Automatic Energy-aware Performance Analysis of Mobile Ad-hoc Networks

Lucia Gallina[1], Tingting Han[2], Marta Kwiatkowska[2], Andrea Marin[1], Sabina Rossi[1] and Alvise Spanò[1]

[1] Università Ca' Foscari Venezia, Italy
[2] University of Oxford, United Kingdom

***Abstract*—We present a framework to automatically evaluate the performance of Mobile Ad-hoc Networks (MANETs) in terms of different kinds of metrics, such as throughput and energy consumption. For this purpose, we use a probabilistic process calculus to model MANETs; then we translate process terms into Markov Decision Processes (MDPs) and use the probabilistic model checker PRISM to automatically evaluate the network performance. We present a case study consisting of a network which uses flooding for communicating, and we analyse how time and energy costs vary when pursuing different power control strategies.**

Keywords*—*process algebra; MANETS; model checking; network throughput; energy consumption

I. Introduction

Mobile ad-hoc networks (MANETs) are systems of mobile devices communicating with each other through wireless links without a pre-established networking infrastructure. The devices may rely on exhaustible power sources, such as batteries, and hence have strict requirements regarding their energy consumption. For these reasons, when dealing with networks of this type, communication protocols have to face the problem of providing full connectivity between network devices while maintaining good performance, both in terms of throughput and of energy conservation (see, e.g., [19], [20]).

In this paper we present a framework to automatically evaluate the performance of MANETs in terms of different kinds of metrics. In our framework we use the Probabilistic Energy-aware Broadcast Unicast and Multicast (PEBUM) calculus, introduced in [7], to model MANETs. This is a probabilistic process calculus particularly aimed at providing performance analysis in terms of metrics such as energy conservation and throughput. A reduction semantics has been defined for the PEBUM calculus, together with an equivalence relation parameterised by a restricted set of schedulers. The equivalence between two networks allows us to express connectivity properties and it can be proved by using a coinductive proof methodology based on a notion of probabilistic bisimulation.

PRISM [12] is a tool for modelling and analysing systems that exhibit a probabilistic behaviour. It supports, among others, the modelling of Markov Decision Processes (MDPs), where nondeterministic and probabilistic aspects coexist.

In this paper we exploit the PRISM tool to perform automated, quantitative verification and analysis of wireless networks for a range of performance metrics. Specifically, we develop a parser to translate a PEBUM process term, representing a network, into an MDP model expressed in the PRISM language. Moreover, we formulate the metrics for computing the time and energy cost of a network transmission in terms of reward structures.

We demonstrate the effectiveness of our framework on a case study consisting of a MANET which uses flooding to forward messages. We analyse the performance of the communication between two static devices by varying the transmission power of the intermediate, possibly mobile, nodes forwarding the packets within the network. Then we study how the time and energy costs vary when pursuing different power control strategies for the intermediate communications.

Related Work. Probabilistic models are nowadays widely used in the design and verification of complex systems. Bernardo et al. introduced EMPA$_{gr}$ [2], an extended Markovian process algebra including probabilities, priority and exponentially distributed durations. One advantage of this calculus is the possibility of modelling both exponentially timed and immediate actions, whose selection is controlled by a priority level associated with them. Palamidessi et al. [9] defined an extension of the applied pi-calculus with nondeterministic and probabilistic choice operators. Among the stochastic process algebras for performance evaluation, one of the most successful is the Performance Evaluation Process Algebra (PEPA), introduced by Hillston in [10], which is used to model systems composed of concurrently active components which cooperate and share work. For both the probabilistic applied pi-calculus and PEPA model checking algorithms built upon the PRISM tool have been developped (see [16], [4]).

Although model checking implementations have already been proposed for other process calculi, our work is specifically aimed at studying the performance of mobile ad-hoc and sensor networks. Indeed, we exploit the main features of the PEBUM calculus which allows one to represent variable transmission ranges and probabilistic movements of nodes within the network area.

Regarding energy usage, several papers address the problem of studying the energy consumption of a specific communication protocol for wireless networks. For instance, in [20] the authors define a Markov reward process (see, e.g., [17]) modelling some protocols for pairwise communications. 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 [19] 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. With respect to the above mentioned works, the model we propose here aims at providing a common framework for both qualitative and quantitative analyses.

Plan of the paper. Section II introduces the PEBUM calculus with its syntax, reduction and observation semantics. Section III presents the parser that translates a PEBUM process term into a MDP model in the PRISM language. Section IV presents our case study and demonstrates the automatic performance analysis with respect to different power control strategies. Section V concludes the paper.

II. The Calculus

We present the PEBUM calculus [7] which models mobile adhoc networks as a collection of nodes, running in parallel and using channels to broadcast messages. It supports multicast communications as well as power control.

Syntax. We use c for *channels*, n for *nodes*, l and k for *locations*, r for *transmission radii*; x for *variables*. *Closed values* contain nodes, locations, transmission radii and any basic value (booleans, integers, ...). *Open values* include variables only. We use u and v for closed values, w for open values, and $\tilde{v}$, $\tilde{w}$ for tuples of values. We write *Loc* for the set of locations.

Networks are defined as the parallel composition of distinct nodes. We write $n[P]_l$ for a node n, located at the location l and executing the process P. We denote by $M_1|M_2$ the parallel composition of two networks and by $\prod_{i\in I} M_i$ the parallel composition of the networks M_i, for $i \in I$. Nodes cannot be created or destroyed. A process P is as follows:

- $\mathbf{0}$ denotes the inactive process;
- $c(\tilde{x}).P$ can receive a tuple $\tilde{w}$ of closed values via channel c and continue as $P\{\tilde{w}/\tilde{x}\}$, i.e., as P with $\tilde{w}$ substituted for $\tilde{x}$. In the process $c(\tilde{x}).P$, the variables in $\tilde{x}$ are bound in P;
- $\bar{c}_{L,r}\langle\tilde{w}\rangle.P$ can send a tuple of closed values $\tilde{w}$ via channel c and continue as P. The tag L is used to maintain the set of observation locations such that $L = Loc$ represents a broadcast transmission, while a finite set of locations L denotes a multicast communication (unicast if L is a singleton). The tag r represents the power of the transmission. Syntactically, tags L and r may be variables, but they must be instantiated when the output prefix is ready to fire;
- $[w_1 = w_2]P, Q$ behaves as P if $w_1 = w_2$, as Q otherwise.
- $A\langle\tilde{w}\rangle$ denotes a process defined via a (possibly recursive) definition $A(\tilde{x}) \stackrel{\text{def}}{=} P$, with $|\tilde{x}| = |\tilde{w}|$ where $\tilde{x}$ contains all channels and variables that appear free in P.

A node can be either *mobile* or *static*. Each node n is associated with a pair $< r_n, \mathbf{J}^n >$, where r_n is a non negative real number denoting the maximum transmission radius that n can use to transmit, while $\mathbf{J}^n$ is the transition probability matrix of a discrete time Markov chain, where $\mathbf{J}^n_{lk}$ is the probability that the node n located at l, after executing a movement, will be located at k. Hence, $\sum_{k\in Loc} \mathbf{J}^n_{lk} = 1$ for all $l \in Loc$. Notice that, for static nodes, $\mathbf{J}^n_{ll} = 1$ for $l \in Loc$ and $\mathbf{J}^n_{lk} = 0$ for $l \neq k$.

Probability distributions. We associate probability distributions with located nodes. More formally, we denote by μ^n_l the probability distribution associated with node n located at l, that is, a function over *Loc* such that $\mu^n_l(k) = \mathbf{J}^n_{lk}$, for $k \in Loc$. Moreover, if $M = \prod_{i\in I} n_i[P_i]_{l_i}$ is a network, then for all $k \in I$, $[\![M]\!]_{\mu^{n_k}_{l_k}}$ denotes the probability distribution over the set of networks induced by $\mu^{n_k}_{l_k}$, i.e., for all networks M':

$$[\![M]\!]_{\mu^{n_k}_{l_k}}(M') = \begin{cases} \mu^{n_k}_{l_k}(l'_k) & \text{if } M' = \prod_{i\in I} n_i[P_i]_{l'_i} \\ & \text{for } l'_i = l_i \forall i \neq k \\ 0 & \text{otherwise} \end{cases}$$

Note that $[\![M]\!]_{\mu^{n_k}_{l_k}}(M')$ is the probability that network M evolves to M' due to the movement of the node n_k located at l_k. We say that M' is in the support of $[\![M]\!]_{\mu^{n_k}_{l_k}}$ if $[\![M]\!]_{\mu^{n_k}_{l_k}}(M') \neq 0$. We write $[\![M]\!]_\Delta$ for the Dirac distribution on network M, namely the probability distribution defined as: $[\![M]\!]_\Delta(M) = 1$ and $[\![M]\!]_\Delta(M') = 0$ for all $M' \neq M$. We let θ range over $\{\mu^n_l \mid n \text{ is a node and } l \in Loc\} \cup \{\Delta\}$.

Reduction Semantics. The dynamics of the calculus is specified by the *probabilistic reduction relation* over networks ($\rightarrow$), described in Table I. As usual in process calculi, it relies on an auxiliary relation, called structural congruence ($\equiv$), such that for instance $M|N \equiv N|M$, $(M|N)|M' \equiv M|(N|M')$ and $M|\mathbf{0} \equiv M$. The probabilistic reduction relation takes the form $M \rightarrow [\![M']\!]_\theta$, which describes a transition that leaves from a network M and leads to a probability distribution $[\![M']\!]_\theta$.

Rule (R-Bcast) models the transmission of a tuple of messages $\tilde{v}$ to the receivers within the locations in L, using channel c and transmission radius r. Indeed, nodes communicate using radio frequencies that enable only message broadcasting. However, a node may decide to communicate with a specific set of nodes: depending on L, it can be a broadcast, multicast or unicast communication, as specified before. The tag r indicates the transmission radius required for that communication and may vary due to the energy consumption strategy adopted by the communication protocol. Notice that a transmission is *nonblocking*: it proceeds even if there are no nodes listening. The messages transmitted may be received only by those nodes which lie in the transmission area of the sender. Function $d(\cdot,\cdot)$ returns the distance between two locations.

Rule (R-Move) states that a node n located at l and executing a moving action will reach a location with the probability described by the distribution μ^n_l that depends on the Markov chain $\mathbf{J}^n$ associated with n. Movements are atomic actions: while moving, a node cannot do anything else.

Rules (R-Par) and (R-Struct) are standard.

Observation Semantics. The central actions of our calculus are transmission and reception of messages. However, only the transmission of messages can be observed. An observer cannot

$$\text{(R-Bcast)}\ \frac{}{n[\bar{c}_{L,r}\langle\tilde{v}\rangle.P]_l \mid \prod_{i\in I} n_i[c(\tilde{x}_i).P_i]_{l_i} \rightarrow [\![n[P]_l \mid \prod_{i\in I} n_i[P_i\{\tilde{v}_i/\tilde{x}_i\}]_{l_i}]\!]_\Delta}$$

where $0 < r \leq r_n$, $\forall i \in I. d(l, l_i) \leq r$ and $|\tilde{x}_i| = |\tilde{v}_i|$

$$\text{(R-Move)}\ \frac{}{n[P]_l \rightarrow [\![n[P]_l]\!]_{\mu_l^n}} \quad \text{(R-Par)}\ \frac{M \rightarrow [\![M']\!]_\theta}{M|N \rightarrow [\![M'|N]\!]_\theta} \quad \text{(R-Struct)}\ \frac{M \rightarrow [\![M']\!]_\theta \ \ M' \equiv N'}{M \rightarrow [\![N']\!]_\theta}$$

TABLE I: Reduction Semantics

be sure whether the recipient actually receives a given value. Instead, if a node receives a message, then surely someone must have sent it. Following [15], we use the term *barb* as a synonym of observable. However, our calculus contains both nondeterministic and probabilistic aspects, where the nondeterministic choices are among the possible probability distributions that a process may follow and arise from the possibility for nodes to perform movements.

Given a network M, we write $M \rightarrow_\theta N$ if $M \rightarrow [\![M']\!]_\theta$ and N is in the support of $[\![M']\!]_\theta$. An execution for M is a (possibly infinite) sequence of steps $M \rightarrow_{\theta_1} M_1 \rightarrow_{\theta_2} M_2 \ldots$ We write $last(e)$ for the final state of a finite execution e, e^j for the prefix execution $M \rightarrow_{\theta_1} M_1 \ldots \rightarrow_{\theta_j} M_j$ of length j of the execution $e = M \rightarrow_{\theta_1} M_1 \ldots \rightarrow_{\theta_j} M_j \rightarrow_{\theta_{j+1}} M_{j+1} \ldots$ We write $M \rightarrow^* M'$ if there exists a finite execution e for M such that $last(e) = M'$. We denote by $behave(M) = \{[\![M']\!]_\theta \mid M \rightarrow [\![M']\!]_\theta\}$ the set of all the distributions M can reach in a single reduction step.

In order to resolve the nondeterminism in a network execution, we use *schedulers* [18] to choose one distribution from $behave(M)$. Given a finite execution e, a *scheduler* is a total function F assigning a distribution $[\![N]\!]_\theta \in behave(last(e))$ to e. Given a network M and a scheduler F, we define the set of executions starting from M and driven by F as:

$$\begin{aligned} Exec_M^F = \{ & e = M \rightarrow_{\theta_1} M_1 \rightarrow_{\theta_2} M_2 \cdots \mid \forall j > 0, \\ & M_{j-1} \rightarrow [\![M'_j]\!]_{\theta_j},\ [\![M'_j]\!]_{\theta_j} = F(e^{j-1}) \\ & \text{and } [\![M'_j]\!]_{\theta_j}(M_j) > 0 \}. \end{aligned}$$

Formally, given a finite execution $e = M \rightarrow_{\theta_1} M_1 \ldots \rightarrow_{\theta_k} M_k$ starting from a network M and driven by a scheduler F we define the probability for M, following F, to perform e, as:

$$P_M^F(e) = [\![M'_1]\!]_{\theta_1}(M_1) \cdot \ldots \cdot [\![M'_k]\!]_{\theta_k}(M_k)$$

where $\forall 0 < j \leq k,\ [\![M'_j]\!]_{\theta_j} = F(e^{j-1})$.

Given a scheduler F, $\sigma Field_M^F$ is the smallest σ-field on $Exec_M^F$ that contains the basic cylinders $e \uparrow$, where $e \in Exec_M^F$. The probability measure $Prob_M^F$ is the unique measure on $\sigma Field_M^F$ such that $Prob_M^F(e\uparrow) = P_M^F(e)$. Given a measurable set of networks H, we denote by $Exec_M^F(H)$ the set of executions starting from M and reaching a state in H. Formally, $Exec_M^F(H) = \{e \in Exec_M^F \mid last(e^j) \in H$ 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 $Prob_M^F(H) = Prob_M^F(Exec_M^F(H))$.

The notion of barb introduced below denotes an *observable transmission* with a certain probability according to a fixed scheduler. In our definition, a transmission is observable only if at least one recipient in the set of the observation locations is able to receive the message.

Definition 2.1: [Barb] Let $M \equiv n[\bar{c}_{L,r}\langle\tilde{v}\rangle.P]_l | M'$. We say that M has a barb on a channel c at locations $K (\neq \emptyset)$, denoted $M \downarrow_{c@K}$, if $\exists K \subseteq L$ such that $d(l, k) \leqslant r, \forall k \in K$.

Definition 2.2: [Probabilistic Barb] We say that a network M has a *probabilistic barb* with probability p on a channel c at locations K according to the scheduler F, written $M \Downarrow_p^F c@K$, if $Prob_M^F(H) = p$ where $H = \{M' \mid M \rightarrow^* M',\ M' \downarrow_{c@K}\}$.

Intuitively, for a given network M and scheduler F, if $M \Downarrow_p^F c@K$ then there is a positive probability that M, driven by F, performs a transmission on channel c and at least one of the recipients in the observation locations is able to correctly listen to it. Hereafter, we introduce a probabilistic observational congruence, in the style of [9], which is a relation relative to a set of schedulers $\mathcal{F}$ (we call it $\mathcal{F}$-relation) and is defined as the largest $\mathcal{F}$-relation as follows. Let $\mathcal{R}^\mathcal{F}$ be a relation over networks and consider the following properties:
Barb preservation. $\mathcal{R}^\mathcal{F}$ is *barb preserving* if $M \mathcal{R}^\mathcal{F} N$ and $M \Downarrow_p^F c$ for some $F \in \mathcal{F}$ implies that there exists $F' \in \mathcal{F}$ such that $N \Downarrow_p^{F'} c$.
Reduction closure. $\mathcal{R}^\mathcal{F}$ is *reduction closed* if $M \mathcal{R}^\mathcal{F} N$ implies that, for all $F \in \mathcal{F}$, there exists $F' \in \mathcal{F}$ such that, for all classes $\mathcal{C} \in \mathcal{N}/\mathcal{R}^\mathcal{F}$, $Prob_M^F(\mathcal{C}) = Prob_N^{F'}(\mathcal{C})$.
Contextuality. $\mathcal{R}^\mathcal{F}$ is *contextual* if $M \mathcal{R}^\mathcal{F} N$ implies that, for every context $\mathcal{C}[\cdot]$, it holds that $\mathcal{C}[M]\, \mathcal{R}^\mathcal{F}\, \mathcal{C}[N]$, where a context is a network term with a hole $[\cdot]$ defined by the grammar: $\mathcal{C}[\cdot] ::= [\cdot] \mid [\cdot]|M \mid M|[\cdot]$.

Definition 2.3: [Probabilistic observational congruence] *Probabilistic observational congruence relative a set $\mathcal{F}$ of schedulers*, written $\cong_p^\mathcal{F}$, is the largest symmetric $\mathcal{F}$-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 behaviour (communications) relative to the corresponding sets of observable locations and the fixed set of schedulers $\mathcal{F}$. In [8] we have developed a bisimulation-based proof technique for $\cong_p^\mathcal{F}$. It provides an efficient method to check whether two networks are related by $\cong_p^\mathcal{F}$.

Cost preorder. In [6] we introduced a preorder, which allows us to compare networks that have the same probabilistic observable behaviours, but different performance in terms of metrics such as energy consumption and time delay. The preorder is based on the definition of a *cost function*, denoted $\mathbf{Cost}$, which returns for given a network M, a set H of goal networks and a scheduler F, the cost for M to reach H following F. Our cost function can be used to define an *efficiency preorder* as follows.

Definition 2.4: [Efficiency preorder] Let $\mathcal{H}$ be a countable set of sets of networks and $\mathcal{F}$ be a set of schedulers. We say that N is *as least as efficient as* M relative to $\mathcal{F}$ and $\mathcal{H}$, denoted

$$N \sqsubseteq_{\langle \mathcal{F},\mathcal{H}\rangle} M,$$

if $N \cong_p^{\mathcal{F}} M$ and, for all schedulers $F \in \mathcal{F}$ and for all $H \in \mathcal{H}$, there exists $F' \in \mathcal{F}$ such that $\mathbf{Cost}_N^{F'}(H) \leq \mathbf{Cost}_M^{F}(H)$.

III. A PARSER FOR PEBUM CALCULUS

In this section we introduce a translator, implemented in F# [5] with FsYacc and FsLex [13], which compiles PEBUM networks into the PRISM language. Our tool parses an input source file with the description of a network according to the syntax introduced in Section II and translates it into the corresponding MDP for PRISM.

Input of the parser. The input of the parser is a text file containing:

- the PEBUM network (see Section II), written in the following grammar:

Networks	**Processes**
$\mathbf{0}$	$\mathbf{0}$
$\mid M_1 \mid M_2$	$\mid (\tilde{x}) \leftarrow c; P$
$\mid n@l\{P\}$	$\mid \tilde{v} \to c@L/r; P$
	$\mid \texttt{if}\, w_1 = w_2 \,\texttt{then}\, P \,\texttt{else}\, Q$
	$\mid \texttt{rec}\, P$

- a transition probability matrix $\mathbf{J}^n$ associated with each node n as illustrated below, where $\mathtt{L_1}$, ..., $\mathtt{L_k}$ are the locations in the network and $\mathtt{p_{i,j}}$ is the probability for the node n to move from the location $\mathtt{L_i}$ to $\mathtt{L_j}$.

$$\mathbf{J}^n = \begin{array}{cccc} \mathtt{L_1} & \mathtt{L_2} & \dots & \mathtt{L_k} \\ \mathtt{p_{1,1}} & \mathtt{p_{1,2}} & \cdots & \mathtt{p_{1,k}} \\ & \dots & & \\ \mathtt{p_{k,1}} & \mathtt{p_{k,2}} & \cdots & \mathtt{p_{k,k}} \end{array}$$

- a triangular matrix $\mathbf{Dist}$, describing the distances between each pair of network locations. The matrix is triangular because we assume the distance to be symmetric.

$$\mathbf{Dist} = \begin{array}{cccc} \mathtt{L_1} & \mathtt{L_2} & \dots & \mathtt{L_k} \\ \mathtt{d_{1,1}} & \mathtt{d_{1,2}} & \dots & \mathtt{d_{1,k}} \\ & & \dots & \\ & & & \mathtt{d_{k,k}} \end{array}$$

The parser processes the input text to collect all the information we need to capture the behaviour of a network expressed by the reduction semantic rules (see Table I). The key is to generate a list of records containing information about the possible synchronisations the network can perform (see (R-Bcast) rule in Table I). This list can be generated by analysing all the network processes, with respect to the probability distribution of each node's mobility (expressed by $\mathbf{J}^n$) and the topology of the network (expressed by $\mathbf{Dist}$).

```
mdp

const L1=1;
const L2=2;

const msg1=1;
const msg2=2;

module n1

loc_n1 : [L1 .. L2] init L1;

x : [0 .. 2] init 0;

proc_n1 : [0 .. 2] init 0;

[] (loc_n1 = L1) -> 1:(loc_n1' = L1) + 0:(loc_n1' = L2);
[] (loc_n1 = L2) -> 0:(loc_n1' = L1) + 1:(loc_n1' = L2);

[c_n2_msg1_l1_l1l2] (proc_n1 = 0) & (( loc_n1 = L1) ) -> (proc_n1' = 1);
[c_msg1_lost] (proc_n1 = 0) -> (proc_n1' = 1);

[c_n1_msg2_l1_l1] (proc_n1 = 1) & (( loc_n1 = L2) )
                  -> (x' = msg2) & (proc_n1' = 2);

endmodule
```

Fig. 1: PRISM module for n_1

Output of the parser. The output of the parser is an MDP written in the PRISM language [12]. As an example, Fig. 1 shows the translation of the simple PEBUM network

$$n_1[\bar{c}_{l_1,r}\langle msg1\rangle.c(x)]_{l_1} \mid n_2[c(y).\bar{c}_{l_1,r}\langle msg2\rangle]_{l_2}.$$

Constants are used to represent the set Loc of network locations as well as the set of messages sent by the network nodes during the execution. Given a network $M \equiv \prod_{i\in I} n_i[P_i]_{l_i}$, each network node $n_i[P_i]_{l_i}$ corresponds to an MDP module, with name n_i. The variable loc_{n_i} refers to the current location of n_i; the variable $proc_{n_i}$ represents the current process status, which is used to control the order of process execution (since processes are deterministic and sequential). If the process executed by the node contains bound variables, they will be defined in the MDP too (see variable x in the example in Fig. 1).

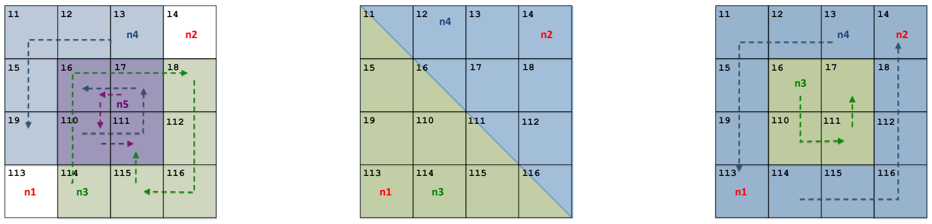
Modules may execute commands (or actions) corresponding to the reduction rules (R-Bcast) and (R-Move) in Table I:

- (R-Bcast) $\quad n[\bar{c}_{L,r}\langle\tilde{v}\rangle.P]_l \mid \prod_{i\in I} n_i[c(\tilde{x}_i).P_i]_{l_i}$
  $\to [\![n[P]_l \mid \prod_{i\in I} n_i[P_i\{\tilde{v}_i/\tilde{x}_i\}]_{l_i}]\!]_\Delta$

Since in PRISM synchronisations are modelled by labelling commands with *actions*, we associate the following *action* label with each message transmission:

$$[\mathtt{c\_n_1 \ldots n_{h1}\_v_1 \ldots v_{h2}\_K_1\_ \ldots \_K_{h3}\_R_1 \ldots \_R_{h4}}]$$

where $\{\mathtt{n_1}, ..., \mathtt{n_{h1}}\}$ is the set of the nodes n_i $(i \in I)$ receiving the message; $(\mathtt{v_1}, ..., \mathtt{v_{h2}})$ is the tuple $\tilde{v}$ of messages; $\{\mathtt{R_1}, ..., \mathtt{R_{h4}}\}$ is the set of locations whose distance from l is



(a) Scheme 1 (b) Scheme 2 (c) Scheme 3

Fig. 2: The different network schemes

less than the radius; $\{\mathtt{K}_1, ..., \mathtt{K}_{h3}\}$ is the intersection between the set of the observation locations and the set $\{\mathtt{R}_1, ..., \mathtt{R}_{h4}\}$.

The module of sender n contains the following command in PRISM (for $L_k = l$):

$$[\mathtt{c\_n_1 ... n_{h1}\_v_1 ... v_{h2}\_K_1\_ ... \_K_{h3}\_R_1 ... \_R_{h4}}](\mathtt{proc\_n{=}counter}) \\ \&(\mathtt{loc\_n{=}L_k}) \rightarrow (\mathtt{proc\_n'{=}counter{+}1});$$

The module of a receiver node $n_i (i \in I)$ contains the following command in PRISM:

$$[\mathtt{c\_n_1 ... n_{h1}\_v_1 ... v_{h2}\_K_1\_ ... \_K_{h3}\_R_1 ... \_R_{h4}}] \\ (\mathtt{proc\_n} = \mathtt{counter})\&((\mathtt{loc\_n_i} = \mathtt{R_1})| \cdots |(\mathtt{loc\_n_i} = \mathtt{R_k})) \\ \rightarrow (\mathtt{proc\_n_i'} = \mathtt{counter} + 1)\&(\mathtt{x_i} = \mathtt{msg});$$

If I is the empty set, then it means that no nodes receive the transmission, and the message is lost. We add a command with a loss-message action to the sender's module:

$$[\mathtt{c\_v_1 ... v_{h2}\_lost}](\mathtt{proc\_n} = \mathtt{counter})\&(\mathtt{loc\_n} = \mathtt{L_k}) \\ \rightarrow (\mathtt{proc\_n'} = \mathtt{counter} + 1);$$

- (R-Move) $\quad n[P]_l \rightarrow [\![n[P]_l]\!]_{\mu_l^n}$

In PRISM the mobility of nodes is expressed using a different command for each row of matrix $\mathbf{J}^n$ associated with each node. In particular, each i-th row of matrix $\mathbf{J}^n$ corresponds to the command (for $\mathtt{L}_1, ..., \mathtt{L}_j \in Loc$):

$$[]\,(\mathtt{loc\_n{=}L_i}) \rightarrow \mathtt{p_{i,1}}{:}(\mathtt{loc\_n'{=}L_1}) + \cdots + \mathtt{p_{i,j}}(\mathtt{loc\_n'{=}L_j});$$

Correctness of the translation. Let $\mathcal{N}$ be the set of PEBUM networks, and let $\mathbf{T}$ be the function mapping networks into corresponding MDPs. Let $M \in \mathcal{N}$ be a network, and let $\mathbf{T}(M)$ be its corresponding MDP model.

The translation captures all the behaviours expressed by the reduction semantics, i.e., there is a one-to-one correspondence between the reduction rules of M and the possible actions in $\mathbf{T}(M)$. This correspondence can be proved by induction on the reduction rules. For (R-Bcast) and (R-Move), the proof follows directly from the definition of the corresponding labelled and unlabelled commands. For (R-Par), the parallel composition is modelled by the fact that for each broadcast reduction the resulting MDP always chooses one of the possible synchronisations among the network nodes (or the message loss) in a nondeterministic way. Finally, the proof of the correspondence for (R-Struct) follows by the structural congruence closure of the reduction barbed congruence.

IV. A CASE STUDY

We present a case study to show the effectiveness of translating our process algebra into an MDP in order to automatise the performance analysis for a range of metrics using PRISM.

The network. We consider two static nodes (n_1 and n_2) communicating in a 60×60 metres network area (see Fig. 2). The distances between cells are determined by considering the centre of each cell and calculating the euclidean distance between each pair of centres (each cell is 15×15 metres). Since the two static nodes are too distant to reach each other (compared to their radius 20), the communication is possible only if there are other intermediate nodes inside the network forwarding the messages between n_1 and n_2. We consider a simple flooding algorithm in which mobile nodes forward the messages in lazy mode, i.e., they store the data to be sent until there is another node ready to receive it. This scenario is typical for gossip protocols.

In the following, we analyse different situations, each with some intermediate nodes that may move in a fixed area of the network (that is formally captured by the probability transition matrix associated with each intermediate node).

Scheme 1: The first network we consider has three mobile nodes able to forward messages inside the network. As shown in Fig. 2.(a), n_3 can move within the bottom-right square of nine locations, n_4 moves within the top-left square of nine locations, while n_5 can only move within the four central cells of the network area.

Scheme 2: The second scheme we consider is a network where, as shown in Fig. 2.(b), one of the two intermediate nodes can move in the triangular area around the position of n_1, while the other one covers the triangular area around n_2.

Scheme 3: In the last scheme, shown Fig. 2.(c), one node can only move inside the central area of the network (4 cells), while the second one can only move in the 12 border cells.

We study the performance of these networks by varying the transmission power of the intermediate nodes, in order to find the best power strategy which optimises the performance in terms of energy consumption and throughput.

The three networks only differ in the transition probability matrices associated with each mobile node, while their behaviour is expressed in the PEBUM calculus as follows where $\mathit{RAD} \in \{20, 30, 40, 50\}$:

$M_{\mathit{RAD}} \equiv \prod_{i=1}^{j} n_i[P_i]_{l_i}$, $j = \{4, 5\}$ where:

- $P_1 \equiv \bar{c}_{l4,20}\langle \texttt{msg\_for\_n2}\rangle.P_1'$
  $P_1' \equiv c(x_1).[x_1 = \texttt{ack\_to\_n1}]\bar{d}_{l13,20}\langle \texttt{ok}\rangle, P_1'$;
- $P_2 \equiv c(x_2).[x_2 = \texttt{msg\_to\_n2}]\bar{c}_{l13,20}\langle \texttt{ack\_to\_n1}\rangle, P_2$
- $P_k \equiv c(x_k).\bar{c}_{\emptyset,\mathit{RAD}}\langle x_k\rangle.P_k, \quad k \in \{3, 4, 5\}$

with $j = 5$ for *scheme 1*, and $j = 4$ for *schemes 2, 3*.

First we prove that, by varying the value of RAD, the probabilistic observation behaviour of M_{RAD} does not change.

Theorem 4.1: For $R1, R2 \in \{20, 30, 40, 50\}$,

$$M_{R1} \cong_p^{\mathcal{F}_\mathcal{M}} M_{R2}.$$

Proof sketch: M_{R1} and M_{R2} have the same probabilistic mobility behaviour (i.e., the probability distribution associated with each node, in both M_{R1} and M_{R2}, does not change). They only differ in the output actions performed by the intermediate nodes, since different transmission radii cause different synchronisations, depending on the locations of the receivers. First, notice that all the actions performed by the intermediate nodes are silent (their outputs have no observable locations). If $R2 \geq R1$, then each silent action of M_{R1} can be performed by M_{R2} too. If $R2 \leq R1$, then the output action can be mimicked by a sequence of movements and outputs by M_{R2}. Since the process modelling the node movements among the locations is irreducible, then each state (location) is reachable from any other in finite time with probability 1, and hence we can find a sequence of silent actions performed by M_{R2} mimicking M_{R1} output with probability 1. ■

For each scheme we build a PRISM MDP, assigning to the variable RAD the values in the set $\{20, 30, 40, 50\}$, in order to investigate how different transmission powers influence the performance of the network.

We use PRISM to verify the following property which computes the maximal probability (among all schedulers) of eventually reaching the goal states:

```
Pmax=? [F goal_state]
```

From the experiments, we obtain probability 1 for all the different schemes, and for all possible radii. In the next paragraph, time and energy cost information is added to the MDP. We investigate how the cost will be affected by varying the network topology and the transmission radii of the nodes.

Time and Energy Costs. We study the energy spent in the transmissions assuming that the cost for the control packets is negligible. We consider the restricted set of schedulers $\mathcal{F} \in \mathit{Sched}$ such that nodes always perform synchronisations where possible and they transmit a message only if at least one node is listening to the channel (i.e., messages are not lost).

We assume quasi-linear cost functions of the form

$$\sum_k w_k \cdot \Phi_k(\_) \tag{1}$$

that are sums of arbitrary functions $\Phi_k(\_)$, each of which maps one metric (e.g., energy cost and throughput) to some common measure. In general, we give a different weight $w_k \geq 0$ to each $\Phi_k(\_)$ to reflect its importance. We assume that $\sum_k w_k = 1$.

In the following, we first introduce the energy cost $\Phi_\mathbf{E}$, then the time cost $\Phi_\mathbf{T}$ and finally the global cost function.

Energy Cost. We associate an energy cost with the probabilistic network reductions, as follows:

$$\mathbf{E}(M, N) = \begin{cases} (\texttt{En}_{elec} \times \texttt{packet\_len} + \\ \ \texttt{En}_{ampl} \times \texttt{packet\_len} \times r^2) \\ \qquad \text{if } M \to_\theta N, M \equiv n[\bar{c}_{L,r}\langle\tilde{v}\rangle.P]_l \mid M', \\ \qquad N \equiv n[P]_l \mid N' \text{ for some } c, L, \tilde{v}, l \\ \\ 0 \quad \text{otherwise.} \end{cases}$$

where $\texttt{En}_{elec}$(nJ/b) is the energy dissipated to run the transmitter circuit, $\texttt{En}_{ampl}$ $(pJ/b/m^2)$ is the Radio amplifier energy. They are constants given a priori (see [14]).

Time Cost. We associate time cost with the probabilistic network reductions as follows:

$$\mathbf{T}(M, N) = \begin{cases} (\texttt{packet\_len/bandwith}) \\ \quad \text{if } M \to_\triangle N, M \equiv n[\bar{c}_{L,r}\langle\tilde{v}\rangle.P]_l \mid M' \\ \quad \text{and } N \equiv n[P]_l \mid N' \text{ for some } c, L, v, l \\ \\ (\texttt{cell\_size/velocity}) \\ \quad \text{if } M \to_{\mu_l^n} N, M \equiv n[P]_l \mid M' \\ \quad \text{and } N \equiv n[P]_k \mid M' \end{cases}$$

where $\texttt{cell\_size}$ (m) is the size of a cell, $\texttt{velocity}$ (m/s) is the velocity of the devices and $\texttt{bandwith}$ (MB) is the channel bandwidth of the network. Again they are constants.

Now let $\mathcal{X} \in \{\mathbf{E}, \mathbf{T}\}$ be one of the two metrics above, and $e = M_0 \to_{\theta_1} M_1 \to_{\theta_2} M_2 \cdots \to_{\theta_k} M_k$ be an execution. We define

$$\mathcal{X}(e) = \textstyle\sum_{i=1}^{k} \mathcal{X}(M_{i-1}, M_i).$$

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

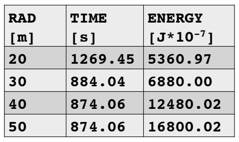
Definition 4.2: Let H be a set of networks and $\mathcal{X} \in \{\mathbf{E}, \mathbf{T}\}$. The average energy and time cost of reaching H from M according to the scheduler F is

$$\mathcal{X}_M^F(H) = \frac{\sum_{e \in \mathit{Paths}_M^F(H)} \mathcal{X}(e) \times \mathit{Prob}_M^F(e)}{\sum_{e \in \mathit{Paths}_M^F(H)} \mathit{Prob}_M^F(e)}.$$

The average energy and time 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.

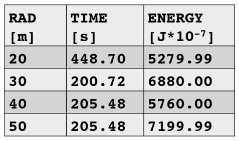
Cost Function. Now consider Eq. (1). Let $\Phi_E = \mathbf{E}$ and $\Phi_T = \mathbf{T}$, and w be the weight of the energy function. We define the cost of reaching a set H from the network M according to the scheduler F as:

$$\mathbf{Cost}_M^F(H) = \alpha\, \mathbf{E}_M^F(H) + \beta\, \mathbf{T}_M^F(H), \tag{2}$$



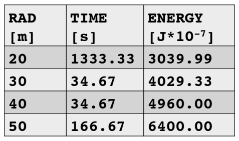
RAD [m]	TIME [s]	ENERGY [J*10^{-7}]
20	1269.45	5360.97
30	884.04	6880.00
40	874.06	12480.02
50	874.06	16800.02

(a) Scheme 1



RAD [m]	TIME [s]	ENERGY [J*10^{-7}]
20	448.70	5279.99
30	200.72	6880.00
40	205.48	5760.00
50	205.48	7199.99

(b) Scheme 2



RAD [m]	TIME [s]	ENERGY [J*10^{-7}]
20	1333.33	3039.99
30	34.67	4029.33
40	34.67	4960.00
50	166.67	6400.00

(c) Scheme 3

Fig. 3: The time-only and energy-only cost

where α and β are weighting coefficients expressed in s^{-1} and J^{-1} respectively, and henceforth we simply consider $\alpha = w$ and $\beta = 1 - w$, with $w \in [0, 1]$.

According to the energy and time costs we introduced above, in our PRISM MDP the constants are given as follows:

```
const double cell_size = 10.0;
const double bandwith = 1.0 ;
const double packet_len = 200.0;
const double velocity = 0.3 ;

//ENERGY FOR COMMUNICATION
//50 nJ/b*packet_len+100 pJ/bit/m^2*packet_len*r^2

const double bcast_pow_n1= 800+1.6*RAD1*RAD1;
const double bcast_pow_n2= 800+1.6*RAD2*RAD2;
const double bcast_pow_n3= 800+1.6*RAD4*RAD4;
const double bcast_pow_n4= 800+1.6*RAD3*RAD3;

const double move_power = 0;

const double move_time = cell_size/velocity;
const double bcast_time = packet_len/1000000 ;
```

where `bcast_pow_ni` and `bcast_time` are the energy and time spent for a single transmission performed by n_i, while `move_power` and `move_time` are the energy and time spent for each single movement step.

The cost can be calculated by defining a reward structure:

```
rewards "cost"
[]       true : (1-w)*move_time + w*(move_power);
[out_ni] true : (1-w)*bcast_time +w*bcast_pow_ni;
...
endrewards
```

Since for all the schemes introduced above we proved that, for each pair $R1, R2 \in \{20, 30, 40, 50\}$, $M_{R1} \cong_p^{\mathcal{F}_\mathcal{M}} M_{R2}$, we are ready to investigate if there exists, for each scheme, a preorder among M_{20}, M_{30}, M_{40} and M_{50}. This means that we have a way to choose the radius optimising the performance of the network in terms of time and energy cost.

We use the PRISM tool to automatically verify the second point of Def. 2.4, by calculating, for each radius in the set $\{20, 30, 40, 50\}$, the minimum cost among all schedulers under which the goal state is reached with probability 1:

```
R{"cost"}min=? [F goal_state]
```

Fig. 3 shows the time and energy costs we calculated for each scheme, while the general cost functions are specified in Fig. 4 (time cost is expressed in s while energy is expressed in $J \times 10^{-7}$). Here we summarize our results.

(i) Time cost ($w = 0$). For each scheme, a higher time cost is needed for $RAD = 20$ (see Fig. 3). This is due to the fact that a small radius requires more rounds of forwarding and more movements of the nodes, both of which cost more time.

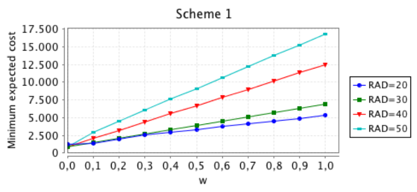
(ii) Energy cost ($w = 1$). In all of the three schemes in Fig. 3, the smallest radius always has the minimum energy cost. This is because, regardless of time, it is always possible that mobile nodes move closer to each other or move close to n_1 or n_2, and this reduces the energy cost. It is worth pointing out that, due to a scaling representation factor, Fig. 4 tends to hide the relative differences in the total weighted cost among the radii for $w = 0$.

(iii) For $0 < w < 1$ in general, we investigate for each scheme how the total cost is affected by choosing different power strategies (i.e., transmission radii). The results are shown in Fig. 4.

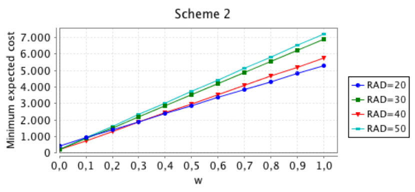
Scheme 1: It can be seen in Fig. 3 that radius 30 is better than 20 in terms of delays (about $\frac{884}{1269} = 70\%$ of time cost for radius 20), and radius 20 is better than 30-50 in terms of energy (about $\frac{5360}{6880} = 78\%$ of the energy cost for radius 30). So the best radius depends on the w chosen: roughly speaking, if $w \leq 0.6$, then radius 30 is better than 20; otherwise 20 is better. We notice that here large radii consistently deteriorate the energy performance of the network, without a real improvement in terms of transmission delays: this is due to the fact that there is a high density of nodes in the network area, which can cause unnecessary and expensive transmissions.

Scheme 2: In Scheme 2 (see Fig. 4.(b)), the smallest (largest) radius still seems to give the best (worst) cost. However, using radius 40 is much better than 30 due to the mobility behaviours associated with the intermediate nodes: with radius 30, n_3 is only able to reach n_1, while n_4 can only forward n_2 communications: this situation enlarges the minimum number of transmissions allowing n_1 and n_2 to complete their communications. Conversely, choosing a larger radius (40) enables n_3 (respectively n_4) to eventually reach n_2 (respectively n_1), drastically reducing the number of transmissions. Since the energy cost using radius 40 is not much higher than the cost spent using radius 20, while the time costs are halved, we can conclude that for the second scheme the best power management strategy for time-aware applications is the use radius 40.

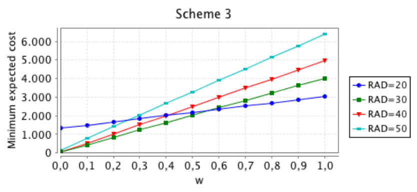
Scheme 3: Finally, let us consider the third scheme (see Fig. 4.(c)). With $w \leq 0.57$ the best solution is radius 30, while with $w > 0.57$ the best strategy is radius 20. Moreover, if we are interested only in reducing the time spent to communicate (see Fig. 3.(c)), radius 40 is the best choice, since the costs



(a) Scheme 1



(b) Scheme 2



(c) Scheme 3

Fig. 4: Cost results for each scheme

are consistently reduced with respect to the other radii, while radius 20 critically increases time cost. Generally speaking, the best solution is to use radius 30. We notice that 30 is the smallest distance that the inner area (4 cells) and the outer area (12 cells) can always directly reach each other, while radius 20 never allows n_3 to reach n_1 and n_2. For these reasons time cost consistently increases for radius 20, while with radius 30 is better (about $\frac{34.67}{1333.33} = 2.5\%$ of the time cost for radius 20). With respect to radii 40 and 50, radius 30 allows one to reduce the energy cost, without significantly deteriorating the time delays.

V. Conclusion and Future work

This paper presented an automatic tool, based on the PEBUM calculus [6], [7], for comparing different costs of networks having the same probabilistic observable behaviour. The automatic quantitative verification of a network has been implemented by model checking. In particular we provided a translation of the PEBUM models into PRISM MDP models [11], which allows one to automatically verify the performance of networks in terms of various metrics.

Recently, we proposed a new version of the PEBUM calculus [3], aimed at capturing the level of interference in mobile ad-hoc networks caused by collisions due to network congestion, as well as attacks by malicious nodes which intentionally make the channel busy. We plan to use our parser to translate this new process algebra in order to automatically verify the performance of a given network in terms of interference level.

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] M. Bernardo and M. Bravetti. Performance measure sensitive congruences for Markovian process algebras. *Theoretical Computer Science*, 290(1):117–160, 2003.

[3] M. Bugliesi, L. Gallina, S. Hamaodu, A. Marin, and S. Rossi. Interference-sensitive preorders for manets. In *Proc. 9th International Conference on Quantitative Evaluation of SysTems (QEST'12)*. IEEE, 2012.

[4] F. Ciocchetta, S. Gilmore, M. L. Guerriero, and J. Hillston. Integrated simulation and model-checking for the analysis of biochemical systems. In *Proc. 3rd International Workshop on Practical Applications of Stochastic Modelling (PASM'08)*, volume 232 of *ENTCS*, pages 17–38, 2008.

[5] F. DeRemer and T. Pennello. Efficient computation of lalr(1) look-ahead sets. *ACM Trans. Program. Lang. Syst.*, 4(4):615–649, October 1982.

[6] L. Gallina, S. Hamadou, A. Marin, and S. Rossi. A framework for throughput and energy efficiency in mobile ad hoc networks. In *Proc. of IFIP Wireless Days 2011*. IEEE Press, 2011.

[7] 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.

[8] L. Gallina, S. Hamadou, A. Marin, and S. Rossi. Connectivity and energy aware preorders for mobile ad hoc networks. Technical Report DAIS-2012-1, University Ca' Foscari Venice, 2012.

[9] 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.

[10] J. Hillston. *A Compositional Approach to Performance Modelling*. Cambridge University Press, 1996.

[11] 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.

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

[13] M.E. Lesk. Lex – a lexical analyzer generator. Computer Science Technical Report 39, Murray Hill, New Jersey: Bell Laboratories, 1975.

[14] T. Venu Madhav and N.V.S.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.

[15] 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.

[16] G. Norman, C. Palamidessi, D. Parker, and P. Wu. Model checking probabilistic and stochastic extensions of the π-calculus. *IEEE Transactions on Software Engineering*, 35(2):209–223, 2009.

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

[18] R. Segala and N.A. 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.

[19] S. Singh, M. Woo, and C.S. 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.

[20] M. Zorzi and R.R. Rao. Error control and energy consumption in communications for nomadic computing. *IEEE Transactions on Computers*, 46(3):279–289, 1997.