Noname manuscript No.
(will be inserted by the editor)

Connectivity and Energy-aware Preorders for Mobile Ad-Hoc Networks

Lucia Gallina · Andrea Marin · Sabina Rossi

Received: date / Accepted: date

Abstract We propose a probabilistic, energy-aware, broadcast calculus for the analysis of both connectivity and energy consumption of MANETs. The semantics of our calculus is expressed in terms of probabilistic automata driven by schedulers to resolve the nondeterministic choice among the probability distributions over target states. We first develop a probabilistic observational congruence together with a bisimulation-based proof technique. Then we define an energy-aware preorder semantics. The observational congruence allows us to verify whether two networks exhibit the same observable probabilistic behaviour in terms of connectivity, while the preorder makes it possible to evaluate the energy consumption of different, but behaviourally equivalent, networks. We show our calculus at work both by modelling the Location Aided Routing (LAR) protocol for large MANETs and by evaluating the energy cost of a Go-Back-N protocol with respect to a Stop-And-Wait in a network with mobility.

Keywords Manets, Process Algebras, Energy Consumption, Performance Evaluation

1 Introduction

Mobile ad-hoc networks (MANETs) are collections of mobile devices communicating with each other through wireless links without a pre-established networking infrastructure. Free node mobility is a main feature of such networks: each device in a MANET can move autonomously in any direction, and therefore its links to other devices may change frequently. These changes in the network topology can cause the nodes to continuously enter and exit each other transmission area and hence highly dynamic routing algorithms are needed to ensure the network connectivity. Moreover, mobile devices often have strict requirements on the energy consumption because their expected life-time usually depends on the energy stored in a battery or other exhaustible power sources. For these reasons, the communication protocols must face the problem of providing good connectivity among the network devices while maintaining good performances both in terms of throughput and energy conservation (see, e.g., [40,46,38]). For larger networks in which some of/all the nodes are aware of their relative or absolute geographical position, e.g., thanks to a Global Positioning System device (GPS), the routing protocols may exploit this information in order to improve the efficiency of packet delivery by controlling the flooding process (see, e.g., [22,43]).

Drawing on earlier work on the subject [11,14,25,29], in the present paper we introduce a calculus to provide a formal basis for the analysis of connectivity and the evaluation of energy consumption in MANETs.

The definition of a general formalism allowing for both qualitative (connectivity) and quantitative (power consumption and throughput) analysis is a challenging topic of research. Indeed, general purpose formalisms for concurrency (e.g., Petri nets) do not deal with the mobility of the devices in a natural way, and hence they do not allow for a modular and hierarchical description of mobile systems. In [5] we presented a calculus with non-atomic output and input actions to capture the presence of interferences caused by the simultaneous transmission of two (or more) nodes. The calculus

Lucia Gallina
E-mail: lgallina@dais.unive.it

Andrea Marin · Sabina Rossi
DAIS, Università Ca' Foscari Venezia, via Torino 155, 30172 Mestre Venezia, Italy
Tel.: +39 041 2348411 Fax: +39 041 2348419
E-mail: {marin,srossi}@dais.unive.it

of [5] is targeted at the evaluation of the level of interference in mobile ad hoc networks, while any quantitative assessment of energy consumption is considered. Here we present a calculus, named Probabilistic EBUM, for formally reasoning about Energy-aware Broadcast, Unicast and Multicast communications of mobile adhoc networks. This is an extension of the EBUM calculus presented in [14,12,13] where probability distributions are used to describe the movements of nodes. Like its predecessor [14,5], our calculus is built around nodes, representing the 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. Instead, wireless synchronizations are non-deterministic, and controlled by sequential processes inside the nodes. Our calculus allows us to model the ability of a node to broadcast a message to any other node within its physical transmission range, and to move in and out of the transmission range of other nodes in the network. Broadcast communications are limited to the transmission cell of the sender, while unicast and multicast communications are modelled by specifying, for each output action, the locations of the intended recipients of the message. The idea of using location-based destination is motivated by the need of efficiently modelling large networks with location-based routing, such as the ones presented in [22,43], and of comparing their efficiency with respect to standard routing algorithms based on flooding. Nevertheless, the routing based on the knowledge of the node's destination address (but not its physical location) can still be implemented in our calculus by specifying the intended recipients' addresses as part of the message content. This reflects the actual implementation of wireless protocols in which messages are broadcast and then filtered by the recipient devices according to the (MAC) address specified in the header of the packet. Another important feature of the Probabilistic EBUM calculus is the possibility for a node to control its transmission power. This is modelled by allowing nodes to modify the transmission radius of their communications through internal actions.

The Probabilistic EBUM calculus deals with both non-deterministic and probabilistic choices. Its semantics is inspired by Segala's probabilistic automata [39] driven by schedulers to resolve the nondeterministic choice among the probability distributions over target states. In this paper we define a probabilistic observational congruence in the style of [30] to equate networks exhibiting the same probabilistic connectivity behaviour. As in [13,12], and in contrast to [29], the notion of observability is associated with nodes listening at specific locations in the network, so as to allow a fine grained analysis of connectivity at different areas within a network. We give a coinductive characterisation of observational congruence based on a labelled transition semantics. This is a bisimulation-based proof technique in the form of a probabilistic labelled bisimilarity which is shown to coincide with the observational equivalence. We also introduce energy-aware preorders over networks to measure the relative energy cost of different, but behaviourally equivalent, networks. We show our framework at work on the analysis of two case-studies. The first one consists in modelling the Location Aided Routing (LAR) protocol [22]: we study how the performances of this protocol vary depending on the characteristics of the specific network, e.g., node density, topology changes and power capacity of the devices. In the second case-study we compare the performances, in terms of energy consumption, of an aggressive protocol for reliable communications (Go-Back-n) and a slower protocol (Stop&Wait).

This paper is an extended and improved version of [11]. The main novelties concern the extension of the calculus through the channel restriction operator (νc) over networks. From a semantic perspective, it simply plays the role of a CCS-style hiding operator, but it is useful to specialise the verification method to some specific class of contexts. Moreover, we define a new equivalence relation that is parametric to a restricted set of executions for a given network: our new definition of *probabilistic barbed congruence* allows us to study the performances of networks focusing the attention only on specific restricted behaviours, abstracting out all the executions that are unrealistic or that are simply non interesting for the aims of the analysis. We also define the labelled semantics which is proved to coincide with the probabilistic observational congruence. This provides the basis for powerful, both inductive and coinductive, proof techniques. Finally, the analysis of the LAR protocol using our Probabilistic E-BUM calculus is totally new.

Related work. Probabilistic models are nowadays widely used in the design and verification of complex systems. In the following we give an overview of the formal frameworks for mobile ad-hoc and sensor networks.

Song and Godskesen [41] propose a probabilistic broadcast calculus for mobile and wireless networks with unreliable connections. The peculiarity of this calculus is the introduction of a *probabilistic mobility function* to model the mobility of nodes. Recently, in [42] the same authors propose a new version of their calculus built upon a *stochastic mobility function* to model the stochastic changes of connectivity. As in our works [12, 11,14] broadcast actions are associated with the locations of the intended recipients of the message. However, differently from our calculus, in [42] any notion of transmission radius is introduced and any performance analysis is considered.

Palamidessi et al. in [17] define the Probabilistic Applied π-calculus: this is a probabilistic extension of Applied π-calculus [1], where both non-deterministic and probabilistic choices are modelled. The authors define both a static equivalence, and an obervational congruence based on the notion of probabilistic barb, which describes the probability, for a given system, to perform a certain observable action. As in our calculus, in order to solve the non-determinism, schedulers (also called polices, or adversaries) have been introduced. They are modelled as functions mapping states into probability distributions. Differently from our work, their semantic is not parameterized over restricted sets of schedulers.

Merro et al. introduce aTCWS (applied Timed Calculus for Wireless Systems) [28]: a timed broadcasting process calculus targeted at security analysis of wireless networks with fixed nodes communicating at the same transmission power and aver the same transmission frequency. The connectivity of the network is expressed by associating with each node a tag containing the list of all its neighbours. The timed model adopted by this calculus is known as the *fictitious clock* approach, and it is based on clock synchronization of nodes. A probabilistic version of TCWS has been introduced in [26]. The main feature of this calculus is the presence of a *simulation up to probability* which allows one to compare networks which exhibit the same behaviour up to a certain probability. The main limitations of such calculus are the absence of mobility and of multiple frequencies.

In [8] Hennessy and Cerone propose a calculus to model the high-level behaviour of Wireless Systems (i.e., MAC-layer protocols). This calculus is characterized by a two-level structure: on one hand, it models both probabilistic and non-deterministic processes behaviour, as well as communications through a fixed set of channels; on the other hand, the topology is expressed through an undirected graph where each edge represents the direct link between a pair of network nodes. Neither a notion of distance nor of transmission radius has been introduced. Furthermore, modelling communication links with an undirected graph presupposes that all nodes use the same fixed radius to communicate, an assumption that is not realistic for MANETs, which include different kinds of devices, with different physical structure and power resources.

De Nicola et al. introduce StoKlaim [9]: a stochastic process algebra, whose underlying processes are Continuous Time Markov Chains, allowing one to describe random phenomena regarding mobile wireless networks.

As far as performance evaluation is concerned, Hillston et al. introduce the process algebra PEPA [19] which has been designed for modelling systems composed of concurrently active components which co-operate and share resources. The authors also provide a tool, the PEPA Workbench [15], which allows a practical use of this process algebra in many applications concerning software architecture and communication protocols.

Bernardo et al. introduce EMPA$_{gr}$ [4], an extended Markovian process algebra including probabilities, priority and exponentially distributed durations. Its peculiarity is the possibility of modelling both exponentially timed and immediate actions, whose selection is controlled by a priority level associated with them.

Other frameworks for performance modelling based on Petri Nets and queueing networks fall short of accounting for node mobility while maintaining a good accuracy in specifying the protocol design [31,3].

As far as energy consumption is concerned, several papers address the problem of studying the energy consumption of a specific communication protocol for wireless networks. For instance, in [46] the authors define a Markov Reward process (see, e.g., [35]) modelling some protocols for pairwise node communications. A steady-state quantitative analysis is then derived and hence the average performance indices computed. In [2] Bernardo et al. present a methodology to predict the impact of the power management techniques on a system functionality and performance. In [40] the authors define a set of metrics on the energy consumption which are then estimated through simulation and show how some changes in the protocols can improve the 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.

Concerning the problem of routing in mobile ad-hoc networks, several different solutions have been proposed. Usually, routing protocols are classified in *proactive* and *reactive*. While proactive protocols continually exchange routing information about all the nodes, (see, e.g., DSDV [34] and WRP [32]), the reactive protocols update the routing table of each node only on-demand (see, e.g., the AODV [36], TORA [33] and DSR [20]). Although proactive routing reduces the latency in sending out packets, due to the continuous up-to-date of the routing tables, reactive routing are more efficient in terms of resource usage, since they update the route tables only on-demand. When dealing with mobile ad-hoc networks the most common strategy is to use hybrid protocols, where both the proactive and the reactive approach coexist in order to provide a good trade-off between latency and overhead.

Plan of the paper. Section 2 introduces the Probabilistic E-BUM calculus and its observational semantics. In Section 3 we present the LTS semantics and define a labelled bisimilarity which is proved to coincide with the observational congruence of the unlabeled semantics. In Section 4 we show how to exploit the LTS semantics for measuring the energy consumption of ad-hoc networks and comparing the average energy cost of networks exhibiting the same connectivity behaviour. In Section 5 we analyse the LAR protocol, comparing it with the simple flooding algorithm usually adopted in reactive routing. Section 6 carries out a quantitative and qualitative comparison of the Stop&Wait and Go-Back-N protocols under a specific scenario. Finally, Section 7 concludes the paper.

2 The Calculus

We introduce the Probabilistic EBUM calculus, an extension of EBUM (a calculus for Energy-aware Broadcast, Unicast, Multicast communications of mobile ad-hoc networks) [13] that models mobile ad-hoc networks as a collection of nodes, running in parallel, and using channels to broadcast messages. Our calculus supports multicast and unicast communications. Moreover, it allows us to model the possibility for a node to control the energy consumption by choosing the transmission radius for its communications.

Syntax. We use letters c and d for *channels*; m and n for *nodes*; l, k and h for *locations*; r for *transmission radii*; x, y and z for *variables*. *Closed values* contain nodes, locations, transmission radii and any basic value (e.g., 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 *Loc* for the set of all locations.

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 nodes, devices running in parallel and using channels to communicate messages. As usual, $\mathbf{0}$ denotes the empty network and $M_1|M_2$ the parallel composition of two networks. We denote by $\prod_{i\in I}M_i$ the parallel composition of the networks M_i, for $i \in I$. We denote by $n[P]_l$ a network node named n, located at the physical location l, and executing the process P. In $(\nu c)M$, the channel c is private with scope M, and we say it is bound in M: we denote by $fc(M)$ the set of channels which are not bound in M. We remark that in our calculus channels are distinct from values and cannot be transmitted; furthermore, given the structure of the syntactic productions, channels may not be dynamically created and thus $(\nu c)M$ simply plays the role of a CCS-style hiding operator[1]. We denote by $\mathcal{N}$ the set of all networks.

Processes are sequential and live within the nodes. Process $\mathbf{0}$ denotes the inactive process. 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 $\tilde{x}$ (where $|\tilde{x}| = |\tilde{w}|$, and $|\cdot|$ denotes the length of the tuple). In the process $c(\tilde{x}).P$, the variables in $\tilde{x}$ are said to be bound in P. Process $\bar{c}_{L,r}\langle\tilde{w}\rangle.P$ 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 physical locations of the intended recipients: $L = Loc$ represents a broadcast transmission, while a finite set of locations L denotes a multicast communication (unicast if L is a singleton). We remark that L is not a set of names, but it is a set of locations. This is due to the fact that we are interested in analyzing ad-hoc routing protocols where the devices are aware of their location and messages are routed efficiently by exploiting such information. If one wish to specify the final destination by means of the physical address of the device, then this should be encoded in the tuple representing the transmitted message, therefore resembling the role of the headers in the real implementation of the transmission protocols. The tag r represents the transmission radius of the sender: the choice of specific transmission ranges may depend on varoius parameters, and is left to the process running inside the transmitter node. We assume that the transmission radius of a communication cannot exceed the maximum transmission radius associated with the sending node. Syntactically, tags L and r associated with an output action on a channel c may be variables, but they must be instantiated when the output prefix is ready to fire. Process $[w_1 = w_2]P, Q$ behaves as P if $w_1 = w_2$, and as Q otherwise. We write $A\langle\tilde{w}\rangle$ to denote 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. We identify processes up to α-conversion and we assume that there are no free variables in a network. We write c_l for $c_{\{l\}}$, $\bar{c}_{L,r}\langle\tilde{w}\rangle$ for $\bar{c}_{L,r}\langle\tilde{w}\rangle.\mathbf{0}$, $\mathbf{0}$ for $n[\mathbf{0}]_l$ and $[w_1 = w_2]P$ for $[w_1 = w_2]P, \mathbf{0}$.

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 trans-

[1] Since channels represent radio frequencies, they are all public and may not be hidden in practice. Indeed, the use of the hiding operator is only meant to specialize the verification method to some specific class of contexts as we will see later.

<table>
<tr><th colspan="2">Networks</th><th colspan="2">Processes</th></tr>
<tr><td>M, N ::= $\mathbf{0}$</td><td>Empty network</td><td>P, Q, R ::= $\mathbf{0}$</td><td>Inactive process</td></tr>
<tr><td>$\mid M_1 \mid M_2$</td><td>Parallel composition</td><td>$\mid c(\tilde{x}).P$</td><td>Input</td></tr>
<tr><td>$\mid (\nu c)M$</td><td>Restriction</td><td>$\mid \bar{c}_{L,r}\langle\tilde{w}\rangle.P$</td><td>Output</td></tr>
<tr><td>$\mid n[P]_l$</td><td>Node (or device)</td><td>$\mid [w_1 = w_2]P, Q$</td><td>Matching</td></tr>
<tr><td></td><td></td><td>$\mid A\langle\tilde{w}\rangle$</td><td>Recursion</td></tr>
</table>

Table 1: Syntax

mission radius of the sender. We presuppose a function $d(\cdot, \cdot)$ which takes two locations and returns the distance separating them (function d can simply be the Euclidean distance between two locations, or a more complex function dealing with potential obstacles).

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 matrix of a discrete time Markov chain: each entry $\mathbf{J}^n_{lk}$ denotes the probability that the node n located at l may move to the location k. Hence, $\sum_{k \in Loc} \mathbf{J}^n_{lk} = 1$ for all locations $l \in Loc$ and nodes n. Static nodes inside a network are associated with the identity Markov chain, i.e., the identity matrix $\mathbf{J}^n_{ll} = 1$ for all $l \in Loc$ and $\mathbf{J}^n_{lk} = 0$ for all $k \neq l$. We denote by μ^n_l the probability distribution associated with node n located at l, that is, the function over Loc such that $\mu^n_l(k) = \mathbf{J}^n_{lk}$, for all $k \in Loc$[2]. We will model the probabilistic evolution of the network according to these distributions.

Probability distributions for networks. Let n be a node of a network M and l its location. We denote by $M\{n : l'/l\}$ the network obtained by substituting l by l' inside the node n and by $[\![M]\!]_{\mu^n_l}$ the probability distribution over the set of networks induced by μ^n_l and defined as follows: for all networks M',

$$[\![M]\!]_{\mu^n_l}(M') \;=\; \begin{cases} \mu^n_l(l') & \text{if } M' = M\{n : l'/l\} \\ 0 & \text{otherwise} \end{cases}$$

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

Example 1 (Probability distributions) Consider the network M defined as

$$n_1[\bar{c}_{L,r_1}\langle\tilde{v}_1\rangle.P_1]_{l_1} \mid n_2[\bar{c}_{L,r_2}\langle\tilde{v}_2\rangle.P_2]_{l_2} \mid m[c(\tilde{x}).P_3]_k$$

where two mobile nodes, n_1 and n_2, communicate with a static receiver node m. Both nodes n_1 and n_2 move back and forth between the two locations l_1 and l_2 according to the probability distribution defined by the discrete time Markov chain with the following transition matrix

$$\mathbf{J} = \begin{vmatrix} 1-p & p \\ q & 1-q \end{vmatrix},$$

where $0 < p, q < 1$. The probability distribution of the network induced by the movement of node n_1 is

$$[\![M]\!]_{\mu^{n_1}_{l_1}}(M') \;=\; \begin{cases} 1-p & \text{if } M' = M \\ p & \text{if } M' = M\{n_1 : l_2/l_1\} \\ 0 & \text{otherwise.} \end{cases}$$

Similarly for the second node we have

$$[\![M]\!]_{\mu^{n_2}_{l_2}}(M') \;=\; \begin{cases} 1-q & \text{if } M' = M \\ q & \text{if } M' = M\{n_2 : l_1/l_2\} \\ 0 & \text{otherwise.} \end{cases}$$

while for the static receiver we have

$$[\![M]\!]_{\mu^m_k}(M') \;=\; \begin{cases} 1 & \text{if } M' = M \\ 0 & \text{otherwise.} \end{cases}$$

Note that $[\![M]\!]_{\mu^m_k} = [\![M]\!]_{\Delta}$. □

Reduction semantics. The dynamics of the calculus is specified by the *probabilistic reduction relation* over networks ($\rightarrow$), described in Table 3. As usual, it relies on an auxiliary relation, called structural congruence ($\equiv$), which is the least contextual equivalence relation satisfying the rules defined in Table 2. The probabilistic reduction relation takes the form $M \rightarrow [\![M']\!]_{\theta}$ denoting a transition that leaves from network M and leads to a probability distribution $[\![M']\!]_{\theta}$.

[2] Notice that $\mathbf{J}^n$ is a matrix, while μ^n_l is a function.

$n[\mathbf{0}]_l \equiv \mathbf{0}$	(Struct Zero)
$n[[v = v]P, Q]_l \equiv n[P]_l$	(Struct Then)
$n[[v_1 = v_2]P, Q]_l \equiv n[Q]_l \quad v_1 \neq v_2$	(Struct Else)
$n[A\langle\tilde{v}\rangle]_l \equiv n[P\{\tilde{v}/\tilde{x}\}]_l$ if $A(\tilde{x}) \overset{\text{def}}{=} P \wedge \lvert\tilde{x}\rvert = \lvert\tilde{v}\rvert$	(Struct Rec)
$M \vert N \equiv N \vert M$	(Struct Par Comm)
$(M \vert N) \vert M' \equiv M \vert (N \vert M')$	(Struct Par Assoc)
$M \vert \mathbf{0} \equiv M$	(Struct Zero Par)
$(\nu c)\mathbf{0} \equiv \mathbf{0}$	(Struct Zero Res)
$(\nu c)(\nu d)M \equiv (\nu d)(\nu c)M$	(Struct Res Res)
$(\nu c)(M \mid N) \equiv M \mid (\nu c)N$ if $c \notin fc(M)$	(Struct Res Par)

Table 2: Structural Congruence

Rule (R-Bcast) models the transmission of a tuple of messages $\tilde{v}$ to the set of locations L using channel c and transmission radius r. Indeed, nodes communicate using radio frequencies that enable only message broadcasting (monopolizing channels is not permitted). However, a node may decide to communicate with a specific node (or group of nodes), this is the reason why we decided to associate with each output action a set of target locations. The cardinality of this set indicates the kind of communication that is used: if $L = Loc$ then the recipients set is the whole network and a broadcast transmission is performed, while if L is a finite set (resp., a singleton) then a multicast (resp., a unicast) communication is done. Notice that L does not play a role in a synchronization reduction, as messages are broadcast and observable (and received) by any active receiver in range. On the other hand, we use L to fine-tune our notion of observation in the definition of barb. Moreover, the index set I in rule (R-Bcast) could be empty, because the output is a non-blocking action, i.e., it could be applied even if no nodes are ready to receive the transmission. A radius r is also associated with an output action on channel c, indicating the transmission radius required for that communication which may depend on the energy consumption strategy adopted by the surrounding protocol.

Rule (R-Move) deals with the possibility for a node to move within the network. A node n located at l and executing a move action will reach a location with a probability described by the distribution μ_l^n that depends on the Markov chain $\mathbf{J}^n$ statically associated with n. Movements are atomic actions: while moving, a node cannot do anything else. In our model, due to the interleaving nature of the calculus, only one node can move at each reduction but this does not mean that only one node can move at a time. Indeed, as usual in interleaving semantics, concurrent events are represented by sequentiality and non-determinism. Rules (R-Par), (R-Res) and (R-Struct) are standard.

Since we are dealing with a probabilistic reduction semantics, which reduces 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

$$M \!\to_\theta\! N$$

if $M \!\to\! \llbracket M' \rrbracket_\theta$ and N is in the support of $\llbracket M' \rrbracket_\theta$. Following [17], an execution for M is a (possibly infinite) sequence of steps $M \!\to_{\theta_1}\! M_1 \!\to_{\theta_2}\! M_2 ....$ We write $Exec_M$ for the set of all possible executions starting from M, $last(e)$ for the final state of a *finite* execution e, e^j for the prefix $M \!\to_{\theta_1}\! M_1 ... \!\to_{\theta_j}\! M_j$ of length j of the execution e of the form $M \!\to_{\theta_1}\! M_1 \cdots \!\to_{\theta_j}\! M_j \!\to_{\theta_{j+1}}\! M_{j+1} \cdots$, and $e\!\uparrow$ for the set of e' such that e is a prefix of e'. The symbol $\to^*$ denotes the transitive and reflexive closure of $\to$.

Observational semantics. Following a standard practice, we formalize the observational semantics for our calculus in terms of a notion *barb*, that provides the basic unit of observation [30]. As in other calculi for wireless communications, the definition of barb is naturally expressed in terms of message transmission. However, the technical development is more involved, as our calculus presents both non-deterministic and probabilistic aspects, where the non-deterministic choices are among the possible probability distributions that a network may follow and arise from the possibility for nodes to perform movements according to the associated discrete time Markov chain.

We denote by $behave(M) = \{\llbracket M' \rrbracket_\theta \mid M \to \llbracket M' \rrbracket_\theta\}$ the set of the possible behaviours of M. In order to solve the non-determinism in a network execution, we consider each possible probabilistic transition $M \to \llbracket M' \rrbracket_\theta$ as arising from a *scheduler* (see [39]).

Definition 1 (Scheduler) A *scheduler* is a total function F assigning to a finite execution e a distribution $\llbracket N \rrbracket_\theta \in behave(last(e))$.

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

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

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

$$\text{(R-Res)}\ \frac{M \rightarrow [\![M']\!]_\theta}{(\nu\tilde{c})M \rightarrow [\![(\nu\tilde{c})M']\!]_\theta} \qquad \text{(R-Struct)}\ \frac{N \equiv M \;\; M \rightarrow [\![M']\!]_\theta \;\; M' \equiv N'}{N \rightarrow [\![N']\!]_\theta}$$

Table 3: Reduction Semantics

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

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

Given a finite execution $e = M \rightarrow_{\theta_1} M_1 ... \rightarrow_{\theta_k} M_k$ starting from a network M and driven by a scheduler F we define

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

where $\forall j \leq k,\ [\![M_j']\!]_{\theta_j} = F(e^{j-1})$. We define the probability space on the executions starting from a given network M as follows. Given a scheduler F, $\sigma Field_M^F$ is the smallest sigma 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 crossing 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 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 location in the set of the target locations is able to receive the message.

Definition 2 (Barb) Let $M \equiv (\nu\tilde{d})(n[\bar{c}_{L,r}\langle\tilde{v}\rangle.P]_l|M')$, with $c \notin \tilde{d}$. 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 $d(l,k) \leqslant r$ for all $k \in K$.

Definition 3 (Probabilistic Barb) A network M has a *probabilistic barb* with probability p on a channel c to the set K of locations, according to the scheduler F, written $M \Downarrow_p^F c@K$, if $Prob_M^F(\{N|N \downarrow_{c@K}\}) = p$.

Intuitively, for a given network M and a scheduler F, if $M \Downarrow_p^F c@K$ then p is the positive probability that M, driven by F, performs a transmission on channel c and at least one of the receivers in the observation locations is able to correctly listen to it.

In the following, we introduce a probabilistic observational congruence, in the style of [17], which is parametric to a restricted set of schedulers. This allows us to ignore unrealistic schedulers like, for example, schedulers giving priority to communication actions over movements, thus canceling the effects that nodes mobility has on the network behaviour.

In order to define a congruence relation among networks, we have to select a set of schedulers guaranteeing that, for each behaviour a network can exhibit, the same behaviour can be exhibited by the network in the presence of any possible context. Hereafter, a context is a network term with a hole $[\cdot]$ defined by the following grammar:

$$\mathcal{C}[\cdot] ::= [\cdot] \mid [\cdot]|M \mid M|[\cdot] \mid (\nu c)[\cdot].$$

The following definition allows us to select the set of schedulers preserving the contextuality, once we have fixed the particular behaviour we want to capture.

Definition 4 Given a scheduler $F \in Sched$, we denote by $F_{\mathcal{C}}$ the set of schedulers F' such that $\forall M_0$, $\forall e \in Exec_{M_0}^F$ of the form

$e = M_0 \rightarrow_{\theta_1} M_1 \rightarrow_{\theta_2} M_2 ... \rightarrow_{\theta_h} M_h$,

$\forall$ context $C_0[\cdot]$ and $\forall e' \in Exec_{C_0[O_0]}^{F'}$ with $M_0 \equiv O_0$ of the form

$e' = C_0[O_0] \rightarrow_{\theta_1'} C_1[O_1] \rightarrow_{\theta_2'} C_2[O_2] ... \rightarrow_{\theta_k'} C_k[O_k]$,

there exists a monotonic surjective function f from $[0-k]$ to $[0-h]$ such that:

(i) $\forall i \in [0-k]$, $O_i \equiv M_{f(i)}$

(ii) $\forall j \in [1-k]$, $\theta_j' = \theta_{f(j)}$ if $M_{f(j-1)} \rightarrow_{\theta_{f(j)}} M_{f(j)}$.

Given a subset $\mathcal{F} \in Sched$ of schedulers, then we define $\mathcal{F}_{\mathcal{C}} = \bigcup_{F\in\mathcal{F}} F_{\mathcal{C}}$.

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

$$\text{(Then)}\ \frac{P \xrightarrow{\eta} P'}{[\tilde{v} = \tilde{v}]P, Q \xrightarrow{\eta} P'} \qquad \text{(Else)}\ \frac{Q \xrightarrow{\eta} Q' \quad \tilde{v}_1 \neq \tilde{v}_2}{[\tilde{v}_1 = \tilde{v}_2]P, Q \xrightarrow{\eta} Q'}$$

$$\text{(Rec)}\ \frac{P\{\tilde{v}/\tilde{x}\} \xrightarrow{\eta} P' \quad A(\tilde{x}) \overset{\text{def}}{=} P}{A\langle\tilde{v}\rangle \xrightarrow{\eta} P'}$$

Table 4: LTS rules for Processes

Example 2 Let $M_0 \equiv m[\bar{c}_{L,r}\langle v\rangle.P]_l$ and $F \in Sched$ such that

$$M_0 \rightarrow_{\Delta} M_1 \in Exec^F_M,$$

with $M_1 \equiv m[P]_l$.

First notice that $F \in F_{\mathcal{C}}$, since we can take the empty context $C[\cdot] \equiv [\cdot]$ and the identity function f such that $f(i) = i$ for all $i \in [0-1]$. In this case $C[M_i] \equiv M_i$ for $i \in \{0, 1\}$ and the property of Definition 4 is satisfied.

Let now consider $N_0 \equiv n[c(x).Q]_k$ such that $d(l, k) \leq r$. All the schedulers allowing M_0 and N_0 to interact are in $F_{\mathcal{C}}$. Indeed, consider $F_1 \in Sched$ such that, by applying rules (R-Bcast),

$$M_0 \mid N_0 \rightarrow_{\Delta} M_1 \mid N_1 \in Exec^{F_1}_{M_0|N_0}$$

with $N_1 \equiv n[Q\{v/x\}]_k$, and consider also F_2 such that, by applying rule (R-Par)

$$M_0 \mid N_0 \rightarrow_{\Delta} M_1 \mid N_0 \in Exec^{F_2}_{M_0|N_0}.$$

Both F_1 and F_2 satisfy the properties of Definition 4, hence $F_1, F_2 \in F_{\mathcal{C}}$.

Now consider again the network N_0.
Let $e' = n[c(x).Q]_k \rightarrow_{\mu^n_k} n[c(x).Q]_{k'} \notin Exec^F_{N_0}$, then $\forall \bar{F} \in Sched$ such that $e' \in Exec^{\bar{F}}_{N_0}$, $\bar{F} \notin F_{\mathcal{C}}$ since $\bar{F}$ does not satisfy the conditions of Definition 4. □

Now we are able to introduce our equivalence relation.

Definition 5 Given a set $\mathcal{F} \in Sched$ of scehdulers, and a relation $\mathcal{R}$ over networks:

- *Barb preservation.* $\mathcal{R}$ is *barb preserving* relative to $\mathcal{F}$ if $M\mathcal{R}N$ and $M{\Downarrow}^F_p c@K$ for some $F \in \mathcal{F}_{\mathcal{C}}$ implies that there exists $F' \in \mathcal{F}_{\mathcal{C}}$ such that $N{\Downarrow}^{F'}_p c@K$.
- *Reduction closure.* $\mathcal{R}$ 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 $\mathcal{C} \in \mathcal{N}/\mathcal{R}$, $Prob^F_M(\mathcal{C}) = Prob^{F'}_N(\mathcal{C})$.
- *Contextuality.* $\mathcal{R}$ is *contextual* if $M\mathcal{R}N$ implies that for every context $\mathcal{C}[\cdot]$, it holds that $\mathcal{C}[M]\,\mathcal{R}\,\mathcal{C}[N]$.

Our probabilistic observational congruence with respect to a restricted set $\mathcal{F}$ of schedulers is defined as the largest relation as follows.

Definition 6 (Observational Congruence) Given a set $\mathcal{F}$ of schedulers, the *probabilistic observational congruence relative to* $\mathcal{F}$, written $\cong^{\mathcal{F}}_p$, is the largest symmetric relation over networks which is reduction closed, barb preserving and contextual.

Two networks are related by $\cong^{\mathcal{F}}_p$ if they exhibit the same probabilistic behaviour (communications) relative to the corresponding sets of intended recipients. In the next section we develop a bisimulation-based proof technique for $\cong^{\mathcal{F}}_p$.

3 A Bisimulation-based Proof Technique

The proof of relation $\cong^{\mathcal{F}}_p$ may be a hard task. In this section we propose a co-inductive proof technique that allows for an algorithmic decision of $\cong^{\mathcal{F}}_p$.

Labelled Transition Semantics. We define a LTS semantics for our calculus, which is built upon two sets of rules: one for processes and one for networks. Table 4 presents the LTS rules for processes. Transitions are of the form $P \xrightarrow{\eta} P'$, where η ranges over input and output actions of the form:

$$\eta ::= c\tilde{v} \mid \bar{c}_{L,r}\tilde{v}.$$

Rules for processes are simple and they do not need deeper explanations. Notice that such rules do not rely on any probabilistic notion since processes only have deterministic transitions.

Table 5 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

$$\text{(Snd)}\ \frac{P \xrightarrow{\bar{c}_{L,r}\tilde{v}} P'}{n[P]_l \xrightarrow{c_L!\tilde{v}[l,r]} [\![n[P']_l]\!]_\Delta} \qquad \text{(Rcv)}\ \frac{P \xrightarrow{c\tilde{v}} P'}{n[P]_l \xrightarrow{c?\tilde{v}@l} [\![n[P']_l]\!]_\Delta}$$

$$\text{(Bcast)}\ \frac{M \xrightarrow{c_L!\tilde{v}[l,r]} [\![M']\!]_\Delta \quad N \xrightarrow{c?\tilde{v}@l'} [\![N']\!]_\Delta \quad d(l,l') \leq r}{\begin{array}{c} M|N \xrightarrow{c_L!\tilde{v}[l,r]} [\![M'|N']\!]_\Delta \\ \scriptstyle N|M \xrightarrow{c_L!\tilde{v}[l,r]} [\![N'|M']\!]_\Delta \end{array}}$$

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

$$\text{(Lose)}\ \frac{M \xrightarrow{c_L!\tilde{v}[l,r]} [\![M']\!]_\Delta}{M \xrightarrow{\tau} [\![M']\!]_\Delta} \qquad \text{(Move)}\ \frac{}{n[P]_l \xrightarrow{\tau} [\![n[P]_l]\!]_{\mu_l^n}}$$

$$\text{(Par)}\ \frac{M \xrightarrow{\gamma} [\![M']\!]_\theta}{\begin{array}{c} M|N \xrightarrow{\gamma} [\![M'|N]\!]_\theta \\ \scriptstyle N|M \xrightarrow{\gamma} [\![N|M']\!]_\theta \end{array}} \qquad \text{(Res)}\ \frac{M \xrightarrow{\gamma} [\![M']\!]_\theta \quad \mathtt{Chan}(\gamma) \neq c}{(\nu c)M \xrightarrow{\gamma} [\![(\nu c)M']\!]_\theta}$$

Table 5: LTS rules for Networks

are used to model the mobility of nodes. Tag γ is defined as follows:

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

Rule (Snd) models the sending of tuple $\tilde{v}$ through channel c to a specific set L of locations with transmission radius r, while rule (Rcv) models the reception of $\tilde{v}$ at l via channel c. 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 $c!\tilde{v}@K \triangleleft R$ represents the transmission of the tuple $\tilde{v}$ of messages via c: 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 are used to denote non-observable actions. Rule (Move) models migration of a mobile node n from a location l to a location k according to the probability distribution μ_l^n, which depends on the Markov chain $\mathbf{J}^n$ statically associated with n. Rule (Res) models the standard channel restriction, where $\mathtt{Chan}(\gamma) = c$ if γ is of the form $c?\tilde{v}@l$ or $c_L!\tilde{v}[l,r]$ or $c!\tilde{v}@K \triangleleft R$, and $\mathtt{Chan}(\tau) = \bot$. Finally, (Par) is standard.

Relating the LTS and reduction semantics. We prove that the LTS-based semantics coincides with the reduction semantics and the notion of observability (barb) given in the previous section.

We first prove that if $M \xrightarrow{\gamma} [\![M']\!]_\Delta$, then the structure of M and M' can be determined up to structural congruence.

Lemma 1 *Let M be a network.*

1. *If $M \xrightarrow{c?\tilde{v}@l} [\![M']\!]_\Delta$, then there exist n, $\tilde{x}$, a (possibly empty) sequence $\tilde{d}$ such that $c \notin \tilde{d}$, a process P and a (possibly empty) network M_1 such that*
$$M \equiv (\nu\tilde{d})(n[c(\tilde{x}).P]_l | M_1)$$
and
$$M' \equiv (\nu\tilde{d})(n[P\{\tilde{v}/\tilde{x}\}]_l | M_1).$$
2. *If $M \xrightarrow{c_L!\tilde{v}[l,r]} [\![M']\!]_\Delta$, then there exist n, a (possibly empty) sequence $\tilde{d}$ such that $c \notin \tilde{d}$, a process P, a (possibly empty) network M_1 and a (possibly empty) set I, with $d(l,l_i) \leq r\ \forall i \in I$, such that:*
$$M \equiv (\nu\tilde{d})(n[\bar{c}_{L,r}\langle\tilde{v}\rangle.P]_l | \prod_{i\in I} n_i[c(\tilde{x}_i).P_i]_{l_i} \mid M_1)$$
and
$$M' \equiv (\nu\tilde{d})(n[P]_l | \prod_{i\in I} n_i[P_i\{\tilde{v}/\tilde{x}_i\}]_{l_i} \mid M_1).$$

Proof The proof follows by induction on the transition rules of Table 5. □

Now we show that the structural congruence respects the transitions of Table 5.

Lemma 2 *If $M \xrightarrow{\gamma} [\![M']\!]_\theta$ and $M \equiv N$, then there exists N' such that $N \xrightarrow{\gamma} [\![N']\!]_\theta$ and $M' \equiv N'$.*

Proof The proof is derived by induction on the depth of the inference $M \xrightarrow{\gamma} [\![M']\!]_\theta$. □

The following theorem establishes the relationship between the reduction semantics and the LTS one.

Theorem 1 (Harmony) *Let M be a network.*

1. *If $M \to [\![M']\!]_\theta$ then there exist $N \equiv M$ and $N' \equiv M'$ such that $N \xrightarrow{\tau} [\![N']\!]_\theta$.*
2. *$M \downarrow_{c@K}$ if and only if there exist $\tilde{v}$, $R \supseteq K$ and $N \equiv M$ such that $N \xrightarrow{c!\tilde{v}@K \lhd R}$.*
3. *If $M \xrightarrow{\tau} [\![M']\!]_\theta$ then $M \to [\![M']\!]_\theta$.*
4. *If $M \xrightarrow{c!\tilde{v}@K \lhd R} [\![M']\!]_\Delta$ then $M \to [\![M']\!]_\Delta$.*

Proof See Appendix. □

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

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

Again, we write $M \xrightarrow{\alpha}_\theta N$ if $M \xrightarrow{\alpha} [\![M']\!]_\theta$ and N is in the support of $[\![M']\!]_\theta$. A *labelled execution* e of a network M is a finite (or infinite) sequence of steps:

$$M \xrightarrow{\alpha_1}_{\theta_1} M_1 \xrightarrow{\alpha_2}_{\theta_2} M_2 \ldots \xrightarrow{\alpha_k}_{\theta_k} M_k\,.$$

With abuse of notation, we define $Exec_M$, $last(e)$, e^j and $e\uparrow$ as for unlabeled executions. Moreover, we denote by $lbehave(M)$ the set of all possible behaviours of M, i.e., $lbehave(M) = \{(\alpha, [\![M']\!]_\theta) \mid M \xrightarrow{\alpha} [\![M']\!]_\theta\}$. Labelled executions arise by resolving the non-determinism of both α and $[\![M]\!]_\theta$. As a consequence, a scheduler[3] for the labelled semantics is a function F assigning a pair $(\alpha, [\![M]\!]_\theta) \in lbehave(last(e))$ with a finite labelled execution e. We denote by $LSched$ the set of all schedulers for the LTS semantics. Given a network M and a scheduler F, we define $Exec^F_M$ as the set of all labelled executions starting from M and driven by F.

From a modelling point of view, we want to distinguish networks that differ for some observable actions, therefore ignoring internal computations of the nodes. Formally, this means that we are interested in weak observational equivalences, that abstract over τ-actions. Hereafter, we introduce the notion of *weak action.*

Definition 7 (Weak Action) We denote by $\Longrightarrow$ the transitive and reflexive closure of $\xrightarrow{\tau}$ and by $\xRightarrow{\alpha}$ the weak action $\Longrightarrow\xrightarrow{\alpha}\Longrightarrow$. We denote by $\xRightarrow{\hat{\alpha}}$ the weak action $\xRightarrow{\alpha}$ if $\alpha \neq \tau$, and $\Longrightarrow$ otherwise.

We denote by $Exec^F_M(\xRightarrow{\alpha}, H)$ the set of executions that, starting from M, according to the scheduler F, lead to a network in the set H by performing $\xRightarrow{\alpha}$. Moreover, $Prob^F_M(\xRightarrow{\alpha}, H) = Prob^F_M(Exec^F_M(\xRightarrow{\alpha}, H))$.

Since we want our bisimilarity to be a complete characterisation of our notion of behavioural equivalence, which has been defined with respect to a restricted set of schedulers $\mathcal{F} \subseteq Sched$ on the reduction semantics, we have to define the set of schedulers $\hat{\mathcal{F}} \in LSched$ for the LTS corresponding to $\mathcal{F}$.

Definition 8 Given a scheduler $F \in Sched$, we denote by $\hat{F}_{\mathcal{C}} \subseteq LSched$ the set of schedulers $\hat{F} \in LSched$ such that $\forall M_0$, $\forall e \in Exec^{\hat{F}}_{M_0}$:

$$e = M_0 \xrightarrow{\alpha_1}_{\theta_1} M_1 \ldots \xrightarrow{\alpha_k}_{\theta_h} M_h$$

$\exists F' \in F_{\mathcal{C}}$, a context C_0 and $e' \in Exec^{F'}_{C_0[O_0]}$ with $O_0 \equiv M_0$ such that

$$e' = C_0[O_0] \to_{\theta'_1} C_1[O_1] \ldots \to_{\theta'_k} C_k[O_k]$$

and there exists a monotone surjective function f from $[0-k]$ to $[0-h]$ such that:

(i) $\forall i \in [1-k]\ O_i \equiv M_{f(i)}$

(ii) $\forall j \in [1-k]$, $\theta_{f(j)} = \theta'_j$ if $M_{f(j-1)} \xrightarrow{\alpha_{f(j)}}_{\theta_{f(j)}} M_{f(j)}$.

For a given a set $\mathcal{F} \subseteq Sched$ of schedulers, we define $\hat{\mathcal{F}}_{\mathcal{C}} = \bigcup_{F \in \mathcal{F}} \hat{F}_{\mathcal{C}}$.

Example 3 Consider the networks M_0 and N_0, and the schedulers F and F_1 introduced in the Example 2. If we take $\hat{F}_1 \in LSched$ such that

$$M_0 \xrightarrow{c_L!v[l,r]}_\Delta M_1 \in Exec^{\hat{F}_1}_{M_0},$$

then, since

$$M_0 \to_\Delta M_1 \in Exec^F_{M_0}$$

the conditions of Definition 8 are satisfied by taking the empty context $C[\cdot] = [\cdot]$ and the identity function $f(i) = i$ for $i \in \{0,1\}$. Hence $\hat{F}_1 \in \hat{F}_{\mathcal{C}}$.

Moreover, if we consider $\hat{F}_2 \in LSched$ such that

$$N_0 \xrightarrow{c?v@k}_\Delta N_1 \in Exec^{\hat{F}_2}_{N_0},$$

since

$$M_0 \mid N_0 \to_\Delta M_1 \mid N_1 \in Exec^{F_1}_{M_0|N_0}$$

with $F_1 \in F_{\mathcal{C}}$, by considering the contexts $C_i[\cdot] \equiv M_i \mid \cdot$ for $i \in \{0,1\}$, and the identity function $f(i) = i$ for $i \in \{0,1\}$ we get $\hat{F}_2 \in \hat{F}_{\mathcal{C}}$ too. □

[3] With abuse of notation, we still use F to denote a scheduler for the LTS semantics.

The following proposition holds.

Proposition 1

1. $Sched_{\mathcal{C}} = Sched.$
2. $\widehat{Sched}_{\mathcal{C}} = LSched.$

Proof The first statement follows straightforwardly from Definition 4. To prove the second statement observe that: $\forall F \in LSched$, $\forall M_0 \in \mathcal{N}$ and $\forall e \in Exec^{F}_{M_0}$ of the form
$e = M_0 \xrightarrow{\alpha_1}_{\theta_1} M_1 ... \xrightarrow{\alpha_k}_{\theta_k} M_k$
it is always possible to find a context $C_0[\cdot]$ and a scheduler $F' \in LSched$ such that $e' \in Exec^{F'}_{C_0[M_0]}$ with
$e' = C_0[M_0] \xrightarrow{\tau}_{\theta_1} ...C_1[M_1]... \xrightarrow{\tau}_{\theta_k} C_k[M_k]$.
By theorem 1, $\exists F'' \in Sched$ such that $e'' \in Exec^{F''}_{C_0[M_0]}$ with
$e'' = C_0[M_0] \rightarrow_{\theta_1} ...C_1[M_1]... \rightarrow_{\theta_k} C_k[M_k]$,
meaning that $F \in \widehat{Sched}_{\mathcal{C}}$ as required. □

In the following we give the definition of probabilistic labelled bisimilarity relative to a given set of schedulers. In the definition below input actions are treated differently from output and silent actions. This is due to the fact that in our model the input is not an observable action, hence two systems are considered equivalent even if they do not have the same behaviour in terms of transmission receptions.

Definition 9 (Probabilistic Labelled Bisimilarity) Let M and N be two networks. An equivalence relation $\mathcal{R}$ over networks is a *probabilistic labelled bisimulation* relative to $\mathcal{F}$ if $M\mathcal{R}N$ implies: for all scheduler $F \in \hat{\mathcal{F}}_{\mathcal{C}}$ there exists a scheduler $F' \in \hat{\mathcal{F}}_{\mathcal{C}}$ such that for all α and for all classes $\mathcal{C}$ in $\mathcal{N}/\mathcal{R}$ it holds:

1. if $\alpha \neq c?\tilde{v}@l$ then
$Prob^{F}_{M}(\xrightarrow{\alpha}, \mathcal{C}) = Prob^{F'}_{N}(\overset{\hat{\alpha}}{\Longrightarrow} \mathcal{C})$;
2. if $\alpha = c?\tilde{v}@l$ then either
$Prob^{F}_{M}(\xrightarrow{\alpha}, \mathcal{C}) = Prob^{F'}_{N}(\overset{\alpha}{\Longrightarrow}, \mathcal{C})$ or
$Prob^{F}_{M}(\xrightarrow{\alpha}, \mathcal{C}) = Prob^{F'}_{N}(\Longrightarrow, \mathcal{C})$.

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

A complete characterisation. In this part, we finally prove that our probabilistic labelled bisimilarity is a complete characterisation of the probabilistic observational congruence of Definition 6.

We first state the following proposition.

Proposition 2 *Let M and N be two networks. If $M\mathcal{R}N$ for some bisimulation $\mathcal{R}$ w.r.t $\mathcal{F}$, then 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 $\mathcal{N}/\mathcal{R}$ it holds:*

1. *if $\alpha \neq c?\tilde{v}@l$ then*
$Prob^{F}_{M}(\overset{\hat{\alpha}}{\Longrightarrow}, \mathcal{C}) = Prob^{F'}_{N}(\overset{\hat{\alpha}}{\Longrightarrow} \mathcal{C})$;
2. *if $\alpha = c?\tilde{v}@l$ then either*
$Prob^{F}_{M}(\overset{\hat{\alpha}}{\Longrightarrow}, \mathcal{C}) = Prob^{F'}_{N}(\overset{\alpha}{\Longrightarrow}, \mathcal{C})$ *or*
$Prob^{F}_{M}(\overset{\alpha}{\Longrightarrow}, \mathcal{C}) = Prob^{F'}_{N}(\Longrightarrow, \mathcal{C})$.

Proof The proof follows by induction on the length of the weak transition $\overset{\hat{\alpha}}{\Longrightarrow}$. □

We can now prove that our bisimilarity is a proof method for our observational congruence, i.e., that $\approx^{\mathcal{F}}_{p}$ is contained in $\cong^{\mathcal{F}}_{p}$.

Theorem 2 (Soundness) *Let M and N be two networks and $\mathcal{F} \subseteq Sched$. If $M \approx^{\mathcal{F}}_{p} N$ then $M \cong^{\mathcal{F}}_{p} N$.*

Proof See Appendix. □

Finally, we prove that the observational congruence is contained the labelled bisimilarity.

Theorem 3 (Completeness) *Let M and N be two networks and $\mathcal{F} \subseteq Sched$. If $M \cong^{\mathcal{F}}_{p} N$ then $M \approx^{\mathcal{F}}_{p} N$.*

Proof See Appendix. □

The following result is a consequence of Theorems 2 and 3.

Theorem 4 (Characterization) *For every set $\mathcal{F} \subseteq Sched$, $\cong^{\mathcal{F}}_{p} = \approx^{\mathcal{F}}_{p}$.*

4 Measuring Energy Consumption

In this section, based on the LTS semantics, we define a preorder over networks which allows us to compare the average energy cost of different networks but exhibiting the same connectivity behaviour relative to a specific set of schedulers $\mathcal{F}$. For this purpose we associate an energy cost with labelled transitions as follows:

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

In other words, the energy cost to reach N from M in one single step is r if M can reach N after firing on a channel of radius[4] r regardless of the message being transmitted is observable or not (or even lost). In the same way, if

$$e = M_0 \xrightarrow{\alpha_1}_{\theta_1} M_1 ... \xrightarrow{\alpha_k}_{\theta_k} M_k$$

[4] Note that considering the radius of the communication channel as the energy cost of the transmitted data is standard [45,6].

is an execution then

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

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 F which are not prefix of any other execution ending in H. More formally,

$$\mathit{Paths}_M^F(H) = \{e \in \mathit{Exec}_M^F(H) \mid \mathit{last}(e) \in H \text{ and } \forall e' \text{ such that } e \text{ is a prefix of } e',\ e' \notin \mathit{Paths}_M^F(H)\}.$$

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

Definition 10 Let H be a set of networks. The average energy cost of reaching H from M according to scheduler F is

$$\mathbf{Cost}_M^F(H) = \frac{\sum_{e\in \mathit{Paths}_M^F(H)}\mathbf{Cost}(e) \times P_M^F(e)}{\sum_{e\in \mathit{Paths}_M^F(H)} P_M^F(e)}.$$

Basically, 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 of mobile networks.

Definition 11 Let $\mathcal{H}$ be a countable set of sets of networks and let $\mathcal{F} \subseteq \mathit{Sched}$ a set of schedulers. We say that N is *more energy efficient than* M relative to $\mathcal{H}$ and $\mathcal{F}$, denoted

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

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

5 Analysis of a location based routing protocol

In this section we consider a network of nodes with mobility and using the Location Aided Routing protocol (LAR) [22]. LAR aims at reducing the number of the packet floods with respect to what is observable in other protocols such as the AODV [36]. This is achieved by assuming that the nodes are aware of their own absolute or relative positions, e.g., because they are equipped with a GPS device [21] or because they are able to derive their distances from a set of fixed nodes. With respect to the analysis of LAR presented in [5], here we consider a quantitative approach that allows us to study the energy efficiency of LAR with respect to AODV under different scenarios. In order to carry out this comparison, we encode the AODV and LAR models described by means of the process calculus that we have defined into a PRISM program [23] and we perform a statistical model checking to estimate the energy consumptions of the protocols. We also prove that AODV and LAR are behaviourally equivalent, i.e., under the modelling assumptions, a packet is correctly delivered by AODV if and only if it is correctly delivered by LAR.

Protocol Description. In very large mobile networks using flooding strategies such as in an AODV style [36] may be very expensive in terms of number of sent packets and hence of node energy consumption. LAR reduces the effect of flooding by guessing the possible location of the destination node. The guess can be driven by several factors, such as the knowledge of the destination node's location in the latest communication joint with some assumptions on the node's maximum movement speed. In this section, we show our framework at work on a simplified version of the LAR protocol, and prove that, under mild assumptions on the node mobility, it is equivalent to the flooding algorithm in terms of the probability of discovering a path. Although it is not possible to establish a general energy-aware preorder between the two protocols, we carry out a statistical model checking to compare some instances of mobile networks.

Simple flooding. The LAR protocol extends the route discovery based on flooding by exploiting information about locations within the network. The simplest route discovery algorithm based on flooding consists of three simple packets: request, reply and error [44], which are forwarded within the network. They are structured as follows:

- *Route Request* packet (RREQ) has the form:

  $$(S, \mathit{Bid}, D, \texttt{seq\#}_S, \texttt{hop\_counter}),$$

  where S is the permanent source address, Bid is the Request Id (unique identifier), D is the permanent address of the destination, $\texttt{seq\#}_S$ denotes the sequence number of the source, and `hop_counter` is the number of hops to reach the destination (which is initially set to 0 and then incremented at each request forwarding).
- *Route Reply* packet (RREP) has the form:

  $$(S, \mathit{Bid}, D, \texttt{seq\#}_D, \texttt{hop\_counter}, \texttt{Lifetime}),$$

  where S, Bid and D are as above, $\texttt{seq\#}_D$ is the sequence number of the destination, `hop_counter` is the number of hops to reach the destination and `Lifetime` is the duration of the route validity.

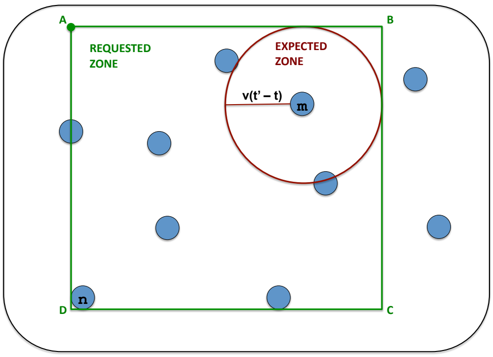


Fig. 1: *Expected* and *Request Zones* in the LAR protocol

- *Route Error* packet (RERR) has the form:

$$(S, D, \texttt{seq}\#_D)\,,$$

where S, D and $\texttt{seq}\#_D$ are as in the previous case.

In flooding algorithms, a node looking for a path to a given destination, simply broadcasts a RREQ within the network. Having sent the packet, the node sets a timeout to manage the cases when the destination does not receive the request, or the reply packet is lost. If the timeout expires, the node broadcasts a new request, using a different sequence number to avoid loops. When the destination finally receives the RREQ, it immediately sends back the corresponding RREP, using unicast communication, i.e., each intermediate node forwards the RREP using the information in its routing table. When, during a communication, a node realizes that a link failed, it broadcasts a RERR and each node will update its routing table.

Exploiting location data: the LAR policy. LAR modifies the flooding algorithm by directing the propagation of the discovery packets to a particular network area based on the expected locations of the destination node which are called *Expected Zone*. This is determined by using the information that the source has previously collected about the destination location. If node S knows that the destination node D was located at location l_1 at epoch t, and it moves with a speed v, then it can calculate the circle area centered at l_1, with radius $v(t' - t)$, where t' is the current epoch. If S does not know anything about D, then the *Expected Zone* coincides with the entire network.

The *Request Zone* is the network area that the source defines to specify a candidate route to the destination. An intermediate node forwards a route request only if it is within the *Request Zone*. There are different ways to define a *Request Zone*: usually choosing a smaller area reduces the message overhead (because it reduces the number of forwarded packets), while a larger area reduces the latency of the route discovery because the network finds a path with higher probability.

LAR behaves similarly to the simple flooding, with the difference that a node that is not inside the *Request Zone* does not forward the request. LAR can use two different policies for determining the *Request Zone*: we focus on the first of such policies, known as *LAR Scheme 1*.

LAR Scheme 1 uses a rectangular *Request Zone*, depending on the position of the source with respect to the *Expected Zone*. In particular, the *Request Zone* will be the smallest rectangle containing both the *Expected Zone* and the position of the source node, as shown in Figure 1.

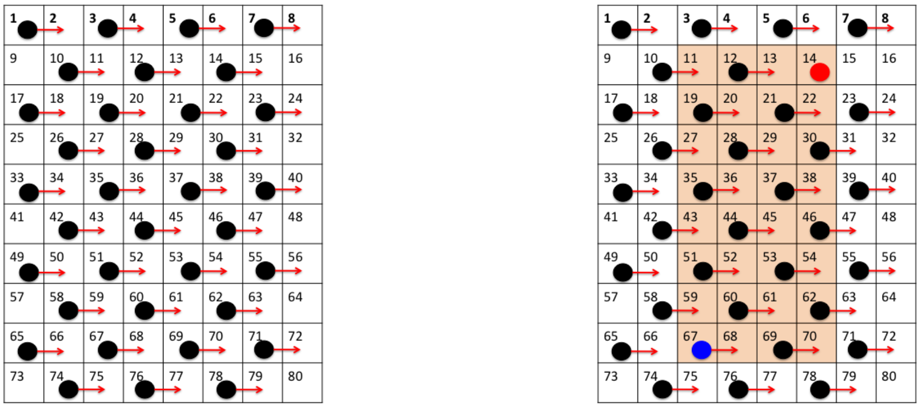
Let (X_S, Y_S) and (X_D, Y_D) the Cartesian coordinates of S and D, and R the radius of the *Expected Zone*. If S is outside the *Expected Zone*, the coordinates of the rectangle area are:

A: $\rightarrow (X_S, Y_D + R)$	B: $\rightarrow (X_D + R, Y_D + R)$
C: $\rightarrow (X_D + R, Y_S)$	D: $\rightarrow (X_S, Y_S)$

If S falls inside the *Expected Zone*, the coordinates of the rectangle area are:

A: $\rightarrow (X_D - R, Y_D + R)$	B:$\rightarrow (X_D + R, Y_D + R)$
C: $\rightarrow (X_D - R, Y_D - R)$	D: $\rightarrow (X_D + R, Y_D - R)$

When S broadcasts its request, it includes the coordinates of the *Request Zone* rectangle (see Figure 1). Once an intermediate node receives a RREQ, this is discarded if its location does not fall within the rectangle specified in the packet. To take into account the location measuring error, a positive value e is added to the radius of the *Expected Zone*, consequently enlarging also the *Request Zone*.



(a) Flooding Area

(b) Location-Aided Routing Area

Fig. 2: Topology of the network

Modelling the network. We encode the simple flooding and the LAR protocols using our calculus. We consider a 80×100 metres area of 35 mobile nodes. We omit the implementation details about how the *Expected Zone* and *Request Zone* are determined according to the specifications of LAR Scheme 1.

We use the following auxiliary functions to simplify the protocol specification:

- $\texttt{gps}$: returns the actual geographical position of the node executing the process (by means, e.g., of GPS technology);
- $\texttt{dist}(l)$: returns the distance from location l and the location of the node executing the process;
- $\texttt{self}$: returns the name (permanent address) of the node executing the process;
- $\texttt{geq}(k, l) = \texttt{true}$ if $k \geq l$, $\texttt{false}$ otherwise;
- $\texttt{inside}(s, A) = \texttt{true}$ if $s \in A$, $\texttt{false}$ otherwise;
- $\texttt{unable}(n) =$ refreshes the route table, removing the existing path to n;
- $\texttt{find\_path}(n) = \texttt{true}$ if there exists a valid path for n in the route table of the node executing the process;
- $\texttt{newBid}$: generates a new unique *Bid* identifier for a packet;
- $\texttt{lastBid}$: returns the latest generated *Bid* identifier;
- $\texttt{control}(Bid) = \texttt{true}$ if the request associated with *Bid* has been already received by the node executing the process.

Each node maintains a *routing table* containing information about the paths to the other nodes in the network. Each entry has the following form:

$$(d, \texttt{seq\#}_d, \texttt{next\_hop}_d, \texttt{hopcount}_d, \texttt{loc}_d, \texttt{v}_d, \texttt{timeout})$$

where d is the destination name, $\texttt{seq\#}_d$ is the sequence number of the route to d, $\texttt{next\_hop}_d$ is the name of the next node to reach d, $\texttt{hopcount}_d$ is the number of hops to reach d, $\texttt{loc}_d$ is the last location known for d, $\texttt{v}_d$ is the average speed of d and $\texttt{timeout}$ is the timeout associated with the entry.

The nodes' *request table* contains the list of all the requests already processed by the node so that loops during the route request forwarding are avoided. For the sake of simplicity, we assume that all the nodes share a common transmission radius $r = 15$ metres.

Let us now consider

$$N = (\nu c)(n[P]_l \mid \textstyle\prod_{i \in I} n_i[Q\_SIMPLE]_{l_i})$$

where a node n (whose location vary among the set $\{16, 23, 30\}$) broadcasts a route request using the simple flooding algorithm to find a path to n_7 (located at 14, as shown in Figure 2 (a)) in the network $\prod_{i \in I} n_i$, and

$$M = (\nu c)(n[P]_l \mid \textstyle\prod_{i \in I} n_i[Q\_LAR1]_{l_i})$$

which is the same network but with nodes in I using the LAR protocol (Scheme 1) instead of the simple flooding algorithm.

Each node n_i moves according to the following matrix J^{n_i}:

$$\begin{array}{ccc} & l_{n_i} & k_{n_i} \\ l_{n_i} & 0.2 & 0.8 \\ k_{n_i} & 0.8 & 0.2 \end{array}$$

where l_{n_i} and k_{n_i} are adjacent locations in the transmission area, as shown by the arrows in Figure 2 (b).

The process executed by node n simply broadcasts a RREQ packet for node n_7 and waits for a RREP packet until a timeout expires. The timeout is modelled using the operator $\oplus$ that behaves as the non-deterministic choice and can be implemented in our calculus by means of the parallel composition and the restriction operator

$$\begin{array}{l}
Q\_X = c(x_1, x_2, x_3, x_4, x_5, x_6, x_7). \\
\quad [x_1 = \texttt{rreq}]([\texttt{control}(x_3) = \texttt{false}]([x_4 = \texttt{self}] \\
\quad \bar{c}_{\texttt{next\_hop}_{x_2},r}\langle(\texttt{rrep}, s, Bid, d, \texttt{seq}\#_s, \texttt{hop\_counter})\rangle.Q\_X, RREQ\_X\langle\tilde{x}\rangle), Q\_X), \\
\quad [x_1 = \texttt{rrep}]([\mathrm{x}_2 = \texttt{self}]\overline{ud}_{\texttt{gps},r}\langle x_2, x_3, x_4, x_5, x_6, x_7\rangle, \\
\quad \bar{c}_{\texttt{next\_hop}_{x_2},r}\langle(\texttt{rrep}, s, Bid, d, \texttt{seq}\#_s, \texttt{hop\_counter})\rangle.Q\_X), \\
\quad [x_1 = \texttt{rerr}]\texttt{unable}(x_4).Q\_X, Q\_X
\end{array}$$

$$\begin{array}{l}
RREQ\_SIMPLE\langle(\texttt{rreq}, s, Bid, d, \texttt{seq}\#_s, \texttt{hop\_counter})\rangle = \\
\quad [\texttt{find\_path}(d) = \texttt{true}]. \\
\quad \bar{c}_{\texttt{next\_hop}_d,r}\langle(\texttt{rrep}, s, Bid, d, \texttt{seq}\#_d, \texttt{hop\_counter} + 1 + \texttt{hopcount}_d, \texttt{timeout})\rangle, \\
\quad \bar{c}_{\mathbf{Loc},r}\langle(\texttt{rreq}, s, Bid, d, \texttt{seq}\#_s, (\texttt{hop\_counter}) + 1)\rangle.Q\_SIMPLE
\end{array}$$

$$\begin{array}{l}
RREQ\_LAR1\langle(\texttt{rreq}, s, Bid, d, \texttt{Request\_Zone}, \texttt{seq}\#_s, \texttt{hop\_counter})\rangle = \\
\quad ([\texttt{inside}(\texttt{gps}, \texttt{Request\_Zone}) = true](\\
\quad [\texttt{find\_path}(d) = \texttt{true}] \\
\quad \bar{c}_{\texttt{next\_hop}_d,r}\langle(\texttt{rrep}, s, Bid, d, \texttt{seq}\#_d, \texttt{hop\_counter} + 1 + \texttt{hopcount}_d, \texttt{timeout})\rangle, \\
\quad \bar{c}_{\texttt{Request\_Zone},r}\langle(\texttt{rreq}, s, Bid, d, \texttt{Request\_Zone}, \texttt{seq}\#_s, (\texttt{hop\_counter}) + 1)\rangle))).Q\_LAR1
\end{array}$$

Table 6: Process specifications used in the case study of Section 5

in the standard way. In case of timeout, a new RREQ is sent. Let

$$\begin{array}{l}
P = \bar{c}_{\mathbf{Loc},r}\langle(\texttt{rreq}, n, \texttt{newBid}, n_7, \texttt{Request\_Zone}, \\
\qquad \texttt{seq}\#_n, 0)\rangle.P'
\end{array}$$

and

$$\begin{array}{l}
P' = P \oplus c(x_1, x_2, x_3, x_4, x_5, x_6, x_7).[x_1 = \texttt{rrep}] \\
\qquad [x_2 = n][x_3 = \texttt{lastBid}] \\
\qquad [x_4 = m][\texttt{geq}(\texttt{hop\_count}_{n_7}, x_7)] \\
\qquad \overline{ok}_{\texttt{gps},r}\langle\texttt{route\_found}\rangle.P'
\end{array}$$

where $x_7 = \texttt{hop\_count}$ in the RREP packet received. Once a route is found, n broadcasts on channel $\texttt{ok}$ a packet that signals this event. Therefore, we consider that the two networks are probabilistically equivalent with respect to their ability to find a route to n_7 if we observe this transmission with the same probability. Notice that, the output on channel c will not be observed by any location because we want to allow the route discovery packets used in the two networks to be arbitrary different.

Hereafter, we use $X \in \{SIMPLE, LAR1\}$ to denote the simple flooding or LAR Scheme 1. The subprocess $RREQ\_SIMPLE$ and the $RREQ\_LAR1$ are defined as shown by Table 6.

In order to compare the behaviour of the protocols, we focus our attention on the following restricted set $\mathcal{F} \subseteq Sched$ of admissible schedulers such that:

1. the timeout for a RREQ identified by Bid occurs when in the networks there are no packets related to Bid;
2. nodes' movements are allowed after every transmission;

Condition 1 on $\mathcal{F}$ is a requirement inherited by the protocol design; the timeout is usually set by knowing the physical dimension of the network. Roughly speaking, we aim at preventing that in the analysis we consider unrealistic schedulers that always choose the timeout option too quickly and hence a route to the destination is never found and those schedulers that wait for an answer indefinitely long. Condition 2 is needed because we do not want to consider those schedulers that never allow for node movements.

The following proposition states that the AODV and LAR protocols are equivalent from a functional point of view. It holds for all networks M and N implementing the LAR and AODV protocols as described adove with arbitrary number of nodes, locations and node distances provided that the DTMC modelling the mobility is ergodic on the set of locations.

Proposition 3 (Functional equivalence of LAR and AODV) *Let M, N, be two networks implementing the LAR and AODV protocols, respectively. Let $\mathcal{M} = \{\bar{M} : M \to^* \bar{M}\} \cup \{\bar{N} : N \to^* \bar{N}\}$ and the set of admissible schedulers $\mathcal{F}$ be defined as above. A sufficient condition for $N \approx_p^{\mathcal{F}} M$ is that the Markov chains $\mathbf{J}^{n_i}$ associated with the mobile nodes n_i $(i \in I)$ are ergodic.*

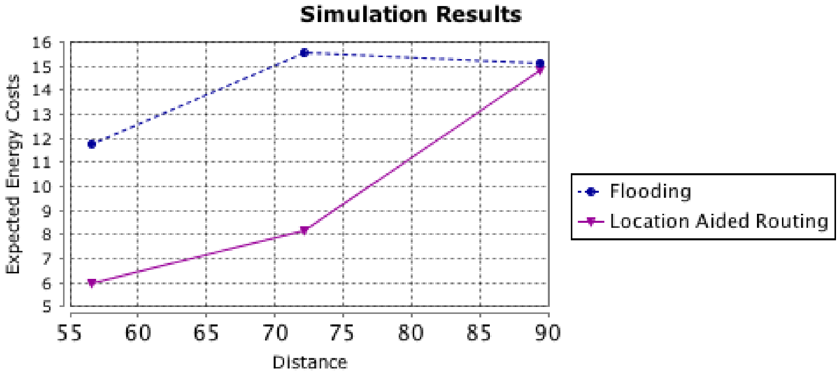
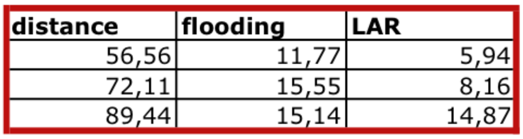


Fig. 3: Plot of the expected energy cost in terms of sent packets per succesfull transmission.



distance	flooding	LAR
56,56	11,77	5,94
72,11	15,55	8,16
89,44	15,14	14,87

Fig. 4: Estimates of the expected energy cost in terms of sent packets per succesfull transmission.

Proof We have to find a relation containing the pair (M, N) that is a probabilistic bisimulation relative to $\mathcal{F}$. Let us consider $Z_i \in \{RREQ, Q\}$, $\bar{P} \in \{P' : P \to^* P'\}$ and

$$\begin{aligned}\mathcal{R} = \{(& n[\bar{P}]_l \mid \textstyle\prod_{i\in I} n_i[Z_i\_SIMPLE]_{l_i}, n[\bar{P}]_l \mid \\ & \textstyle\prod_{i\in I} n_i[Z_i\_LAR1]_{l_i}) : \\ & N \to^* n[\bar{P}]_l \mid \textstyle\prod_{i\in I} n_i[Z_i\_SIMPLE]_{l_i}\}.\end{aligned}$$

In order to prove that $\mathcal{R} \subseteq \approx_p^{\mathcal{F}}$ we have to show that, for all pairs $(\bar{N}, \bar{M}) \in \mathcal{R}$ and 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 $\mathcal{N}/\mathcal{R}$ it holds:

1. if $\alpha \neq c?\tilde{v}@l$ then
   $Prob^F_{\bar{N}}(\xrightarrow{\alpha}, \mathcal{C}) = Prob^{F'}_{\bar{M}}(\overset{\hat{\alpha}}{\Longrightarrow} \mathcal{C})$;
2. if $\alpha = c?\tilde{v}@l$ then either
   $Prob^F_{\bar{N}}(\xrightarrow{\alpha}, \mathcal{C}) = Prob^{F'}_{\bar{M}}(\overset{\alpha}{\Longrightarrow}, \mathcal{C})$ or
   $Prob^F_{\bar{N}}(\xrightarrow{\alpha}, \mathcal{C}) = Prob^{F'}_{\bar{M}}(\Longrightarrow, \mathcal{C})$.

We start from τ actions and consider $\bar{N} \xrightarrow{\tau} [\![\bar{N}']\!]_\theta$. Then, $\forall \mathcal{C} \in \mathcal{N}/\mathcal{R}$, we have:
$Prob_{\bar{N}}(\xrightarrow{\tau}, \mathcal{C}) = \sum_{\hat{N} \in spt([\![\bar{N}']\!]_\theta) \cap \mathcal{C}} [\![\bar{N}']\!]_\theta(\hat{N})$.

If the action is due to the application of rule (Move) we are done, because, for each pair $(\bar{N}, \bar{M}) \in \mathcal{R}$, $\bar{M}$ can perform exactly the same movements as $\bar{N}$, hence there will exists $F' \in \hat{\mathcal{F}}_{\mathcal{C}}$ such that: $Prob^F_{\bar{N}}(\xrightarrow{\tau}, \mathcal{C}) = Prob^{F'}_{\bar{M}}(\xrightarrow{\tau} \mathcal{C})$, and we are done.

If the action is the result of the application of rule (Lose), by applying rule (Bcast) backwardly we get $\bar{N} \xrightarrow{c_K!\tilde{v}[l,r]} [\![\bar{N}']\!]_\Delta$.

If $l \in$ `Request_Zone` then we are done, because, analysing of the process P_LAR1 with respect to the process P_SIMPLE we realize that the protocol packets are forwarded exactly in the same way inside the `RequestZone`.

If $l \notin$ `Request_Zone`, then $\bar{M} \overset{c_K!\tilde{v}[l,r]}{\not\longrightarrow}$ because the routing protocol packets are forwarded only inside the `Request_Zone`. However, this does not mean that $\bar{M}$ will not reach an equivalent state with the same probability. By the initial hypothesis that all the Markov matrices are ergodic, $\bar{M}$ can enter the `Request_Zone` with probability 1, send the message, and come back to the previous location again with probability 1, and we get $Prob^F_{\bar{N}}(\xrightarrow{\tau}, \mathcal{C}) = 1 = Prob^F_{\bar{M}}(\Longrightarrow, \mathcal{C})$ as required.

As concerns the input and the observable actions the proof is trivial, since the input actions are the same for both protocols, and we applied the restriction to channel c, hence the only observable output is the transmission of `route_found` through the channel `ok` by the node n, which behaves in the same way for both protocols. □

Given that the two networks M and N defined at the beginning of this section are functionally equivalent, we compare their energy efficiency by simulation. In order to carry out the simulations we resort to the statistical model checker implemented in PRISM [23]. This technique is commonly used when dealing with models with large state spaces. The simulation model for the PRISM has been automatically generated by the tool introduced in [24].

We have compared the two different networks with the sender node n located in each of the locations in the set $\{16, 23, 30\}$.

The simulations have been performed with an average of 10000 independent experiments, a maximum confidence interval width of 1% of the estimated measure based on 95% of confidence.

The plot (see Figure 3) shows the relation among the distance between sender and receiver and the energy consuption of AODV and LAR expressed in terms of number of sent packets for each succesfull transmission. For larger distances, since a larger *Request Zone* is involved, using LAR protocol still requires a large set of nodes to forward the message, while for smaller distances the improvement brought by the protocol is more evident, since the *Request Zone* is smaller, drastically reducing the number of retransmissions. This supports the intuitive idea that LAR protocol is useful especially in the cases where the expected distance between the sender and the receiver is small.

In Figure 4 we show the numerical comparison between the LAR protocol and the AODV for the considered scenarios.

6 Analysing the SW-ARQ and GBN-ARQ Protocols

In the following we briefly recall the salient features of SW-ARQ and GBN-ARQ protocols. In SW-ARQ protocol, the sender pushes a packet into the channel with a delay that is given by ratio between the packet size and the channel bandwidth (pushing time). Once the packet is in the channel we observe two delays: one is that required to reach the destination and the other one is that required for the acknowledge packet (ACK) to go back to the transmitter. The sum of the two is known as the round trip time. In SW-ARQ protocol the sender sends a packet only once the acknowledge of the previous one has been received. If the round trip time (or an upper bound) is known by the protocol designer, a possible error in the transmission is detected by a timeout mechanism, i.e., if the sender does not receive an ACK from the receiver before a deadline, then it assumes that an error occurred and sends again the same packet. If the round trip time is much higher than the pushing time, then SW-ARQ protocols are very inefficient and exploit only a minimal part of the channel capacity. With respect to SW protocols, GBN takes advantage of the pipelining of the packets, i.e., a sequence of n packets can be sent without receiving any confirmation. This widely used technique is known to highly improve the throughput of the sender, but it is expensive from the energy consumption point of view (see, e.g., [27]) since correctly received packets may be required to be resent. Indeed, once the sender realizes that a packet p has not been received (using a timeout), it has to resend all the packets already sent starting from p. In this way, it can be shown that throughput is really improved and the protocol can use the full channel capacity.

Assumptions on the models. In this case study, we consider a single transmitter node using ARQ-based error recovery protocol to communicate with a receiver node over a wireless channel. Transmissions occur in fixed-size time slots whose size is the time required by the sender to push a packet into the channel. We assume the round trip time to be a multiple of the time slot. For both SW and GBN protocols, the transmitter continuously sends packets until it detects a transmission error. Notice that although in actual implementations of the ARQ protocols errors are usually detected by means of a timeout mechanism, in this context we use negative-acknowledge (NACK) feedbacks which simplify the protocol encoding and are equivalent for the analysis purposes if we assume to know the number of slots that the round trip time consists of. Here, we consider an error-free feedback channel [5] and assume that the ACK or NACK of each transmitted packet arrives at the sender node one slot after the beginning of its transmission slot. Therefore, the feedback of a packet is received exactly after its transmission for the SW-protocol and in case of a failure (NACK), the packet is automatically resent. Instead for the GBN protocol, a feedback for the ith packet arrives exactly after the transmission of the $(i + n - 1)$th packet and in case of a failure the transmission restarts from the ith packet. We model both SW-ARQ and GBN-ARQ-based protocols for a communication channel of capacity $n = 3$ in our framework. Observe that in this way we do not take into account the round trip time for SW-ARQ protocols, however this does not affect the analysis that we will carry out later, i.e., the expected energy cost for each packed correctly received. We consider a unique static receiver $rec < 0, I >$ where I denotes the identity matrix. We model the transmitter as a mobile node $send(< r, J^s >)$ whose reachable locations are l_1, which represents the "good state" of the channel, where the receiver lies within the transmission radius of the channel and l_2 the "bad state", where the destination is no longer reachable (see Figure 5). The mobility of the sender is modelled by the two state Markov chain with the following transition probability matrix

$$J^s = \begin{vmatrix} p & 1-p \\ 1-q & q \end{vmatrix},$$

[5] A very standard assumption [27].

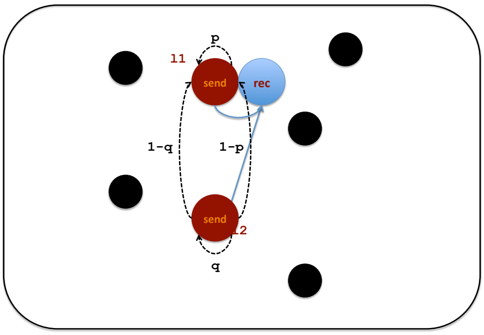


Fig. 5: Topology of the network and mobility of the sender

where p and q are the probabilities of the stability of the node in two successive time slots in its good and bad states, respectively.

Modelling the Protocols In our analysis, we assume that the energy consumption of the feedback messages is negligible. Therefore, they are sent over channels with zero radius. For this reason the static receiver rec is located at l_1, i.e., at the same location of the sender in its good state, so that the feedback will be received with no cost. Note that the sender still transmits over channels with radius r and thus it consumes an amount of energy equal to r for each fired packet.

The process executed by rec, the receiver node, is the same for both protocols and modelled as the process

$$REC\langle i\rangle = c^{(i)}(x).\bar{c}_{l_1,0}\langle ACK(i)\rangle.REC\langle i+1\rangle$$

which, upon receiving packet p_i over the channel $c^{(i)}$, sends $ACK(i)$ over the channel c and waits for the next packet on $c^{(i+1)}$.

For each channel $c^{(i)}$, we use a static auxiliary node $b_i(\langle 0, I\rangle)$ located at l_2, the bad state of the sender, capturing bad transmissions over $c^{(i)}$. It executes the following process which upon receiving packet p_i over the channel $c^{(i)}$, sends $NACK(i)$ over the channel c:

$$BAD\langle i\rangle = c^{(i)}(x).\bar{c}_{l_2,0}\langle NACK(i)\rangle.BAD\langle i\rangle.$$

Now we introduce the full model of the protocol GBN-ARQ.

We start by modelling its sender node. Recall that, as a simplifying assumption, the channel capacity is 3. It executes the following process:

$$\begin{aligned}GB\langle i\rangle = \bar{c}^{(i)}_{l_1,r}\langle p_i\rangle.c(x_1).\bar{c}^{(i+1)}_{l_1,r}\langle p_{i+1}\rangle.c(x_2).\\ \bar{c}^{(i+2)}_{l_1,r}\langle p_{i+2}\rangle.c(x_3)[x_1 = NACK(i)]GB\langle i\rangle,\\ SEND\langle i+3, x_2, x_3\rangle\end{aligned}$$

where the process $SEND$ is defined as follows.

$$\begin{aligned}SEND\langle i, x, y\rangle = \bar{c}^{(i)}_{l_1,r}\langle p_i\rangle.c(z).\\ [x = NACK(i-2)]GB\langle i-2\rangle,\\ SEND\langle i+1, y, z\rangle.\end{aligned}$$

Though that the feedback of a packet is received after the transmission of its two successors, for practical reason, we read a feedback of a packet right after sending it. Indeed, since we do not want feedback to be costly, both sender and receiver must be located at the same place when the feedback is sent. However, the sender node will verify it only after having sent the following two packets.

Recall that the receiver node in our modelling above, reads each packet p_i on its specific channel $c^{(i)}$. Thus, in the GBN, if the transmitter sends p_1 while being in its good state, then moves to bad and sends p_2 and finally moves back to the good state and sends p_3, then the later packet will not be read by the receiver as it is blocked on $c^{(2)}$. Then, the firing on $c^{(3)}$ is lost and this models the fact that packets sent after a bad packet is just a wasting of energy. But since the sender process $GB\langle i\rangle$ is blocked on the feedback channel c, we introduce a static auxiliary node $lose(\langle 0, I\rangle)$ located at l_1 and executing the process:

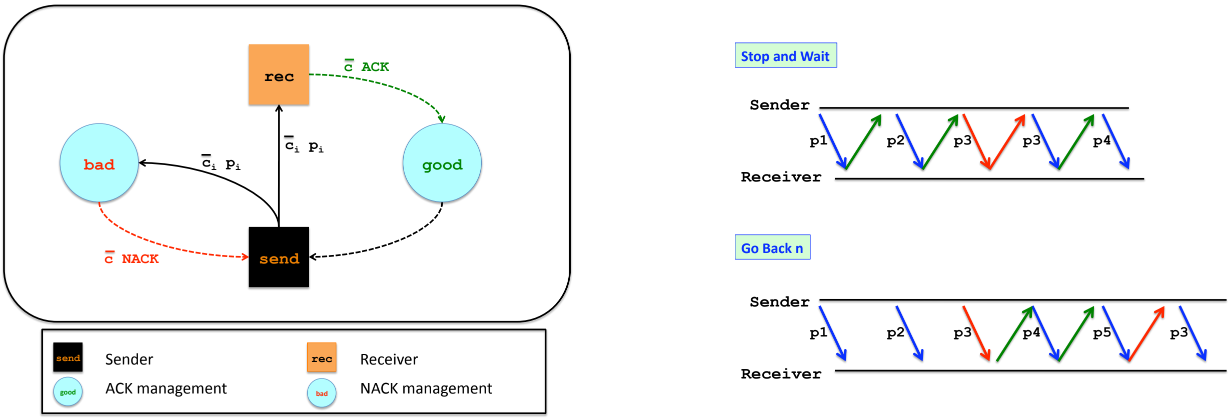
$$WAST = \bar{c}_{\emptyset,0}\langle LOST\rangle.WAST$$

Now on to the SW-ARQ-based protocol. This is very simple since it always sends one packet and waits for its feedback. The sender process is defined as follows.

$$\begin{aligned}SW\langle i\rangle = \bar{c}^{(i)}_{l_1,r}\langle p_i\rangle.c(x).\\ [x = NACK(i)]SW\langle i\rangle, SW\langle i+1\rangle.\end{aligned}$$

The full protocols are then modelled as the network

$$\begin{aligned}GBN = (\nu c^{(1)}, c^{(2)} ...)(send[GB\langle 1\rangle]_{l_1} \mid\\ rec[REC\langle 1\rangle]_{l_1} \mid lose[WAST]_{l_1} \mid\\ \textstyle\prod_{i\geq 1} b_i[BAD\langle i\rangle]_{l_2})\end{aligned}$$



(a) Structure of the communications

(b) Example of GBN and SW behaviour

Fig. 6: Description and example of the network communications

and

$$SW = (\nu c^{(1)}, c^{(2)} ...)(send[SW\langle 1\rangle]_{l_1} \mid rec[REC\langle 1\rangle]_{l_1} \mid \textstyle\prod_{i\in I} b_i[BAD\langle i\rangle]_{l_2}).$$

Measuring the Energy Cost of the Protocols. This section analyzes the energy consumption of the above ARQ-based protocols. In order to compare the observational behaviours of the protocols, we assume that the communications over the feedback channel are observable for any observer node located at l_1. Thus the protocols are equivalent with respect to a set of schedulers $\mathcal{F}$ if for all schedulers F in $\mathcal{F}$ driving one of the protocols, there exists a scheduler F' in $\mathcal{F}$ driving the other one such that both protocols correctly transmit the same packets with the same probabilities. Therefore, we consider the following set of schedulers denoted $\mathcal{F}_{alt}$ which:

1. always alternates between sending packets and node's movement so that at each interaction of the transmitter with the channel, the later can be either good or bad;
2. gives priority to acknowledgment actions (ACK and NACK) to model the standard assumption of an error-free feedback channel;
3. allows interaction with the outside environment only through its observable actions so that we capture exactly the observable behaviour of the protocol.

Notice that the assumptions on the schedulers would be stricter if one desires to carry out an analysis of the throughput. Under these assumptions, we can prove the following results which shows that, the SW-ARQ protocol is more energy efficient of the GBN-ARQ one.

Proposition 4 $GBN \approx_p^{\mathcal{F}_{alt}} SW$.

Proof We give here a sketch of the proof. For each sender's window size we will choose, the only observable actions are the acknowledgments sent by the static node *rec*. All other actions are silent, since we apply the restriction on each $c^{(i)}$. For all $i \geq 1$ $rec[REC\langle i\rangle]_{l_1}$ sends the acknowledgment $ACK(i)$ if and only if the relative packet p_i has been correctly received, hence, all the executions performed by GBN and SW are of the form:

$$\Longrightarrow \xrightarrow{c!ACK(1)@\{l1\}\triangleleft\{l1\}} \Longrightarrow \xrightarrow{c!ACK(2)@\{l1\}\triangleleft\{l1\}} \Longrightarrow \ldots$$

Since the number of transmissions performed by the sender do not affect the probabilities, the bisimulation between the two different protocols can be proved. □

We compare their energy efficiency in the context of the set $\mathcal{H} = \{H_k \mid k \geq 1\}$ where H_k means that all the packets up to k have been correctly transmitted and is defined as $H_k = H_k^1 \cup H_K^2$ where

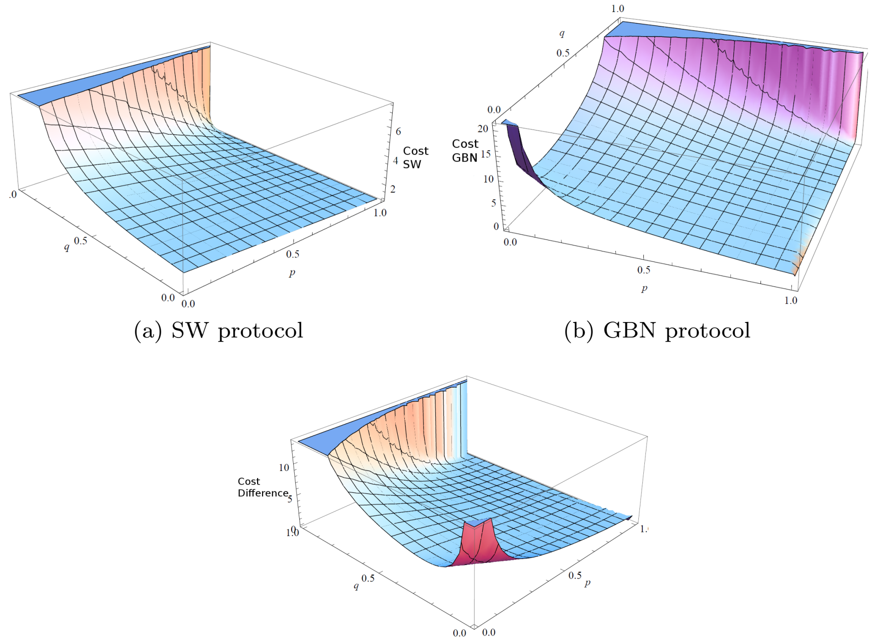
$$H_k^1 = \{M \mid M \equiv send[\bar{c}_{l_1,r}^{(k+1)}\langle p_{k+1}\rangle.P]_{l_1} \mid rec[REC\langle k+1\rangle]_{l_1} \mid loose[WAST]_{l_1} \mid \textstyle\prod_{i\geq 1} b_i[BAD\langle i\rangle]_{l_2}\}$$

for some process P and

$$H_k^2 = \{N \mid N \equiv send[SW\langle i+1\rangle]_{l_1} \mid rec[REC\langle k+1\rangle]_{l_1} \mid \textstyle\prod_{i\in I} b_i[BAD\langle i\rangle]_{l_2}\}.$$

Then, we compute the energy consumption of the protocols assuming that we start by a move action at the good state so that the first message could be lost if it moves to the bad state[6]. The results are summarized

[6] The analysis for the other case is similar.



(a) SW protocol

(b) GBN protocol

(c) $\mathrm{cost}_{GBN}(p,q) - \mathrm{cost}_{SW}(p,q)$

Fig. 7: Energy cost functions for SW and GBN and their comparison.

in the following propositions and illustrated in Figure 7.

Proposition 5 *If $q \neq 1$ then for all $F \in \mathcal{F}_{alt}$,*

$$\mathbf{Cost}^F_{SW}(H_k) = \left(1 + \frac{1-p}{1-q}\right) kr.$$

Proposition 6 *If $q \neq 1$ then for all $F \in \mathcal{F}_{alt}$,*

$$\mathbf{Cost}^F_{GBN}(H_k) = \\ kr\Big(p + \tfrac{(p-1)}{(-1+q)(1+p^2-q+q^2-p+2pq)} \cdot \\ \tfrac{1-2p^2+2p^2q+4q-4q^2+2q^3+2p-6pq+4pq^2}{-p^2+p^2+(-p+pq)(-1+2q)+q(2+-2q+q^2)}\Big).$$

These results can be derived by applying the Chapman-Kolmogorv's forward equations to compute the probability of consecutive failures in the sending of the same packet. Each of these failures (except the first) causes the waste of a number of sent packets equals to the window size. It can be observed that the number of wasted windows has a geometric distribution. Then, the mean of total packets sent to obtain a success, can be straightforwardly derived.

To conclude this section, we note that while both protocols increasingly enjoy bad performance in term of energy consumption when the channel deteriorates, i.e., when q is increasing (see Figures 7-(a) and 7-(b)), the GBN protocol deteriorates faster. Indeed, as illustrated by Figure 7-(c) as the channel deteriorates the additional energy required by GBN protocol to correctly transmit the same number of packets increases to infinite. Thus, the gain of having a high throughput results in a very high energy consumption.

Finally we can conclude that the GBN protocol is much more energy consuming than SW.

Theorem 5 *It holds that $SW \sqsubseteq_{\langle \mathcal{H}, \mathcal{F}_{alt} \rangle} GBN$.*

Proof The proof follows straightforwardly from Propositions 4, 5 and 6.

7 Conclusion

Ad-hoc networks is a new area of mobile communication networks that has attracted significant attention due to its challenging problems. The main goal of our work is to provide a formal model to reason about the problem of limiting the power consumption of communications while maintaining acceptable performances. Indeed, one of the most critical challenges in managing mobile ad-hoc networks is actually to find a good trade-off between network connectivity and power saving.

Even though not all the devices have the ability of adjusting their transmission power, modern technologies are quickly evolving, and there exist devices that are enabled to choose among two or more different power levels. For this reason many researches have proposed algorithms and protocols with the aim of providing a way to decide the best transmission power for node communications in a given network [7,37], or to develop energy-aware routing protocols [10,16].

In this paper, we presented the Probabilistic EBUM calculus which, due to its characteristics of modelling broadcast, multicast and unicast communications and also modelling the ability of a node to change its transmission power in accordance with the protocol it is executing, results to be a valid formal model for the analysis, evaluation and comparison of energy-aware protocols and algorithms specifically developed for wireless ad-hoc networks. The model we presented can clearly be extended with different metrics for measuring, e.g., the level of interference or the number of collisions and losses. Moreover, it provides a basis for the definition of other verification techniques, like e.g., bisimulation-based preorders, in the style of [18], which integrate both observational properties and quantitative ones.

Acknowledgements Work partially supported by the Italian MIUR - PRIN Project *CINA: Compositionality, Interaction, Negotiation and Autonomicity.*

References

1. .Abadi, M., Fournet, C.: Mobile Values, New Names, and Secure Communication. SIGPLAN Not. **36**(3), 104–115 (2001)
2. Acquaviva, A., Aldini, A., Bernardo, M., Bogliolo, A., Bontà, E., Lattanzi, E.: A methodology based on formal methods for predicting the impact of dynamic power management. In: Formal Methods for Mobile Computing, *lncs*, vol. 3465, pp. 51–58. Springer Berlin / Heidelberg (2005)
3. Beccuti, M., Pierro, M.D., Horvàth, A., Horvàth, A., Farkas, K.: A Mean Field Based Methodology for Modeling Mobility in Ad Hoc Networks. In: Proc. of 73rd IEEE Vehicular Technology Conference (VTC Spring), pp. 1–5. IEEE, Budapest, HU (2011)
4. Bernardo, M., Bravetti, M.: Performance Measure Sensitive Congruences for Markovian Process Algebras. Theoretical Computer Science **290**(1), 117–160 (2003)
5. Bugliesi, M., Gallina, L., Hamadou, S., A., M., Rossi, S.: Behavioral equivalences and interference metrics for mobile ad-hoc networks. Performance Evaluation **73**, 41–72 (2014)
6. Burkhart, M., von Rickenbach, P., Wattenhofer, R., Zollinger, A.: Does Topology Control Reduce Interference? In: Proc. of the 5th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc'04), pp. 9–19. ACM (2004)
7. Calamoneri, T., Clementi, A., Monti, A., Rossi, G., Silvestri, R.: Minimum-energy broadcast in random-grid ad-hoc networks: Approximation and distributed algorithms. In: Proc. of the 11th International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM '08), pp. 354–361. ACM (2008)
8. Cerone, A., Hennessy, M.: Modelling Probabilistic Wireless Networks. Logical Methods in Computer Science **9**(3), 1 – 68 (2013)
9. De Nicola, R., Katoen, J.P., Latella, D., Massink, M.: STOKLAIM: A Stochastic Extension of KLAIM. Tech. Rep. 2006-TR-01, ISTI (2006)
10. Ferrari, G., Malvassori, S.A., Bragalini, M., Tonguz, O.: Physical layer-constrained routing in ad-hoc wireless networks: a modified aodv protocol with power control. In: Proc. of the International Workshop on Wireless Ad-hoc Networks (IWWAN'05) (2005)
11. Gallina, L., Hamadou, S., Marin, A., Rossi, S.: 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), *LNCS*, vol. 6751, pp. 316–330. Springer-Verlag (2011)
12. Gallina, L., Rossi, S.: A Calculus for Power-aware Multicast Communications in Ad-hoc Networks. In: Proc. of the 6th IFIP International Conference on Theoretical Computer Science (TCS'10), pp. 20–31. Springer (2010)
13. Gallina, L., Rossi, S.: Sender- and Receiver-centered Interference in Wireless ad-hoc Networks. In: Proc. of IFIP Wireless Days 2010. IEEE (2010)
14. Gallina, L., Rossi, S.: A Process Calculus for Energy-aware Multicast Communications of Mobile ad-hoc Networks. Wireless Communications and Mobile Computing (2012)
15. Gilmore, S., Hillston, J.: The PEPA Workbench: A Tool to Support a Process Algebra-based Approach to Performance Modelling. In: Computer Performance Evaluation Modelling Techniques and Tools, *lncs*, vol. 794, pp. 353–368. Springer Berlin / Heidelberg (1994)
16. Gomez, J., Campbell, A.T.: Variable-range transmission power control in wireless multihop networks. IEEE Transactions on Mobile Computing (TMC) **6**(1), 87–99 (2007)
17. Goubault-Larrecq, J., Palamidessi, C., Troina, A.: A Probabilistic Applied Pi-Calculus. In: Proc. of the 5th Asian Symposium on Programming Languages and Systems (APLAS '07), *LNCS*, vol. 4807/2009, pp. 175–190. Springer-Verlag (2007)
18. Hennessy, M.: A calculus for costed computations. Logical Methods in Computer Science **7**(1) (2011)
19. Hillston, J.: A Compositional Approach to Performance Modelling. Distinguished Dissertations in Computer Science. Cambridge University Press (2005)
20. Johnson, D.B., Maltz, D.: Dynamic Source Routing in Ad hoc Wireless Networks. In: Mobile Computing, *The Kluwer International Series in Engineering and Computer Science*, vol. 353, pp. 153–181. Springer US (1996)
21. Kaplan, E.: Understanding GPS: Principles and Applications. Artech House Publishing (1996)
22. Ko, Y., Vaidya, N.: Locationaided Routing (LAR) in Mobile Ad hoc Networks. Wireless Networks **6**, 307–321 (2000)
23. Kwiatkowska, M.Z., Norman, G., Parker, D.: PRISM 4.0: Verification of Probabilistic Real-time Systems. In: CAV, pp. 585–591 (2011)
24. L. Gallina, T.H., Kwiatkowska, M., Marin, A., Rossi, S., Spanò, A.: Automatic Energy-aware Performance Analysis of Mobile Ad-hoc Networks. In: Proc. of IFIP Wireless Days Conference (WD'12). IEEE Press (2012)
25. Lanese, I., Sangiorgi, D.: An Operational Semantics for a Calculus for Wireless Systems. Theoretical Computer Science **411**(19), 1928–1948 (2010)
26. Lanotte, R., Merro, M.: Semantic Analysis of Gossip Protocols for Wireless Sensor Networks. In: CONCUR 2011 Concurrency Theory, *lncs*, vol. 6901, pp. 156–170. Springer Berlin / Heidelberg (2011)

27. Le, L., Hossain, E., Zorzi, M.: Queueing Analysis for GBN and SR ARQ Protocols under Dynamic Radio Link Adaptation with Non-zero Feedback Delay. IEEE Transactions on Wireless Communications **6**(9), 3418–3428 (2007)
28. Macedonio, D., Merro, M.: A Semantic Analysis of Wireless Network Security Protocols. In: NASA Formal Methods, *lncs*, vol. 7226, pp. 403–417. Springer Berlin / Heidelberg (2012)
29. Merro, M.: An Observational Theory for Mobile Ad Hoc Networks. Information and Computation **207**(2), 194–208 (2009)
30. Milner, R., Sangiorgi, D.: Barbed Bisimulation. In: Proc. of International Colloquium on Automata, Languages and Programming (ICALP '92), *LNCS*, vol. 623, pp. 685–695. Springer-Verlag (1992)
31. Mohimani, G., Ashtiani, F., Javanmard, A., Hamdi, M.: Mobility Modeling, Spatial Traffic Distribution, and Probability of Connectivity for Sparse and Dense Vehicular Ad Hoc Networks. IEEE Trans. on Vehicular Technology **58**(4) (May, 2009)
32. Murthy, S., Garcia-Luna-Aceves, J.: An Efficient Routing Protocol for Wireless Networks. Mobile Networks and Applications **1**, 183–197 (1996)
33. Park, V., Corson, M.: A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks. In: Proc. of the 16th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '97), vol. 3, pp. 1405–1413. ieee (1997)
34. Perkins, C., Bhagwat, P.: Highly dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. In: Proc. of the Conference on Communications architectures, protocols and applications, SIGCOMM '94, pp. 234–244. ACM, New York, NY, USA (1994)
35. Ross, S.M.: Stochastic Processes, 2nd edn. John Wiley & Sons (1996)
36. Royer, E.M., Perkins, C.E.: Multicast Operation of the Ad-hoc On-demand Distance Vector Routing Protocol. In: Proc. of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking, pp. 207–218. ACM (1999)
37. Sanchez, M., Manzoni, P., Haas, Z.J.: Determination of critical transmission range in ad-hoc networks. In: Proc. of the Multiaccess, Mobility and Teletraffic for Wireless Communications Conference (MMT '99) (1999)
38. Sarkar, S., Majumder, K.: A survey on power aware routing protocols for mobile ad-hoc network. In: Proceedings of the International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2013, *Advances in Intelligent Systems and Computing*, vol. 247, pp. 313–320. Springer (2014)
39. Segala, R., Lynch, N.: Probabilistic Simulations for Probabilistic Processes. In: Proc. of the 5th International Conference on Concurrency Theory (CONCUR '94), *LNCS*, vol. 836, pp. 481–496. Springer-Verlag (1994)
40. Singh, S., Woo, M., Raghavendra, C.: 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), pp. 181–190. ACM (1998)
41. Song, L., Godskesen, J.: Probabilistic mobility models for mobile and wireless networks. In: Proc. the 6th IFIP TC 1/WG 202 international conference on Theoretical Computer Science (TCS'10), *IFIP Advances in Information and Communication Technology*, vol. 323, pp. 86–100. Springer Boston (2010)
42. Song, L., Godskesen, J.: Broadcast abstraction in a stochastic calculus for mobile networks. In: Proc. the 7th IFIP TC 1/WG 202 international conference on Theoretical Computer Science (TCS'12), *lncs*, vol. 7604, pp. 342–356. Springer-Verlag (2012)
43. Stojmenovic, I., Lin, X.: Power-aware Localized Routing in Wireless Networks. IEEE Transactions on Parallel and Distributed Systems **12**(11), 1122–1133 (2001)
44. Tanenbaum, A.S.: Computer Networks. Prentice-Hall (2003)
45. Wattenhofer, R., Li, L., Bahl, P., Wang, Y.: Distributed Topology Control for Power Efficient Operation in Multihop Wireless Ad hoc Networks. In: Proc. of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'01), vol. 3, pp. 1388–1397. ieee (2001)
46. Zorzi, M., Rao, R.: Error Control and Energy Consumption in Communications for Nomadic Computing. IEEE Transactions on Computers **46**(3), 279–289 (1997)

Appendix

This supplement contains the proofs of some of the results presented in the paper.

Proof of Theorem 1

1. The first part is proved by induction on the reduction $M \longrightarrow [\![M']\!]_\theta$.
   Suppose that $M \longrightarrow [\![M']\!]_\theta$ is due to the application of the rule (R-Move). It means that $M \equiv M' \equiv n[P]_l$, for some name n, location l, some (possibly empty) process P, and $\theta = \mu_l^n$. We simply apply (Move) to obtain:
   $$\frac{}{n[P]_l \xrightarrow{\tau} [\![n[P]_l]\!]_{\mu_l^n}}.$$
   Suppose that $M \longrightarrow [\![M']\!]_\theta$ is due to the application of the rule (R-Par) with $M \equiv M_1 \mid M_2$, $M' \equiv M_1' \mid M_2$ and:
   $$\frac{M_1 \longrightarrow [\![M_1']\!]_\theta}{M_1 \mid M_2 \longrightarrow [\![M_1' \mid M_2]\!]_\theta}.$$
   By induction hypothesis there exist $N \equiv M_1$ and $N' \equiv M_1'$ such that $N \xrightarrow{\tau} [\![N']\!]_\theta$, then by applying rule (Par) we get:
   $$\frac{N \xrightarrow{\tau} [\![N']\!]_\theta}{N \mid M_2 \xrightarrow{\tau} [\![N' \mid M_2]\!]_\theta},$$
   and the statement follows since by applying the rules of structural congruence we have $N \mid M_2 \equiv M_1 \mid M_2 \equiv M$ and $N' \mid M_2 \equiv M_1' \mid M_2 \equiv M'$.
   Suppose that $M \longrightarrow [\![M']\!]_\theta$ is due to the application of the rule (R-Res) with $M \equiv (\nu c)M_1$ and $M' \equiv (\nu c)M_1'$ for some channel c and some networks M_1 and M_1', then
   $$\frac{M_1 \longrightarrow [\![M_1']\!]_\theta}{(\nu c)M_1 \longrightarrow [\![(\nu c)M_1']\!]_\theta}.$$
   By induction hypothesis there exist $N \equiv M_1$ and $N' \equiv M_1'$ such that $N \xrightarrow{\tau} [\![N']\!]_\theta$, then by applying rule (Res), since $\mathtt{Chan}(\tau) \neq c$ we get:
   $$\frac{N \xrightarrow{\tau} [\![N']\!]_\theta}{(\nu c)N \xrightarrow{\tau} [\![(\nu c)N']\!]_\theta},$$

and the statement follows since by applying the rules of structural congruence we have $(\nu c)N \equiv (\nu c)M_1 \equiv M$ and $(\nu c)N' \equiv (\nu c)M_1' \equiv M'$.
Suppose that $M \longrightarrow \llbracket M' \rrbracket_\theta$ is due to the application of the rule (R-Bcast). Then
$M \equiv n[\bar{c}_{L,r}\langle\tilde{v}\rangle.P]_l \mid \prod_{i\in I} n_i[c(\tilde{x}_i).P_i]_{l_i}$,
$M' \equiv n[P]_l \mid \prod_{i\in I} n_i[P_i\{\tilde{v}/\tilde{x}_i\}]_{l_i}$
for some name n, channel c, location l, radius r, some set L of locations, some tuple $\tilde{v}$ of messages, some (possibly empty) process P, some (possibly empty) set I of networks. By applying the rules (Snd), (Rcv), $\mid I \mid$ times the rule (Bcast) and, finally the rule (Lose), we obtain
$n[\bar{c}_{L,r}\langle\tilde{v}\rangle.P]_l \mid \prod_{i\in I} n_i[c(\tilde{x}_i).P_i]_{l_i}$
$\xrightarrow{\tau}\llbracket n[P]_l \mid \prod_{i\in I} n_i[P_i\{\tilde{v}/\tilde{x}_i\}]_{l_i}\rrbracket_\Delta$,
as required.
Finally suppose that the reduction $M \longrightarrow \llbracket M' \rrbracket_\theta$ is due to an application of rule (R-Struct):

$$\frac{M \equiv N \quad N {\longrightarrow} \llbracket N' \rrbracket_\theta \quad N' \equiv M'}{M {\longrightarrow} \llbracket M' \rrbracket_\theta}.$$

By induction hypothesis there exist $N_1 \equiv N$ and $N_2 \equiv N'$ such that $N_1 \xrightarrow{\tau} \llbracket N_2 \rrbracket_\theta$. The statement follows since by applying the rules of the structural congruence we have $M \equiv N \equiv N_1$ and $M' \equiv N' \equiv N_2$.

2. The second part of the theorem follows straightforwardly from Lemma 1 and the definition of Barb.
   $\Rightarrow$ If $M \downarrow_{c@K}$, by the definition of Barb:
   $M \equiv (\nu\tilde{d})(n[\bar{c}_{L,r}\langle\tilde{v}\rangle.P]_l \mid M_1)$, for some n, $\tilde{v}$, L, r, some (possibly empty) sequence $\tilde{d}$ with $c \notin \tilde{d}$, some process P and some (possibly empty) network M_1, with $K \subseteq \{k \in L$ such that $d(l,k) \leq r\}$ and $K \neq \emptyset$.
   By applying the rules (Snd), (Par) and (Res):

$$\frac{n[\bar{c}_{L,r}\langle\tilde{v}\rangle.P]_l \xrightarrow{c_L!\tilde{v}[l,r]} \llbracket n[P]_l \rrbracket_\Delta}{M \xrightarrow{c_L!\tilde{v}[l,r]} \llbracket (\nu\tilde{d})(n[P]_l \mid M_1\rrbracket_\Delta)};$$

   then we can apply rule (Obs):
   $n[\bar{c}_{L,r}\langle\tilde{v}\rangle.P]_l \mid M_1 \xrightarrow{c!\tilde{v}@K\triangleleft R}$
   $\llbracket n[P]_l \mid M_1 \rrbracket_\Delta$,
   where $R = \{l' \in \mathit{Loc} : d(l,l') \leq r\}$, and $K \subseteq L \cap R$ as required.
   $\Leftarrow$ If $M \xrightarrow{c!\tilde{v}@K\triangleleft R} \llbracket M' \rrbracket_\Delta$, because $M \xrightarrow{c_L!\tilde{v}![l,r]} \llbracket M' \rrbracket_\Delta$, by applying Lemma 1 there exist n, some (possibly empty) sequence $\tilde{d}$ such that $c \notin \tilde{d}$, some process P, some (possibly empty) network M_1 and a set I, such that $\forall i \in I$ with $d(l,l_i) \leq r$:
   $M \equiv (\nu\tilde{d})(n[\bar{c}_{L,r}\langle\tilde{v}\rangle.P]_l|$
   $\prod_{i\in I} n_i[c(\tilde{x}_i).P_i]_{l_i} \mid M_1)$
   and
   $M' \equiv (\nu\tilde{d})(n[P]_l|$
   $\prod_{i\in I} n_i[P_i\{\tilde{v}/\tilde{x}_i\}]_{l_i} \mid M_1)$.
   Since $K \neq \emptyset$, by applying the definition of barb we conclude $M \downarrow_{c@K}$.
3. The third part of the theorem is proved by induction on the derivation $M \xrightarrow{\tau} \llbracket M' \rrbracket_\theta$.
   Suppose that $M \xrightarrow{\tau} \llbracket M' \rrbracket_\theta$ is due to an application of the rule (Move), i.e., $M \equiv n[P]_l$, $M' \equiv n[P]_l$, for some name n, some (possibly empty) process P, some location l, $\theta = \mu_l^n$ and

$$\frac{}{n[P]_l \xrightarrow{\tau} \llbracket n[P]_l \rrbracket_{\mu_l^n}},$$

   hence , by applying (R-Move) we get:

$$\frac{}{n[P]_l {\longrightarrow} \llbracket n[P]_l \rrbracket_{\mu_l^n}}.$$

   If $M \xrightarrow{\tau} \llbracket M' \rrbracket_\theta$ is due to an application of (Lose):

$$\frac{M \xrightarrow{c_L!\tilde{v}[l,r]} \llbracket M' \rrbracket_\Delta}{M {\xrightarrow{\tau}} \llbracket M' \rrbracket_\Delta},$$

   for some channel c, some set L of locations, some tuple $\tilde{v}$ of messages, some location l and radius r. By applying Lemma 1, there exist n, $\tilde{v}$, a (possibly empty) sequence $\tilde{d}$ such that $c \notin \tilde{d}$, a process P, a (possibly empty) network M_1 and a (possibly empty) set I with $d(l,l_i) \leq r$ $\forall i \in I$ such that:
   $M \equiv (\nu\tilde{d})(n[\bar{c}_{L,r}\langle\tilde{v}\rangle.P]_l|$
   $\prod_{i\in I} n_i[c(\tilde{x}_i).P_i]_{l_i} \mid M_1)$
   and
   $M' \equiv (\nu\tilde{d})(n[\bar{c}_{L,r}\langle\tilde{v}\rangle.P]_l|$
   $\prod_{i\in i} n_i[P_i\{\tilde{v}/\tilde{x}_i\}]_{l_i} \mid M_1)$.
   Finally, by applying rules (R-Bcast), (R-Res) and (R-Struct) we get $M \longrightarrow \llbracket M' \rrbracket_\theta$.
   Suppose that $M \xrightarrow{\tau} \llbracket M' \rrbracket_\theta$ is due to the application of (Res) with $M \equiv (\nu c)M_1$ and $M' \equiv (\nu c)\llbracket M_1' \rrbracket_\theta$, for some channel c and for some networks M_1 and M_1'. Then we have:

$$\frac{M_1 \xrightarrow{\tau} \llbracket M_1' \rrbracket_\theta}{(\nu c)M_1 \xrightarrow{\tau} \llbracket (\nu c)M_1' \rrbracket_\theta}.$$

   By induction hypothesis $M_1 \longrightarrow \llbracket M_1' \rrbracket_\theta$, hence, by applying rule (R-Res) we get $(\nu c)M_1 \longrightarrow \llbracket (\nu c)M_1' \rrbracket_\theta$.
   Finally, suppose that $M \xrightarrow{\tau} \llbracket M' \rrbracket_\theta$ is due to the application of (Par) with $M \equiv M_1 \mid M_2$, $M' \equiv M_1' \mid M_2$ and

$$\frac{M_1 \xrightarrow{\tau} \llbracket M_1' \rrbracket_\theta}{M_1|M_2 \xrightarrow{\tau} \llbracket M_1'|M_2 \rrbracket_\theta}.$$

   By induction hypothesis $M_1 \longrightarrow \llbracket M_1' \rrbracket_\theta$, hence, by applying rule (R-Par) we get $M_1|M_2 \longrightarrow \llbracket M_1'|M_2 \rrbracket_\theta$.
4. The last part of the theorem follows from the definition of barb and Lemma 1. Formally, since $M \xrightarrow{c!\tilde{v}@K\triangleleft R} \llbracket M' \rrbracket_\Delta$ because $M \xrightarrow{c_L!\tilde{v}[l,r]} \llbracket M' \rrbracket_\Delta$ for some location l, radius r and set L of intended recipients, by applying Lemma 1, there exist n, a (possibly empty) sequence $\tilde{d}$ with $c \notin \tilde{d}$, a process P, a (possibly empty) network M_1 and a (possibly empty) set I such that:
   $M \equiv (\nu\tilde{d})(n[\bar{c}_{L,r}\langle\tilde{v}\rangle.P]_l \mid$
   $\prod_{i\in I} n_i[c(\tilde{x}_i).P_i]_{l_i} \mid M_1)$
   and
   $M' \equiv (\nu\tilde{d})(n[P]_l \mid \prod_{i\in I} n_i[P_i\{\tilde{v}/\tilde{x}_i\}]_{l_i} \mid M_1)$.
   Then, by applying the rules (R-Bcast), (R-Par) and (R-Res) we get:
   $(\nu\tilde{d})(n[\bar{c}_{L,r}\langle\tilde{v}\rangle.P]_l \mid \prod_{i\in I} n_i[c(\tilde{x}_i).P_i]_{l_i} \mid M_1)$
   $\longrightarrow \llbracket (\nu\tilde{d})(n[P]_l \mid \prod_{i\in I} n_i[P_i\{\tilde{v}/\tilde{x}_i\}]_{l_i} \mid M_1)\rrbracket_\Delta$,
   and, by applying (R-Struct), we obtain $M \longrightarrow \llbracket M' \rrbracket_\Delta$, as required.

□

Proof of Theorem 2

We have to prove that $\approx_p^{\mathcal{F}}$ is:

1. probabilistic barb preserving
2. reduction closed
3. contextual.

1. To prove that the probabilistic labelled bisimilarity $\approx_p^{\mathcal{F}}$ is *barb preserving* we have to show that if $M \approx_p^{\mathcal{F}} N$ then, for each scheduler $F \in \mathcal{F}_{\mathcal{C}}$, for each channel c and for each set K of locations such that $M \Downarrow_p^F c@K$, there exists $F' \in \mathcal{F}_{\mathcal{C}}$ such that $N \Downarrow_p^{F'} c@K$.

Assume that $M \Downarrow_p^F c@K$ for some $F \in \mathcal{F}_{\mathcal{C}}$. Then, by Definition 3 we have $Prob^F_M(H) = p$, where $H = \{M' : M' \downarrow_{c@K}\}$. We can partition H into a set of equivalence classes with respect to $\approx_p^{\mathcal{F}}$. Formally, $\exists J$ such that $H \subseteq \cup_{j \in J} \mathcal{C}_j$, and $\forall j \in J$ we have $\mathcal{C}_j \in \mathcal{N}/ \cong_p^{\mathcal{F}}$ and $H \cap \mathcal{C}_j \neq \emptyset$. Hence:

$Prob^F_M(H) = \sum_{e \in Exec^F_M(H)} P^F_M(e) = \sum_{j \in J} Prob^F_M(\mathcal{C}_j) = p$.

By Theorem 1 and by Definition 8 there exists $\hat{F} \in \hat{\mathcal{F}}_{\mathcal{C}}$ such that $\forall j \in J$:

$Prob^F_M(\mathcal{C}_j) = Prob^{\hat{F}}_M(\Longrightarrow, \mathcal{C}'_j)$

where $\mathcal{C}'_j = \mathcal{C}_j \cup \{\hat{M} \mid \exists \hat{M}' \in \mathcal{C}_j \text{ and } \hat{M} \equiv \hat{M}'\}$.

Now, since $\forall \hat{M}$ such that $\hat{M} \equiv \hat{M}' \in \mathcal{C}_j$, by applying rule (R-Struct) and by Definition 4 $\hat{M} \cong_p^{\mathcal{F}} \hat{M}'$, we get $\{\hat{M} : \hat{M} \equiv \hat{M}' \in \mathcal{C}_j\} \subseteq \mathcal{C}_j$, that means $\mathcal{C}'_j = \mathcal{C}_j$ $\forall j \in J$. Hence we get:

$\sum_{j \in J} Prob^F_M(\mathcal{C}_j) = \sum_{j \in J} Prob^{\hat{F}}_M(\Longrightarrow, \mathcal{C}_j)$.

Since $M \approx_p^{\mathcal{F}} N$, there exists $\hat{F}' \in \hat{\mathcal{F}}_{\mathcal{C}}$ such that, by Proposition 2, for all $j \in J$:

$Prob^{\hat{F}}_M(\Longrightarrow, \mathcal{C}_j) = Prob^{\hat{F}'}_N(\Longrightarrow, \mathcal{C}_j)$.

We then have:

$p = \sum_{j \in J} Prob^{\hat{F}'}_N(\Longrightarrow, \mathcal{C}_j)$.

Again, by Theorem 1, Proposition 2 and Definition 4, there exists $F' \in \mathcal{F}_{\mathcal{C}}$ such that for all $j \in J$:

$Prob^{\hat{F}'}_N(\Longrightarrow, \mathcal{C}_j) = Prob^{F'}_N(\mathcal{C}_j)$ and

$p = \sum_{j \in J} Prob^{\hat{F}'}_N(\Longrightarrow, \mathcal{C}_j) = \sum_{i \in J} Prob^{F'}_N(\mathcal{C}_j) = Prob^{F'}_N(H)$,

i.e., $N \Downarrow_p^{F'} c@K$ as required.

2. To prove that probabilistic labelled bisimilarity $\approx_p^{\mathcal{F}}$ is reduction closed, we have to show that if $M \approx_p^{\mathcal{F}} N$, then for all $F \in \mathcal{F}_{\mathcal{C}}$, there exists $F' \in \mathcal{F}_{\mathcal{C}}$ such that for all classes $\mathcal{C} \in \mathcal{N}/ \cong_p^{\mathcal{F}}$, $Prob^F_M(\mathcal{C}) = Prob^{F'}_N(\mathcal{C})$.

By Theorem 1 and by Definition 8 we deduce that $\exists \hat{F} \in \hat{\mathcal{F}}_{\mathcal{C}}$ such that $Prob^F_M(\mathcal{C}) = Prob^{\hat{F}}_M(\Longrightarrow, \mathcal{C}')$, where $\mathcal{C}' = \mathcal{C} \cup \{\hat{M} : \hat{M} \equiv \hat{M}' \in \mathcal{C}\}$, but since $\forall \hat{M}$ such that $\hat{M} \equiv \hat{M}' \in \mathcal{C}$, by applying rule (R-Struct) and by Definition 4 $\hat{M} \cong_p^{\mathcal{F}} \hat{M}'$ we get $\{\hat{M} : \hat{M} \equiv \hat{M}' \in \mathcal{C}\} \subseteq \mathcal{C}$, i.e., $\mathcal{C}' = \mathcal{C}$.

By Proposition 2 we have that $\exists \hat{F}' \in \hat{\mathcal{F}}_{\mathcal{C}}$ such that $Prob^{\hat{F}}_M(\Longrightarrow, \mathcal{C}) = Prob^{\hat{F}'}_N(\Longrightarrow, \mathcal{C})$.

Finally, by Theorem 1 and by Definitions 8 and 4, $\exists F' \in \mathcal{F}_{\mathcal{C}}$ such that $Prob^{\hat{F}'}_N(\Longrightarrow, \mathcal{C}) = Prob^{F'}_N(\mathcal{C})$, as required.

3. In order to prove that probabilistic labelled bisimilarity $\approx_p^{\mathcal{F}}$ is contextual we have to prove that, if $M \approx_p^{\mathcal{F}} N$:

1. $M \mid O \approx_p^{\mathcal{F}} N \mid O$ $\forall O \in \mathcal{N}$.
2. $(\nu d)M \approx_p^{\mathcal{F}} (\nu d)N$ $\forall d \in \mathbf{C}$.

Case 1.

Let us consider the relation

$$\mathcal{R} = \{(M \mid O, N \mid O) : M \approx_p^{\mathcal{F}} N\}.$$

We prove that for all scheduler $F \in \hat{\mathcal{F}}_{\mathcal{C}}$ there exists a scheduler $F' \in \hat{\mathcal{F}}_{\mathcal{C}}$ such that for all α and for all classes $\mathcal{C}$ in $\mathcal{N}/\approx_p^{\mathcal{F}}$:

1. if $\alpha = \tau$ then
   $Prob^F_{M|O}(\xrightarrow{\tau}, \mathcal{C}) = Prob^{F'}_{N|O}(\Longrightarrow, \mathcal{C})$.
   If $P, Q \in \mathcal{C}$, then, by definition of $\mathcal{R}$, $P \equiv \bar{P} \mid \bar{O}$, $Q \equiv \bar{Q} \mid \bar{O}$ and $\bar{P} \approx_p^{\mathcal{F}} \bar{Q}$. But then there exists $\mathcal{D} \in \mathcal{N}/ \approx_p^{\mathcal{F}}$ such that $\mathcal{D} = \{\bar{P} : \bar{P} \mid \bar{O} \in \mathcal{C}\}$. Now we have three cases to consider:

(i) if $M \mid O \xrightarrow{\tau} [\![M \mid O']\!]_\theta$ because $O \xrightarrow{\tau} [\![O']\!]_\theta$ the proof is simple, because for all $\bar{M}$ in the support of $[\![M \mid O']\!]_\theta$ such that $\bar{M} \in \mathcal{C}$, it holds that $\bar{M} \equiv M \mid O''$ and, since $M \approx_p^{\mathcal{F}} N$, $N \mid O'' \in \mathcal{C}$ too, by definition of $\mathcal{R}$. By Definition 4 there exists $\bar{F} \in \mathcal{F}_{\mathcal{C}}$ such that, by applying rule (R-Par) to the reduction $O \to [\![O']\!]_\theta$, $N \mid O \to [\![O' \mid N]\!]_\theta \in Exec^{\bar{F}}_{N|O}$. By Theorem 1 and by Definition 8 $\exists F' \in \hat{\mathcal{F}}_{\mathcal{C}}$ such that $Prob^{\bar{F}}_{N|O}(\mathcal{C}) = Prob^{F'}_{N|O}(\Longrightarrow, \mathcal{C})$, hence $Prob^F_{M|O}(\xrightarrow{\tau}, \mathcal{C}) = Prob^{F'}_{N|O}(\Longrightarrow, \mathcal{C})$ as required.

(ii) If $M \mid O \xrightarrow{\tau} [\![M' \mid O]\!]_\theta$ because $M \xrightarrow{\tau} [\![M']\!]_\theta$, by Definition 8 there exists a scheduler $F_1 \in \hat{\mathcal{F}}_{\mathcal{C}}$ such that $Prob^F_{M|O}(\xrightarrow{\tau}, \mathcal{C}) = Prob^{F_1}_M(\xrightarrow{\tau}, \mathcal{D})$. But since $M \approx_p^{\mathcal{F}} N$, there exists $F_2 \in \hat{\mathcal{F}}_{\mathcal{C}}$ such that $Prob^{F_1}_M(\xrightarrow{\tau}, \mathcal{D}) = Prob^{F_2}_N(\Longrightarrow, \mathcal{D})$. For each execution:
$N \xrightarrow{\tau}_{\theta_1} \dots \xrightarrow{\tau}_{\theta_k} N_k \in Exec^{F_1}_N(\Longrightarrow, \mathcal{D})$,
there exists a scheduler $\bar{F} \in \mathcal{F}_{\mathcal{C}}$ such that
$N \to_{\theta_1} N_1 \dots \to_{\theta_k} N_k \in Exec^{\bar{F}}_N$.
By Definition 4, since $\mathcal{F}_{\mathcal{C}}$ captures the interactions of N with any context, $\exists \bar{F}' \in \mathcal{F}_{\mathcal{C}}$ such that, by applying rule (R-Par) to each step in e:
$N \mid O \to_{\theta_1} \dots \to_{\theta_k} N_k \mid O \in Exec^{\bar{F}'}_{N|O}$.
By Definition 8 we finally get $F' \in \hat{\mathcal{F}}_{\mathcal{C}}$ such that:
$Prob^{F_2}_N(\Longrightarrow, \mathcal{D}) =$
$Prob^{\bar{F}}_N(\mathcal{D}) = Prob^{\bar{F}'}_{N|O}(\mathcal{C}) =$
$Prob^{F'}_{N|O}(\Longrightarrow, \mathcal{C})$.

(iii) If $M \mid O \xrightarrow{\tau} [\![M' \mid O']\!]_\Delta$ due to a synchronization between M and O, then there are two cases to consider. If $M \xrightarrow{c_L!\tilde{v}[l,r]} [\![M']\!]_\Delta$ and $O \xrightarrow{c?\tilde{v}@k} [\![O']\!]_\Delta$, for some tuple $\tilde{v}$ of messages, channel c, locations l, k and radius r, such that $d(l,k) \leq r$, we can apply rule (Obs) obtaining $M \xrightarrow{c!\tilde{v}@K\lhd R} [\![M']\!]_\Delta$ for some $R = \{l' \mid d(l,l') \leq r\}$ with $k \in R$ and $K = L \cap R$. Therefore, by Definition 8 there exists $F_1 \in \hat{\mathcal{F}}_{\mathcal{C}}$ such that:
$Prob^F_{M|O}(\xrightarrow{\tau}, \mathcal{C}) = Prob^{F_1}_M(\xrightarrow{c!\tilde{v}@K\lhd R}, \mathcal{D})$.
Since $N \approx_p^{\mathcal{F}} M$, there exists $F_2 \in \hat{\mathcal{F}}_{\mathcal{C}}$ such that
$Prob^{F_1}_M(\xrightarrow{c!\tilde{v}@K\lhd R}, \mathcal{D}) =$
$Prob^{F_2}_N(\overset{c!\tilde{v}@K\lhd R}{\Longrightarrow}, \mathcal{D})$,
where each execution e belonging to $Exec^{F_2}_N(\overset{c!\tilde{v}@K\lhd R}{\Longrightarrow}, \mathcal{D})$ is of the form
$e = N \xrightarrow{\tau}_{\theta_1} N_1 \xrightarrow{\tau}_{\theta_2} \dots N_{i-1} \xrightarrow{c!\tilde{v}@K\lhd R}_\Delta$
$N_i \xrightarrow{\tau}_{\theta_{i+1}} \dots N'$,
with $k \in R$, and, by applying rule (Obs) backwardly, $N_{i-1} \xrightarrow{c!\tilde{v}[l',r']}_\Delta N_i$ for some l' and r' such that $d(l',k) \leq r'$. We can apply rule (Bcast) obtaining $N_{i-1} \mid O \xrightarrow{c!\tilde{v}[l',r']}_\Delta N_i \mid O'$ without changing the probability. Finally if we take $F' \in LSched$ which applies rule (Lose) to the output action, we obtain the required result:
$Prob^{F_2}_N(\overset{c!\tilde{v}@K\lhd R}{\Longrightarrow}, \mathcal{D}) = Prob^{F'}_{N|O}(\Longrightarrow, \mathcal{C})$.
We have finally to prove that $F' \in \hat{\mathcal{F}}_{\mathcal{C}}$. We start by the consideration that, by Definition 1, for any execution of the form $\overset{\alpha}{\Longrightarrow}$ in $\hat{\mathcal{F}}_{\mathcal{C}}$, where α is a silent or an output action there exists a correspondent reduction in $\mathcal{F}_{\mathcal{C}}$. Since by Definition 4, for any context, there exists a scheduler in $\mathcal{F}_{\mathcal{C}}$ mimicking the behaviour exhibited by N when interacting with the given context, we can affirm that $\exists \bar{F} \in \mathcal{F}_{\mathcal{C}}$ such that $Exec^{\bar{F}}_{N|O}$ contains all the reductions corresponding to the execu-

tions of $Exec^{F'}_{N|O}$. Hence, by Definition 8, $F' \in \hat{\mathcal{F}}_{\mathcal{C}}$, as required.

If $M \xrightarrow{c?\tilde{v}@k} [\![M']\!]_{\Delta}$ and $O \xrightarrow{c_L!\tilde{v}[l,r]} [\![O']\!]_{\Delta}$, for some message $\tilde{v}$, channel c, locations l, k and radius r, such that $d(l,k) \leq r$, then by Definition 8 $\exists F_1 \in \hat{\mathcal{F}}_{\mathcal{C}}$ such that:
$Prob^F_{M|O}(\xrightarrow{\tau}, \mathcal{C}) = Prob^{F_1}_M(\xrightarrow{c?\tilde{v}@k}, \mathcal{D})$,
and, since $M \approx^{\mathcal{F}}_p N$, there exists $F_2 \in \hat{\mathcal{F}}_{\mathcal{C}}$ such that
$Prob^{F_1}_M(\xrightarrow{c?\tilde{v}@k}, \mathcal{D}) = Prob^{F_2}_N(\overset{c?\tilde{v}@k}{\Longrightarrow}, \mathcal{D})$
or
$Prob^{F_1}_M(\xrightarrow{c?\tilde{v}@k}, \mathcal{D}) = Prob^{F_2}_N(\Longrightarrow, \mathcal{D})$.
In the first case, since by hypothesis $k \in R$, also N is able to synchronize with O, for all executions
$e = N \xrightarrow{\tau}_{\theta_1} N_1 \xrightarrow{\tau}_{\theta_2} ...N_{i-1} \xrightarrow{c?\tilde{v}@k}_{\Delta}$
$N_i \xrightarrow{\tau}_{\theta_{i+1}} ...N' \in Exec^{F_2}_N(\overset{c?\tilde{v}@k}{\Longrightarrow}, \mathcal{D})$,
since by hypothesis $d(l,k) \leq r$, by applying rule (Bcast) we get $N_{i-1} \mid O \xrightarrow{c_L!\tilde{v}[l.r]} N_i \mid O'$, and there exists a matching execution:
$N \mid O \xrightarrow{\tau}_{\theta_1} N_1 \mid O \xrightarrow{\tau}_{\theta_2} ...N_{i-1} \mid O$
$\xrightarrow{c_L!\tilde{v}[l,r]}_{\Delta} N_i \mid O' \xrightarrow{\tau}_{\theta_{i+1}} ...N' \mid O'$.

By applying the rule (Lose) to the action $N_{i-1} \mid O \xrightarrow{c_L!\tilde{v}[l,r]}_{\Delta} N_i \mid O'$ and by Definition 4 $\exists \bar{F}' \in \mathcal{F}_{\mathcal{C}}$ such that,
$Prob^{\bar{F}'}_{N|O}(\mathcal{C}) = Prob^{F_2}_N(\mathcal{D})$.

By Definition 8 there exists $F' \in \hat{\mathcal{F}}_{\mathcal{C}}$ such that, $Prob^{F'}_{N|O}(\Longrightarrow, \mathcal{C}) = Prob^{\bar{F}'}_{N|O}(\mathcal{C})$.

If N is not able to receive the message the proof is analogous, because $\exists F' \in \hat{\mathcal{F}}_{\mathcal{C}}$ such that, for each execution of $Exec^{F_1}_N(\Longrightarrow, \mathcal{D})$:
$N \xrightarrow{\tau}_{\theta_1} N_1 ... \xrightarrow{\tau}_{\theta_k} N_k$,
by applying rule (Par) to each step:
$N \mid O \xrightarrow{\tau}_{\theta_1} N_1 \mid O ... \xrightarrow{\tau}_{\theta_k} N_k \mid O$,
and by applying rule (Bcast) and (Lose) to O, and then (Par) to $N_k \mid O$, we get:
$N \mid O \xrightarrow{\tau}_{\theta_1} N_1 \mid O ... \xrightarrow{\tau}_{\theta_k} N_k \mid O \xrightarrow{\tau}_{\Delta} N_k \mid O' \in Exec^{F'}_{N|O}(\Longrightarrow, \mathcal{C})$,
hence, since the output of O does not change the probabilities of the executions, we get:
$Prob^F_{M|O}(\Longrightarrow, \mathcal{C}) = Prob^{F_1}_M(\Longrightarrow, \mathcal{D}) =$
$Prob^{F_2}_N(\Longrightarrow, \mathcal{D}) = Prob^{F'}_{N|O}(\Longrightarrow, \mathcal{C})$.

2. if $\alpha = c!\tilde{v}@K \lhd R$ then
   $Prob^F_{M|O}(\xrightarrow{c!\tilde{v}@K \lhd R}, \mathcal{C}) =$
   $Prob^{F'}_{N|O}(\overset{c!\tilde{v}@K \lhd R}{\Longrightarrow}, \mathcal{C})$.
   The proof is analogous to point (iii) of the previous item.
3. if $\alpha = c?\tilde{v}@k$ then
   $Prob^F_{M|O}(\xrightarrow{\alpha}, \mathcal{C}) = Prob^{F'}_{N|O}(\overset{\alpha}{\Longrightarrow}, \mathcal{C})$
   or
   $Prob^F_{M|O}(\xrightarrow{\alpha}, \mathcal{C}) = Prob^{F'}_{N|O}(\Longrightarrow, \mathcal{C})$.
   If $P, Q \in \mathcal{C}$, then by definition of $\mathcal{R}$, $P \equiv \bar{P} \mid \bar{O}$, $Q \equiv \bar{Q} \mid \bar{O}$ and $\bar{P} \approx^{\mathcal{F}}_p \bar{Q}$. But then there exists $\mathcal{D} \in \mathcal{N}/\approx^{\mathcal{F}}_p$ such that $\mathcal{D} = \{\bar{P} : \bar{P} \mid \bar{O} \in \mathcal{C}\}$. Now we have two cases to consider:
   (i) The transition is due to an action performed by O, hence $O \xrightarrow{\alpha}_{\Delta} O'$ and $M \mid O' \in \mathcal{C}$. But since $M \approx^{\mathcal{F}}_p N$, then also $N \mid O' \in \mathcal{C}$, and, by Definition 8 there exists $F' \in \hat{\mathcal{F}}_{\mathcal{C}}$ such that by applying rule (Par) to $O \xrightarrow{\alpha} O'$, we get $N \mid O \xrightarrow{\alpha} N \mid O'$ obtaining:
   $Prob^F_{M|O}(\xrightarrow{\alpha}, \mathcal{C}) = Prob^{F'}_{N|O}(\overset{\alpha}{\Longrightarrow}, \mathcal{C})$.
   (ii) The transition is due to an action performed by M. In this case, by Definition 8 $\exists F_1 \in \hat{\mathcal{F}}_{\mathcal{C}}$ such that:
   $Prob^F_{M|O}(\xrightarrow{\alpha}, \mathcal{C}) = Prob^{F_1}_M(\xrightarrow{\alpha}, \mathcal{D})$.
   Since $M \approx^{\mathcal{F}}_p N$, there exists $F_2 \in \hat{\mathcal{F}}_{\mathcal{C}}$ such that
   $Prob^{F_1}_M(\xrightarrow{\alpha}, \mathcal{D}) = Prob^{F_2}_N(\overset{\alpha}{\Longrightarrow}, \mathcal{D})$,
   or
   $Prob^{F_1}_M(\xrightarrow{\alpha}, \mathcal{D}) = Prob^{F_2}_N(\Longrightarrow, \mathcal{D})$.
   In both cases, for each $e \in Exec^{F_1}_N(\overset{\hat{\alpha}}{\Longrightarrow}, \mathcal{D})$:
   $e = N \xrightarrow{\alpha_1}_{\theta_1} N_1 ... \xrightarrow{\alpha_k}_{\theta_k} N_k$
   by applying rule (Par) to each step we get:
   $N \mid O \xrightarrow{\alpha_1}_{\theta_1} N_1 \mid O ... \xrightarrow{\alpha_k}_{\theta_k} N_k \mid O$.
   Hence, $\exists F' \in LSched$ such that:
   $Prob^{F_2}_N(\overset{\alpha}{\Longrightarrow}, \mathcal{D}) = Prob^{F'}_{N|O}(\overset{\alpha}{\Longrightarrow}, \mathcal{C})$,
   or
   $Prob^{F_2}_N(\Longrightarrow, \mathcal{D}) = Prob^{F'}_{N|O}(\Longrightarrow, \mathcal{C})$.
   In order to prove that $F' \in \hat{\mathcal{F}}_{\mathcal{C}}$, we start by the consideration that, by Definition 8 there exists at least a context $C[\cdot]$ and $\exists \bar{F} \in \mathcal{F}_{\mathcal{C}}$ such that $C[N] \rightarrow C'[N']$, and, by the reduction rules we get:
   $$C[\cdot] \equiv (\nu\tilde{d})m[\bar{c}_{L,r}\langle\tilde{v}\rangle.P]_l \mid M_1$$
   for some $\tilde{d}$ such that $c \notin \tilde{d}$, some m, some set L of locations, some process P, some (possibly empty) network M_1, some location l and some radius r such that $d(l,k) \leq r$. Then, by Definition 4 we have that there exists a scheduler allowing $m[\bar{c}_{L,r}\langle\tilde{v}\rangle.P]_l \rightarrow [\![m[P]_l]\!]_{\Delta}$, and again by Definition 4 there exists a scheduler allowing the reduction $m[\bar{c}_{L,r}\langle\tilde{v}\rangle.P]_l \mid N \mid O \rightarrow^* [\![m[P]_l \mid N' \mid O']\!]_{\Delta}$, and hence, by Definition 8, $F' \in \hat{\mathcal{F}}_{\mathcal{C}}$ as required.

Case 2.

Let us consider now the relation

$$\mathcal{S} = \{((\nu d)M, (\nu d)N) : M \approx^{\mathcal{F}}_p N\}.$$

Let $\mathcal{C} \in \mathcal{N}/\mathcal{S}$: if $P, Q \in \mathcal{C}$, then by definition of $\mathcal{S}$ $P \equiv (\nu\bar{d})\bar{P}$, $Q \equiv (\nu\bar{d})\bar{Q}$ and $\bar{P} \approx^{\mathcal{F}}_p \bar{Q}$. But then $\exists \mathcal{D} \in \mathcal{N}/\approx^{\mathcal{F}}_p$ such that $\mathcal{D} = \{\bar{P} : (\nu\bar{d})\bar{P} \in \mathcal{C}\}$.

We have to prove that, $\forall F \in \hat{\mathcal{F}}_{\mathcal{C}}$, $\exists F' \in \hat{\mathcal{F}}_{\mathcal{C}}$ sucht that, $\forall \mathcal{C} \in \mathcal{N}/\mathcal{S}$, $\forall \alpha$:

1. $\alpha = \tau$ implies that
   $Prob^F_{(\nu d)M}(\xrightarrow{\tau}, \mathcal{C}) = Prob^{F'}_{(\nu d)N}(\Longrightarrow, \mathcal{C})$.
   Since $\texttt{Chan}(\tau) = \bot$, by Definition 8 $\exists F_1 \in \hat{\mathcal{F}}_{\mathcal{C}}$ such that
   $Prob^F_{(\nu d)M}(\xrightarrow{\tau}, \mathcal{C}) = Prob^{F_1}_M(\xrightarrow{\tau}, \mathcal{D})$
   and, since $M \approx^{\mathcal{F}}_p N$ $\exists F_2 \in \hat{\mathcal{F}}_{\mathcal{C}}$ such that: $Prob^{F_1}_M(\xrightarrow{\tau}, \mathcal{D}) = Prob^{F_2}_N(\Longrightarrow, \mathcal{D})$.
   Finally we can take $F' \in LSched$ mimicking the executions in the set $Exec^{F_2}_N(\Longrightarrow, \mathcal{D})$, when applying the restriction on N. Hence:
   $Prob^{F_2}_N(\Longrightarrow, \mathcal{D}) = Prob^{F'}_{(\nu d)N}(\Longrightarrow, \mathcal{C})$.
   In order to prove that $F' \in \hat{\mathcal{F}}_{\mathcal{C}}$, we start by the consideration that, by Definition 4, for any context there exists a scheduler in $\mathcal{F}_{\mathcal{C}}$ mimicking the behaviour of N when interacting with the given context. Hence $\exists \bar{F} \in \mathcal{F}_{\mathcal{C}}$ such that $Exec^{\bar{F}}_{(\nu d)N}$ contains all the reductions corresponding to the executions in $Exec^{F'}_{(\nu d)N}$, i.e., by Definition 8, $F' \in \hat{\mathcal{F}}_{\mathcal{C}}$ as required.
2. $\alpha = c!\tilde{v}@K \lhd R$
   Since $\texttt{Chan}(c!\tilde{v}@K \lhd R) \neq d$, by Definition 8 $\exists F_1 \in \hat{\mathcal{F}}_{\mathcal{C}}$ such that
   $Prob^F_{(\nu d)M}(\xrightarrow{\alpha}, \mathcal{C}) = Prob^{F_1}_M(\xrightarrow{\alpha}, \mathcal{D})$,
   then since $M \approx^{\mathcal{F}}_p N$, $\exists F_2 \in \hat{\mathcal{F}}_{\mathcal{C}}$ such that
   $Prob^{F_1}_M(\xrightarrow{\alpha}, \mathcal{D}) = Prob^{F'}_N(\overset{\alpha}{\Longrightarrow}, \mathcal{D})$.
   Therefore, since $\texttt{Chan}(\alpha) \neq d$, $\exists F' \in LSched$ such that:
   $Prob^{F_2}_N(\overset{\alpha}{\Longrightarrow}, \mathcal{D}) = Prob^{F_2}_{(\nu d)N}(\overset{\alpha}{\Longrightarrow}, \mathcal{C})$.
   We prove that $F' \in \hat{\mathcal{F}}_{\mathcal{C}}$ as in the previous cases.

3. $\alpha = c?\tilde{v}@k$
Again, since $\mathtt{Chan}(c?\tilde{v}@k) \neq d$, by Definition 8 $\exists F_1 \in \hat{\mathcal{F}}_{\mathcal{C}}$ such that
$Prob^{F}_{(\nu d)M}(\xrightarrow{\alpha}, \mathcal{C}) = Prob^{F_1}_{M}(\xrightarrow{\alpha}, \mathcal{D})$.
Since $M \approx^{\mathcal{F}}_{p} N$, there exists $F_2 \in \hat{\mathcal{F}}_{\mathcal{C}}$ such that
$Prob^{F_1}_{M}(\xrightarrow{\alpha}, \mathcal{D}) = Prob^{F_2}_{N}(\overset{\alpha}{\Longrightarrow}, \mathcal{D})$ or
$Prob^{F_1}_{M}(\xrightarrow{\alpha}, \mathcal{D}) = Prob^{F_2}_{N}(\Longrightarrow, \mathcal{D})$,
when N is not able to receive $\tilde{v}$. In both cases we can apply rule (Res) to N, since $\mathtt{Chan}(\tau) = \bot$ and $\mathtt{Chan}(c?\tilde{v}@k) \neq d$. Hence, there exists $F' \in LSched$ such that the required result holds, i.e.,
$Prob^{F_2}_{N}(\overset{\alpha}{\Longrightarrow}, \mathcal{D}) = Prob^{F'}_{(\nu d)N}(\overset{\alpha}{\Longrightarrow}, \mathcal{C})$ or
$Prob^{F_2}_{N}(\Longrightarrow, \mathcal{D}) = Prob^{F'}_{(\nu d)N}(\Longrightarrow, \mathcal{C})$.
Again, we prove that $F' \in \hat{\mathcal{F}}_{\mathcal{C}}$ as in the previous cases. □

Proof of Theorem 3

In order to prove the completeness of the probabilistic labelled bisimilarity we show that the relation

$$\mathcal{R} = \{(M, N) : M \cong^{\mathcal{F}}_{p} N\}$$

is a probabilistic labelled bisimulation.

We have to prove that, $\forall F \in \hat{\mathcal{F}}_{\mathcal{C}}$ $\exists F' \in \hat{\mathcal{F}}_{\mathcal{C}}$ such that, $\forall \mathcal{C} \in \mathcal{N}/\mathcal{R}$, $\forall \alpha$:

if $\alpha = \tau$ then $Prob^{F}_{M}(\xrightarrow{\tau}, \mathcal{C}) = Prob^{F'}_{N}(\Longrightarrow, \mathcal{C})$.
By Theorem 1 and by Definition 8 we know that $\exists \bar{F} \in \mathcal{F}_{\mathcal{C}}$ such that $Prob^{F}_{M}(\xrightarrow{\tau}, \mathcal{C}) = Prob^{\bar{F}}_{M}(\mathcal{C})$, and, since $M \cong^{\mathcal{F}}_{p} N$, $\exists \bar{F}' \in \mathcal{F}_{\mathcal{C}}$ such that $Prob^{\hat{F}}_{M}(\mathcal{C}) = Prob^{\bar{F}'}_{N}(\mathcal{C})$. Again by Theorem 1 and by Definition 8 $\exists F' \in \hat{\mathcal{F}}_{\mathcal{C}}$ such that $Prob^{\hat{F}'}_{N}(\mathcal{C}) = Prob^{F'}_{N}(\Longrightarrow, \mathcal{C} \cup \{\bar{N} \equiv N' \in \mathcal{C}\})$, but since $\cong^{\mathcal{F}}_{p}$ is closed under structural equivalence, $\forall \bar{N} \equiv N' \in \mathcal{C}$, $\bar{N} \in \mathcal{C}$, and hence: $Prob^{F}_{M}(\xrightarrow{\tau}, \mathcal{C}) = Prob^{F'}_{N}(\Longrightarrow, \mathcal{C})$.

if $\alpha = c!\tilde{v}@K \triangleleft R$ then
$Prob^{F}_{M}(\xrightarrow{\alpha}, \mathcal{C}) = Prob^{F'}_{N}(\overset{\alpha}{\Longrightarrow}, \mathcal{C})$.
First we notice that $Prob^{F}_{M}(\xrightarrow{c!\tilde{v}@K \triangleleft R}, \mathcal{C})$ is either 0 or 1.
If $Prob^{F}_{M}(\xrightarrow{c!\tilde{v}@K \triangleleft R}, \mathcal{C}) = 0$ we are done, because it will be enough to take any scheduler $F' \in \hat{\mathcal{F}}_{\mathcal{C}}$ not allowing observable output actions on the channel c, and we get $Prob^{F}_{M}(\xrightarrow{c!\tilde{v}@K \triangleleft R}, \mathcal{C}) = Prob^{F'}_{N}(\overset{c!\tilde{v}@K \triangleleft R}{\Longrightarrow}, \mathcal{C})$.
If $Prob^{F}_{M}(\xrightarrow{c!\tilde{v}@K \triangleleft R}, \mathcal{C}) = 1$, by Theorem 1 and by Definition 8 $\exists \bar{F} \in \mathcal{F}_{\mathcal{C}}$ such that $M \Downarrow^{\bar{F}}_{1} c@K$, and this means that $\exists \bar{F}' \in \mathcal{F}_{\mathcal{C}}$ such that $N \Downarrow^{\bar{F}'}_{1} c@K$, hence, again by Theorem 1 and by Definition 8 there exist $F' \in \hat{\mathcal{F}}_{\mathcal{C}}$ and R' such that $K \subseteq R'$ and $Prob^{\bar{F}'}_{N}(\mathcal{C}) = Prob^{F'}_{N}(\overset{c!\tilde{v}@K \triangleleft R'}{\Longrightarrow}, \mathcal{C})$.
We proved that $\exists R'$ with $Prob^{F}_{M}(\xrightarrow{c!\tilde{v}@K \triangleleft R}, \mathcal{C}) = Prob^{F'}_{N}(\overset{c!\tilde{v}@K \triangleleft R'}{\Longrightarrow}, \mathcal{C})$, now we want to show that $R' = R$. In order to mimic the effect of the action $c!\tilde{v}@K \triangleleft R$, we build the following context
$C[\cdot] = \prod_{i=1}^{n}(n_i[c(\tilde{x}_i).[\tilde{x}_i = \tilde{v}]\bar{\mathtt{f}}^{(i)}_{k_i,r}\langle \tilde{x}_i \rangle]_{k_i} \mid m_i[\mathtt{f}^{(i)}(\tilde{y}_i).\bar{\mathtt{ok}}^{(i)}_{k_i,r}\langle \tilde{y}_i \rangle]_{k_i})$,
where $R = \{k_1, ..., k_n\}$, n_i, m_i, $\mathtt{ok}^{(i)}$ and $\mathtt{f}^{(i)}$ fresh $\forall i \in [1-n]$. Since $M \xrightarrow{c!\tilde{v}@K \triangleleft R}$, then the message is reachable by all nodes n_i, hence, by Definition 4 $\exists \bar{F}_1 \in \mathcal{F}_{\mathcal{C}}$ such that $C[M] \longrightarrow^{*} \hat{M}$, where
$\hat{M} \equiv M' \mid \prod_{i=1}^{n}(n_i[\mathbf{0}]_{k_i} \mid m_i[\bar{\mathtt{ok}}^{(i)}_{k_i,r}\langle \tilde{v}_i \rangle]_{k_i} \equiv M' \mid \prod_{i=1}^{n}(m_i[\bar{\mathtt{ok}}^{(i)}_{k_i,r}\langle \tilde{v}_i \rangle]_{k_i})$,
with $\hat{M} \not\downarrow_{\mathtt{f}^{(i)}@R}$ and $\hat{M} \Downarrow^{\bar{F}_1}_{1} \mathtt{ok}^{(i)}@R$, $\forall i \in [1-n]$.
The absence of the barb on the channels $\mathtt{f}^{(i)}$ together with the presence of the barb on the channels $\mathtt{ok}^{(i)}$ ensures that all the locations in R have been able to receive the message. Since $C[M] \cong^{\mathcal{F}}_{p} C[N]$, $\exists \bar{F}_2 \in \mathcal{F}_{\mathcal{C}}$ such that $Prob^{\bar{F}_1}_{C[M]}(\mathcal{C}') = Prob^{\bar{F}_2}_{C[N]}(\mathcal{C}')$ where $\hat{M} \in \mathcal{C}'$.
Therefore, $C[N] \longrightarrow^{*} \hat{N}$ with $\hat{N} \not\downarrow_{\mathtt{f}^{(i)}@R}$ and $\hat{N} \Downarrow^{\bar{F}_2}_{1} \mathtt{ok}^{(i)}@R$.
The constrains on the barbs allow us to deduce that
$\hat{N} \equiv N' \mid \prod_{i=1}^{n}(n_i[\mathbf{0}]_{k_i} \mid m_i[\bar{\mathtt{ok}}^{(i)}_{k_i,r}\tilde{v}_i]_{k_i}) \equiv N' \mid \prod_{i=1}^{n}(m_i[\bar{\mathtt{ok}}^{(i)}_{k_i,r}\tilde{v}_i]_{k_i})$,
which implies $N \overset{c!\tilde{v}@K \triangleleft R}{\Longrightarrow} N'$, or $N \Longrightarrow N'$ in case (Lose) has been applied to the output action on the channel c.
Since $\hat{M}, \hat{N} \in \mathcal{C}$, then $\hat{M} \cong^{\mathcal{F}}_{p} \hat{N}$, and since $\cong^{\mathcal{F}}_{p}$ is contextual, it results $(\nu \mathtt{ok}^{(1)}...\mathtt{ok}^{(n)})\hat{M} \cong^{\mathcal{F}_{\mathcal{M}}}_{p} (\nu \mathtt{ok}^{(1)}...\mathtt{ok}^{(n)})\hat{N}$.
By applying (Struct Res Par):
$(\nu \mathtt{ok}^{(1)}...\mathtt{ok}^{(n)})\hat{M} \equiv$
$M' \mid (\nu \mathtt{ok}^{(1)}...\mathtt{ok}^{(n)})\prod_{i=1}^{n}(m_i[\bar{\mathtt{ok}}^{(i)}_{k_i,r}\langle \tilde{v}_i \rangle]_{k_i}) \equiv M'$
and
$(\nu \mathtt{ok}^{(1)}...\mathtt{ok}^{(n)})\hat{N} \equiv$
$N' \mid (\nu \mathtt{ok}^{(1)}...\mathtt{ok}^{(n)})\prod_{i=1}^{n}(m_i[\bar{\mathtt{ok}}^{(i)}_{k_i,r}\langle \tilde{v}_i \rangle]_{k_i}) \equiv N'$
and, since the network
$(\nu \mathtt{ok}^{(1)}...\mathtt{ok}^{(n)})\prod_{i=1}^{n}(m_i[\bar{\mathtt{ok}}^{(i)}_{k_i,r}\langle \tilde{v}_i \rangle]_{k_i})$
is silent, we can derive that $M' \cong^{\mathcal{F}}_{p} N'$. But since $N' \in \mathcal{C}$ and $N \overset{c!\tilde{v}@K \triangleleft R}{\Longrightarrow} N'$, by Definition 8 $\exists F' \in \hat{\mathcal{F}}_{\mathcal{C}}$ such that:
$Prob^{F'}_{N}(\overset{c!\tilde{v}@K \triangleleft R}{\Longrightarrow}, \mathcal{C}) = 1 = Prob^{F}_{M}(\overset{c!\tilde{v}@K \triangleleft R}{\Longrightarrow}, \mathcal{C})$,
as required.

if $\alpha = c?\tilde{v}@k$ then $Prob^{F}_{M}(\xrightarrow{\alpha}, \mathcal{C}) = Prob^{F'}_{N}(\overset{\alpha}{\Longrightarrow}, \mathcal{C})$ or $Prob^{F'}_{N}(\Longrightarrow, \mathcal{C})$.
We notice that $Prob^{F}_{M}(\xrightarrow{c?\tilde{v}@k}, \mathcal{C})$ is either 0 or 1.
If $Prob^{F}_{M}(\xrightarrow{c?\tilde{v}@k}, \mathcal{C}) = 0$ we are done, because it will be enough to take any scheduler $F' \in \hat{\mathcal{F}}_{\mathcal{C}}$ not allowing input actions on the channel c, and we get $Prob^{F}_{M}(\xrightarrow{c?\tilde{v}@k}, \mathcal{C}) = Prob^{F'}_{N}(\overset{c?\tilde{v}@k}{\Longrightarrow}, \mathcal{C})$.
If $Prob^{F}_{M}(\xrightarrow{c?\tilde{v}@k}, \mathcal{C}) = 1$, because $M \xrightarrow{c?\tilde{v}@k} [\![M']\!]_{\Delta}$, by Definition 4 there exists at least a context $C[\cdot]$ and $\exists \bar{F} \in \mathcal{F}_{\mathcal{C}}$ such that $C[M] \rightarrow C'[M']$, and by Theorem 1 we deduce that:
$C[\cdot] \equiv (\nu \tilde{d})m[\bar{c}_{L,r}\langle \tilde{v} \rangle.P]_l \mid M_1$ and
$C'[\cdot] \equiv (\nu \tilde{d})m[P]_l \mid M'_1$
for some m, some tuple $\tilde{d}$ of channels such that $c \notin \tilde{d}$, some set L of messages, some radius r, some process P, some location l such that $d(l, k) \leq r$ and some (possibly empty) networks M_1 and M'_1.
By Definition 4, for any context there exists a scheduler in $\mathcal{F}_{\mathcal{C}}$ allowing m to perform the output when interacting with any context. Hence we can build the following context:
$C_1[\cdot] = \cdot \mid m[\bar{c}_{L,r}\langle \tilde{v} \rangle.P]_l \mid m_1[c(\tilde{x}).\bar{\mathtt{f}}_{k,r'}\langle \tilde{x} \rangle.\bar{\mathtt{ok}}_{k,r'}\langle \tilde{x} \rangle]_k$,
in order to mimic the behaviour of the networks, with m static, $\mathtt{f}$ and $\mathtt{ok}$ fresh channels, $r' > 0$ and $d(l, k) > r'$ $\forall l \in Loc$ such that $l \neq k$. Hence, there exists a scheduler $\bar{F}_1 \in \mathcal{F}_{\mathcal{C}}$ such that: $C_1[M] \longrightarrow^{*} M' \mid m[P]_l \mid m_1[\bar{\mathtt{ok}}_{k,r'}\langle \tilde{v} \rangle]_k \in Exec^{\bar{F}_1}_{C[M]}$,
with $M' \mid m[P]_l \mid m[\bar{\mathtt{ok}}_{k,r'}\langle \tilde{v} \rangle]_k \not\downarrow_{\mathtt{f}@k}$ and
$M' \mid m[P]_l \mid m[\bar{\mathtt{ok}}_{k,r'}\langle \tilde{v} \rangle]_k \Downarrow^{\bar{F}_1}_{1} \mathtt{ok}@k$.
The reduction sequence above must be matched by a corresponding reduction sequence $C_1[N] \longrightarrow^{*} N' \mid m[P]_l \mid m[\bar{\mathtt{ok}}_{k,r'}\langle \tilde{v} \rangle]_k$, with
$M' \mid m[P]_l \mid m[\bar{\mathtt{ok}}_{k,r'}\langle \tilde{v} \rangle]_k \cong_p$
$N' \mid m[P]_l \mid m[\bar{\mathtt{ok}}_{k,r'}\langle \tilde{v} \rangle]_k \not\downarrow_{\mathtt{f}@k}$ and
$N' \mid m[P]_l \mid m[\bar{\mathtt{ok}}_{k,r'}\langle \tilde{v} \rangle]_k \Downarrow^{\hat{F}_2}_{1} \mathtt{ok}@k$ for some $\bar{F}_2 \in \mathcal{F}_{\mathcal{C}}$.
This does not ensure that N actually performed the input action, but we can conclude that there exists $F' \in LSched$ and N' such that either $N \overset{c?\tilde{v}@k}{\Longrightarrow} N'$ or $N \Longrightarrow N'$. Since $M' \mid m[P]_l \mid m[\bar{\mathtt{ok}}_{k,r'}\langle \tilde{v} \rangle]_k \cong_p N' \mid m[P]_l \mid m[\bar{\mathtt{ok}}_{k,r'}\langle \tilde{v} \rangle]_k$

and $\cong_p^{\mathcal{F}}$ is is a contextual relation, we can easily derive $M' \cong_p^{\mathcal{F}} N'$ (applying the rules for structural equivalence), i.e., $\exists F' \in LSched$ such that:
$Prob_M^F(\xrightarrow{c?\tilde{v}@k}, \mathcal{C}) = 1 = Prob_N^{F'}(\overset{c?\tilde{v}@k}{\Longrightarrow}, \mathcal{C})$
or
$Prob_M^F(\xrightarrow{c?\tilde{v}@k}, \mathcal{C}) = 1 = Prob_N^{F'}(\Longrightarrow, \mathcal{C})$.
Now we have only to prove that $F' \in \hat{\mathcal{F}}_{\mathcal{C}}$, but this follows straightforwardly by Definition 8, since $\bar{F}_2 \in \mathcal{F}_{\mathcal{C}}$. □