assuming that the energy spent for each transmission is fixed and that all the nodes have the same physical characteristics, we simply count the number of transmissions rather than summing up their energy cost.

The cost function is expressed in terms of a PCTL formula in the PRISM property specification language, augmented with *rewards* (or costs), which are real-valued quantities associated with states and/or transitions (see the Appendix for more details). Specifically, we verify the formula $\mathbf{R} = ?$ [**F** `goal`], which expresses the cumulative expected energy cost to complete the communication.

Validation of the simulator. We have validated our simulations with those proposed in [7]. Specifically, the same kind of estimates shown in [7] can be computed by simulating our models in PRISM. For the static and mobile networks described above, we show the estimates of the fractions of nodes that are expected to be reached by a transmission in Figure 3 and 4, respectively.

Simulation of static networks. The estimates for the static network are shown in Figure 5. The simulations have been performed with an average of 10000 experiments, and a maximum confidence interval width of 1% of the estimated measure based on 95% of confidence. The plots show how the distance between sender and receiver critically influences the energy performance of the algorithm. For a distance larger then 30m we have a monotonic decreasing plot showing that, for large distances, the gossip protocol can cause energy to be wasted. Using the standard flooding strategy (**psend** = 1.0) all the cases converge to 49, because each node will forward the message exactly one time.

We can verify that there exists a preorder among different network configurations within the confidence of the simulation.

PROPOSITION 2.

$$\forall \textbf{psend} \in \{0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 1.0\}$$

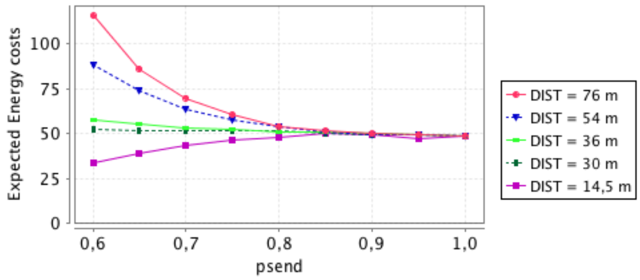

Figure 5: (SN) Expected energy cost

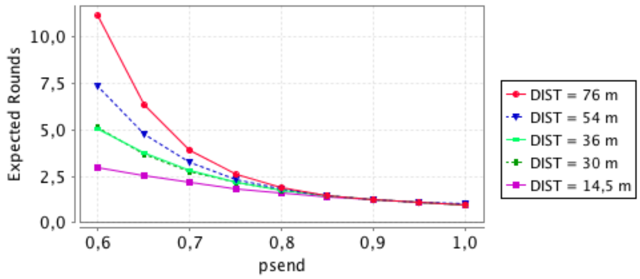

Figure 6: (SN) Expected number of transmissions for a successful communication

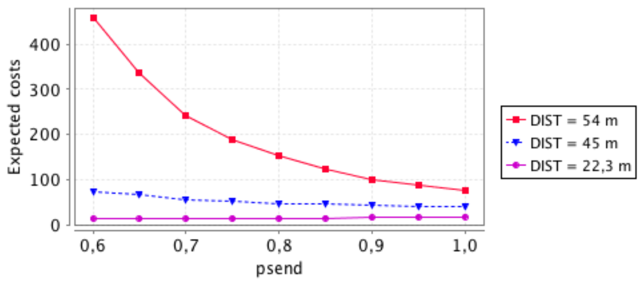

Figure 7: (MN) Expected energy cost

and $\forall j_1 < j_2 \in \{12, 23, 35, 37, 44\}$*:*

$$M_{j_1} \sqsubseteq_{\mathcal{H}}^{\mathcal{F}_{\mathbf{psend}}} M_{j_2}$$

where $\mathcal{H}$ is the set of network configurations where the communication has been successfully completed.

Figure 6 shows the expected number of retransmissions that the sender node must perform before the communication is successfully completed. Notice that, the smaller the value of **psend**, the higher is the probability that the message is lost during the path, forcing a new transmission (for the sake of simplicity we do not model the acknowledgements, but assume that the sender node will wait for an acknowledgement until a timeout occurs, then it will transmit again); hence, even if a small value of **psend** reduces the forwarding explosion, it may increase the number of replications.

Simulation of networks with mobility. Figure 7 shows the estimates of the expected energy cost for a successful transmission in the WSN with mobility.

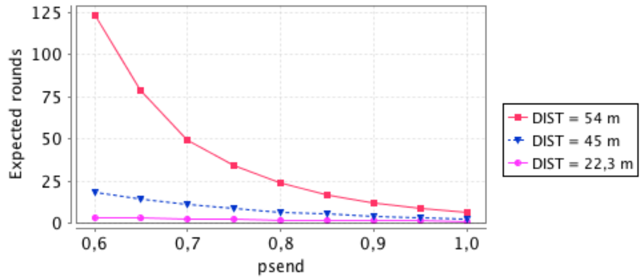

Figure 8: (MN) Expected number of transmissions for a successful communication

The mobility of the nodes critically increases the size of the state space, hence the obtained results have wider confidence intervals (ranging from 5% to 10% of the measure) than those obtained with the static network simulation, based on 95% of confidence. However, the results are very similar to the previous case: for distances larger than 25m the gossip protocol causes a very high energy waste. Also, in this case we can verify that there exists a preorder among different network configurations:

Proposition 3.

$$\forall \mathbf{psend} \in \{0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 1.0\}$$

and $\forall h_1 < h_2 \in \{12, 17, 22\}$*:*

$$N_{h_1} \sqsubseteq_{\mathcal{H}}^{\mathcal{F}_{\mathbf{psend}}} N_{h_2}$$

where $\mathcal{H}$ is the set of network configurations where the communications has been successfully completed.

Figure 8 shows the average number of retransmissions that the sender must perform before the communication has successfully completed. It is worth noticing that, with respect to the static network, in this case the distance between the sender and the receiver has a greater impact on the protocol performances.

5. RELATED WORK AND CONCLUSIONS

A large amount of research on sensor networks has been reported in the last decade. From the energy consumption viewpoint, several papers address the problem of studying the energy consumption for specific communication protocols. For instance, in [18] the authors define a Markov reward process (see, e.g., [14]) modelling some protocols for point to point reliable transmissions. A steady-state quantitative analysis is then obtained, and from this the average performance indices are computed. In [1], Bernardo et al. present a methodology to predict the impact of power management techniques on system functionality and performance. In [16], the authors define a set of metrics to analyse the energy consumption which are then estimated through simulation, and show how some modifications in the protocols can improve their efficiency. In [7], gossip protocols running on WSNs are studied, but the authors develop an ad hoc simulator to estimate their performance. By contrast, in our setting, a general purpose tool, e.g., PRISM, can be used since the performance indices or properties to be evaluated (or estimated) can be formally specified in a rigorous logic. Moreover, with respect to all the above mentioned contributions, the model we propose here aims at providing a common framework for automatically performing both qualitative and quantitative analyses. Indeed, PRISM can be used for different purposes. The energy preorder defined in Section 3 can be efficiently decided for small network components using model checking methods, and hence one may decide to replace a node with another which is behaviourally equivalent but less energy consuming; conversely, when the complexity of the process underlying the model makes exact analyses infeasible, approximate (or statistical) model checking can be employed. This corresponds to the well-known Monte Carlo simulation; using the temporal logic implemented in the tool, one can verify a property within a certain level of confidence (e.g., the network equipped with a certain protocol is connected with a confidence of 99.9%). To the best of our knowledge, such an integrated qualitative and quantitative approach supported by the same tool represents a novelty in the study of WSNs.

6. REFERENCES

[1] A. Acquaviva, A. Aldini, M. Bernardo, A. Bogliolo, E. Bontà, and E. Lattanzi. A methodology based on formal methods for predicting the impact of dynamic power management. In *Formal Methods for Mobile Computing*, volume 3465 of *lncs*, pages 51–58. Springer Berlin / Heidelberg, 2005.

[2] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless Sensor Networks: A Survey. *Computer Networks*, 38(4):393 – 422, 2002.

[3] A. Dimakis, A. Sarwate, , and M. Wainwright. Geographic Gossip: Efficient Aggregation for Sensor Networks. In *Proc. of the 5th international conference on Information processing in sensor networks*, pages 69–76. ACM, 2006.

[4] L. Gallina. *Formal Models for Qualitative and Quantitative Analysis of Mobile Ad Hoc and Sensor Networks*. PhD thesis, DAIS, University Ca' Foscari of Venice, 2013.

[5] L. Gallina, S. Hamadou, A. Marin, and S. Rossi. A probabilistic energy-aware model for mobile ad-hoc networks. In *Proc. of the 18th International Conference on Analytical and Stochastic Modelling Techniques and Applications (ASMTA'11)*, volume 6751 of *LNCS*, pages 316–330. Springer-Verlag, 2011.

[6] J. Goubault-Larrecq, C. Palamidessi, and A. Troina. A probabilistic applied pi-calculus. In *Proc. of the 5th Asian Symposium on Programming Languages and Systems (APLAS'07)*, volume 4807/2009 of *LNCS*, pages 175–190. Springer-Verlag, 2007.

[7] Z. Haas, J. Halpern, and L. Li. Gossip-based Ad Hoc Routing. *IEEE/ACM Trans. Netw.*, 14(3):479–491, 2006.

[8] A. Hinton, M. Kwiatkowska, G. Norman, and D. Parker. Prism: A tool for automatic verification of probabilistic systems. In *Proc. of the 12th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'06)*, volume 3920 of *LNCS*, pages 441–444. Springer-Verlag, 2006.

[9] M. Z. Kwiatkowska, G. Norman, and D. Parker. Prism 4.0: Verification of probabilistic real-time systems. In G. Gopalakrishnan and S. Qadeer, editors, *CAV*, volume 6806 of *Lecture Notes in Computer Science*, pages 585–591. Springer, 2011.

[10] P. Kyasanur, R. Choudhury, and I. Gupta. Smart Gossip: An Adaptive Gossip-based Broadcasting Service for Sensor Networks. In *Mobile Adhoc and Sensor Systems (MASS), 2006 IEEE International Conference on*, pages 91 –100, 2006.

[11] T. V. Madhav and N. Sarma. Maximizing Network Lifetime through Varying Transmission Radii with Energy Efficient Cluster Routing Algorithm in Wireless Sensor Networks. *International Journal of Information and Electronics Engineering*, 2(2):205–209, 2012.

[12] M. Merro. An observational theory for mobile ad hoc networks. *Information and Computation*, 207(2):194–208, 2009.

[13] R. Milner and D. Sangiorgi. Barbed bisimulation. In *Proc. of International Colloquium on Automata, Languages and Programming (ICALP'92)*, volume 623 of *LNCS*, pages 685–695. Springer-Verlag, 1992.

[14] S. M. Ross. *Stochastic Processes*. John Wiley & Sons, 2nd edition, 1996.

[15] R. Segala and N. Lynch. Probabilistic simulations for probabilistic processes. In *Proc. of the 5th International Conference on Concurrency Theory (CONCUR'94)*, volume 836 of *LNCS*, pages 481–496. Springer-Verlag, 1994.

[16] S. Singh, M. Woo, and C. Raghavendra. Power-aware routing in mobile ad hoc networks. In *Proc. of the 4th annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'98)*, pages 181–190. ACM, 1998.

[17] G. Zhou, C. Huang, T. Yan, T. He, J. A. Stankovic, and T. F. Abdelzaher. Mmsn: Multi-frequency media access control for wireless sensor networks. In *In IEEE INFOCOM*, page 7, 2006.

[18] M. Zorzi and R. Rao. Error control and energy consumption in wireless communications channels. In *Proc. of the 34th Annual Allerton Conference on Communication, Control and Computing*, pages 825 – 834, 1996.

APPENDIX

PRISM [8] is a probabilistic model checker, a tool for formal modelling and analysis of systems that exhibit random or probabilistic behaviour. It supports a wide range of probabilistic models, such as *Markov decision processes* (MDPs), *discrete-time Markov chains* (DTMCs) and *continuous-time Markov chains* (CTMCs), and these models are described using the PRISM language. Moreover, PRISM provides a specification language to specify rewards and quantitative properties, and it supports the automated analysis of these properties with respect to the probabilistic models.

Modelling the network. We translate process terms into DTMCs in order to automatically evaluate the performance, in terms of energy, of a sensor network using a basic gossip protocol for the information exchange among the sensor nodes. In the PRISM language we define a constant **psend**, ranging from 0.6 to 1, and we study the different models that are generated by the different values assigned to **psend**.

Each node $[P_i]^{\mathbf{J}}_{l_i}$ is translated into a corresponding module `Pi`, which is associated with two variables:

```
module P8
steps8 : [0 .. 2] init 2;
l8 : [15 .. 20] init 15;

[] (l8 = 15) → 0.8 : (l8′ = 20) + 0.8 : (l8′ = 15);
[] (l8 = 20) → 0.8 : (l8′ = 15) + 0.8 : (l8′ = 20);

//beginning of a new round
[round] no_one_sending → (steps8′ = 2);
//beginning of a new round

//transmission
//[c8] (steps8 = 1) & (conn38 | conn78 | conn810 | conn813) → (steps8′ = 0);

//receives
[c3] (steps8 = 2)& (conn38) → psend : (steps8′ = 1) + (1 − psend) : (steps8′ = 0);
[c3] (steps8! = 2) |!(conn38) → (steps8′ = steps8)
[c7] (steps8 = 2)&(conn78) → psend : (steps8′ = 1) + (1 − psend) : (steps8′ = 0);
[c7] (steps8! = 2) |!(conn78) → (steps8′ = steps8)
[c10] (steps8 = 2)&(conn810) → psend : (steps8′ = 1) + (1 − psend) : (steps8′ = 0);
[c10] (steps8! = 2) |!(conn810) → (steps8′ = steps8)
[c13] (steps8 = 2)&(conn813) → psend : (steps8′ = 1) + (1 − psend) : (steps8′ = 0);
[c13] (steps8! = 2) |!(conn813) → (steps8′ = steps8)

endmodule
```

Table 5: The PRISM module for a node

- $\mathtt{steps_i}$: controls the sequentiality of the process executed by the sensor node. In particular, $\mathtt{steps_i} = 2$ means that the node is ready to receive, $\mathtt{steps_i} = 1$ means that the node is ready to transmit, and $\mathtt{steps_i} = 0$ means that the node has completed a transmission.
- $\mathtt{l_i}$: is the variable containing the actual location of the sensor node.

Table 5 shows the implementation of a single node. Each transition of the PRISM model corresponds to a transition of the labelled transition system underlying the PEBUM network term: the unlabelled commands model the possible node's movements, while the labelled commands are used to model synchronisations. Each action tagged with [ci] represents a transmission from the source sensor node P_i.

Each node starts its transmission ($\mathtt{steps8} = 1$) only if there is at least one of the neighbours ready to receive it ($\mathtt{conn38} \mid \mathtt{conn78} \mid \mathtt{conn810} \mid \mathtt{conn813}$). This is a standard strategy for gossip protocols. However, since the neighbour nodes will forward the message only with a certain probability, the presence of at least one receiver inside the sender node's area does not ensure the completeness of the communication. A node receives the packet only if it is inside the transmission area of the sender node, and forwards it with probability psend:

$$[\mathtt{c13}]\ (\mathtt{steps8} = 2) \& (\mathtt{conn813}) \rightarrow \mathtt{psend} : (\mathtt{steps8}' = 1) + (1 - \mathtt{psend}) : (\mathtt{steps8}' = 0);$$

Property specification. PRISM provides a *property specification language* which supports various temporal logics as well as extensions for rewards (or costs). Rewards can be associated to states or to transitions. The *cumulative rewards* in the PRISM language are expressed as:

```
rewards ‘‘reward_name’’
    [transition] condition : value;
    condition : value;
endrewards
```

In particular, [transition] condition : value associates a reward value to each transition tagged with [transition] when the condition condition is true, while condition : value associates a reward value to each state where the condition condition is true.

In the PRISM specification language, $\mathbf{R} = ?[\mathbf{F}\ \mathtt{goal\_state}]$ denotes the cumulative expected reward to eventually reach goal_state.

We use PRISM to find the value of **psend** which minimises the costs of communications.