Università
Ca'Foscari
Venezia

**Dipartimento
di Scienze Ambientali
Informatica e Statistica**

Technical Report Series

Rapporto di Ricerca DAIS-2011-3

Aprile 2011

L. Gallina, S. Hamadou, A. Marin

S. Rossi

A Probabilistic Energy-Aware Model for Ad-Hoc Networks

A Probabilistic Energy-Aware Model for Mobile Ad-Hoc Networks

Lucia Gallina, Sardaouna Hamadou, Andrea Marin, and Sabina Rossi

Università Ca' Foscari, Venezia (Italy)
{lgallina,sh,marin,srossi}@dais.unive.it

Abstract. We propose a probabilistic, energy-aware, broadcast calculus for the analysis of mobile ad-hoc networks. The semantics of our model is expressed in terms of Segala's probabilistic automata driven by schedulers to resolve the nondeterministic choice among the probability distributions over target states. We develop a probabilistic observational congruence and a energy-aware preorder semantics. The observational congruence allows us to verify whether two networks exhibit the same observable probabilistic behaviour (connectivity), while the preorder is used to compare the energy consumption of different, but behaviourally equivalent, networks. As an application, we analyse and compare the energy consumption of two well-known automatic repeat request (ARQ)-based error control protocols: stop-and-wait (SW) and go-back-N (GBN).

1 Introduction

Mobile ad-hoc networks (MANETs) consist of a collection of nodes that communicate with each other through wireless links without a pre-established networking infrastructure. A common feature of most of these networks is free node mobility: each device in a MANET is free to move independently in any direction, and will therefore change its links to other devices 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 connectivity. Moreover, mobile devices may have strict requirements on the energy consumption because their expected life-time often depends on the energy stored in a battery or other exhaustible power sources. Hence, the communication protocols must face the problem of providing a full connectivity among the network devices while maintaining good performance both in terms of throughput and of energy conservation (see, e.g., [14, 12]).

The definition of a general formalism capable of expressing both qualitative (connectivity) and quantitative (power consumption and throughput) analysis is a challenging topic of research.

In this paper we define a calculus, named Probabilistic E-BUM, for formally reasoning about probabilistic energy-aware broadcast, unicast and multicast communications of mobile ad-hoc networks. This is an extension of the E-BUM calculus [3] where probability distributions are used to describe the movements of 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. The connectivity of a node is represented by a location and a transmission radius. 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 addresses of the intended recipients of the message. Moreover, the possibility for a node to control its transmission power is modeled by enabling nodes to modify the transmission radius of their communications.

The Probabilistic E-BUM calculus deals with both nondeterministic and probabilistic choices. Its semantics is expressed in terms of Segala's probabilistic automata [11] driven by schedulers to resolve the nondeterministic choice among the probability distributions over target states. In this paper we propose a notion of probabilistic observational congruence in the style of [8] and also define a labelled semantics to model the interactions of the system with the surrounding environmnet. We provide a labelled bisimilarity as an efficient proof method for the observational congruence. Intuitively, two networks are deemed equivalent if they exhibit the same probabilistic behaviour (connectivity) relative to the corresponding set of intended recipients and to a specific set of schedulers. Moreover, based on the labelled semantics we define an energy-aware preorder over networks which allows us to compare the average energy cost of different networks but exhibiting the same connectivity behaviour. This property can be used to replace a network component with a less energy consuming one while maintaining connectivity. As an application we analyse and compare the energy consumption of two well-known automatic repeat request (ARQ)-based error control protocols: stop-and-wait (SW) and go-back-N (GBN).

Related works.

Probabilistic and stochastic models are nowadays widely used in the design and verification of complex systems. Song and Godskesen [13] proposed a probabilistic broadcast calculus for mobile and wireless networks whose connections are unreliable. The peculiarity of this calculus is the introduction of a *probabilistic mobility function* which defines the mobility rules of the connections. Palamidessi et al. [4] defined an extension of the applied pi-calculus with nondeterministic and probabilistic choice operators while Priami introduced a stochastic extension of the pi-caluclus [7], which allows one to describe dynamically reconfigurable or mobile networks. It associates exponentially probability distributions to the actions of processes, governing their duration, giving rice to a Marokovian process. Another important stochastic process algebra for performance evalutation is PEPA[5], introduced by Jane Hillston at al. which is used for modelling systems composed of concurrently active components which co-operate and share work. Here cooperation is modelled in CSP style, using share names, in order to represent something more general then communications. Bernardo et al. introduced EMPA$_{gr}$ [2], an extended Markovian process algebra including probabilities, priority and exponentially distributed durations. The peculiarity of this calculus is the modelling of exponentially timed as well as

immediate actions, whose selection is control by a priority level associated with them.

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 [14] the authors define a Markov Reward process [9] modelling some protocols for pairwise node communications. Quantitative analysis in steady-state is then derived and hence the average performance indices computed. In [12] 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 provididing a common framework for both qualitative and quantitative analysis.

Plan of the paper. Section 1 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 energy consumption of two well-known automatic repeat request (ARQ)-based error control protocols: stop-and-wait (SW) and go-back-N (GBN). Finally, Section 6 concludes the paper.

2 The Calculus

We introduce the Probabilistic E-BUM calculus, an extension of E-BUM (a calculus for Energy-aware Broadcast, Unicast, Multicast communications in mobile ad-hoc networks) [3] 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 administrate energy consumption by choosing the optimal transmission radius to communicate with the desired recipients.

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 (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 (which represent devices), running in parallel, using channels to communicate messages. As usual, $\mathbf{0}$ denotes the empty network and $M_1|M_2$ represents the parallel composition of two 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 $|\tilde{x}| = |\tilde{w}|$). 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

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

Table 1: Syntax

to maintain the set of locations of the intended recipients: $L = \infty$ represents a broadcast transmission, while a finite set of locations L denotes a multicast communication (unicast if L is a singleton). The tag r represents the power of the transmission: we assume that the choice of the transmission power may depend on precise strategies which are implemented in the communication protocol; hence it is reasonable considering the transmission radius of a communication as an information given by the process running in the sender node. Syntactically, the tags L and r associated to the channel c in an output action 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. In the process $c(\tilde{x}).P$, the variables in $\tilde{x}$ are bound 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 $c_{\{l\}}$, $\bar{c}_{L,r}\langle w\rangle$ for $\bar{c}_{L,r}\langle w\rangle.\mathbf{0}$, and $[w_1 = w_2]P$ for $[w_1 = w_2]P, \mathbf{0}$.

Nodes cannot be created or destroyed. We write $n[P]_l$ for a node named n located at the physical location l, and executing a process P. Each node n is associated to a pair $< r_n, \mathbf{J}^n >$, where r_n is a positive (possibly 0) real number denoting the maximum transmission radius that n can use to transmit, while $\mathbf{J}^n$ is a Markov chain, where $\mathbf{J}^n_{lk}$ is the probabilty that the node n located at l, after executing a movement action, will be located at k. Hence, $\sum_{k \in Loc} \mathbf{J}^n_{lk} = 1$ for all locations $l \in Loc$. Stationary nodes are associated to 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 $l \neq k$. Therefore, if the initial location of a stationary node is l, then l is the only location reachable according to the node's Markov chain.

Nodes connectivity is verified by looking at the physical location and the transmission radius of the sender: if a node broadcasts a message, this information will be received only by the nodes that lie in the area delimited by the transmission radius of the sender. In the definition of the operational semantics we then assume the possibility of comparing locations so to determine wether a node lies or not within the transmission cell of another node. We do so by means of a function $d(\cdot, \cdot)$ which takes two locations and returns the distance separating them (the function d can be simply the euclidian distance between two locations, or a more complex function considering the possible obstacles of the surrounding environment).

In the following, we denote by $\mathcal{N}$ the set of all networks.

$n[[v = v]P, Q]_l \equiv n[P]_l$	(Struct Then)
$n[[v_1 = v_2]P, Q]_l \equiv n[Q]_l \quad v_1 \neq v_2$	(Struct Else)
$n[A\langle\tilde{v}\rangle]_l \equiv n[P\{\tilde{v}/\tilde{x}\}]_l \quad \text{if } A(\tilde{x}) \stackrel{\text{def}}{=} P \wedge |\tilde{x}| = |\tilde{v}|$	(Struct Rec)
$M \mid N \equiv N \mid M$	(Struct Par Comm)
$(M \mid N) \mid M' \equiv M \mid (N \mid M')$	(Struct Par Assoc)
$M \mid \mathbf{0} \equiv M$	(Struct Zero Par)

Table 2: Structural Congruence

Probability distributions. A network M is defined as the parallel composition of pairwise-distinct nodes moving independently from each other. We denote by $\prod_{i \in I} M_i$ the parallel composition of the networks M_i, for $i \in I$. In our framework, the mobility of the nodes is the only source of probability. We associate probability distributions to located nodes and model the probabilistic evolution of the network according to these distributions. More formally, we denote by μ_l^n the probability distribution associated to the node n located at l, that is a function over the set *Loc* of locations such that, for all $k \in Loc$, $\mu_l^n(k) = \mathbf{J}_{lk}^n$ denoting the probability that the node n located at l moves at the location k.

Let $M = \prod_{i \in I} n_i[P_i]_{l_i}$ be a network, then for all k in I, $[\![M]\!]_{\mu_{l_k}^{n_k}}$ denotes the probability distribution over the set of networks induced by $\mu_{l_k}^{n_k}$ and defined as follows: for all network M':

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

Note that $[\![M]\!]_{\mu_{l_k}^{n_k}}(M')$ is the probability that the network M evolves to M' due to the movement of the node n_k located at l_k. We say that M' is in the support of $[\![M]\!]_{\mu_{l_k}^{n_k}}$ if $[\![M]\!]_{\mu_{l_k}^{n_k}}(M') \neq 0$. 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, θ' range over

$$\{\mu_l^n \mid n \text{ is a node and } l \in Loc\} \cup \{\Delta\}.$$

Reduction Semantics. The dynamics of the calculus is specified by the *probabilistic reduction relation* over networks ($\rightarrow$), described in Table 3. As usual in process calculi, 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 $M \rightarrow [\![M']\!]_\theta$, which describes a transition that leaves from network M and leads to a probability distribution $[\![M']\!]_\theta$.

Rule (R-Bcast) models the transmission of a tuple of messages $\tilde{v}$ to the set of intended recipients 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

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

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

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

$$\text{(R-Struct)}\ \frac{M \rightarrow [\![M']\!]_\theta \ M' \equiv N'}{M \rightarrow [\![N']\!]_\theta}$$

Table 3: Reduction Semantics

we decided to associate to each output action a set of transmission recipients. The cardinality of this set indicates the kind of communication that is used: if $L = \infty$ 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 realized. The recipients set indicates which are the nodes really interested in receiving that particular message, but we know that every message sent from a node will be potentially received by all the devices lying within the transmission cell of the sender. If two nodes want to share a secret, they must use cryptography to hide the message. A radius r is also associated to the channel c, indicating the transmission radius required for that communication which may depend on the energy consumption strategy adopted by the surrounding protocol. In our calculus transmission is a *non-blocking action*: transmission proceeds even if there are no nodes listening for messages.

Rule (R-Move) deals with node mobility. A node n located at l and executing a moving 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 to n. Movements are atomic actions: while moving, a node cannot do anything else.

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 $M \rightarrow_\theta N$ if $M \rightarrow [\![M']\!]_\theta$, and N is in the support of $[\![M']\!]_\theta$. Following [4], an execution for M is a (possibly infinite) sequence of steps $M \rightarrow_{\theta_1} M_1 \rightarrow_{\theta_2} M_2 ....$

In the rest of the paper, 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 prifix execution $M \rightarrow_{\theta_1} M_1 ... \rightarrow_{\theta_j} M_j$ of lenght j of the execution $e = M \rightarrow_{\theta_1} M_1 \cdots \rightarrow_{\theta_j} M_j \rightarrow_{\theta_{j+1}} M_{j+1} \cdots$, and $e \uparrow$ for the set of e' such that $e \leq_{prefix} e'$. Given a network M and a probability distribution $[\![M']\!]_\theta$, we write $M \rightarrow^* M'$ if there exists a finite execution $e \in Exec_M$ such that $last(e) = M'$.

Observational Semantics. The central actions of our calculus are transmission and reception of messages. However, only the transmission of messages can be observed. An observer cannot be sure whether a recipient actually receives a

given value. Instead, if a node receives a message, then surely someone must have sent it. Following [8], we use the term *barb* as a synonymous of observable. However our calculus presents both non-deterministic and probabilistic aspects, where the non-deterministic choices are among the possible probability distributions that a process may follow and arise from the possibility for nodes to perform arbitrary, unpredicatble, movements.

We denote by $behave(M) = \{[\![M']\!]_\theta \mid M \rightarrow [\![M']\!]_\theta \}$ the set of the possible behaviours of M. In order to solve the non-determinism in a network execution, we consider each possible probabilistic transition $M \rightarrow [\![M']\!]_\theta$ as arising from a *scheduler* (see [11]). A *scheduler* is a total function F assigning to a finite execution e a distribution $[\![N]\!]_\theta \in behave(last(e))$. We shall denote the set of schedulers by *Sched*. Given a network M and a scheduler F, we define the set of executions starting from M and driven by F as:

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

Formally, 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 $Exex_M^F(H) = \{e \in Exec_M^F \mid last(e^j) \in H \text{ for some } j\}$. We denote the probability for a network M to evolve into a network in H according to the policy given by F as $Prob_M^F(H) = Prob_M^F(Exec_M^F(H))$.

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 intended recipients is able to receive the message.

Definition 1 (Barb). *We say that a network M has a* barb *on a channel c for a given scheduler F, written $M \downarrow_c^F$, if*

i $M \equiv n[\bar{c}_{L,r}\langle\tilde{v}\rangle.P]_l \mid N$
ii *there exists* $k \in L$ *such that* $d(l,k) \leq r$
iii $\forall e \in Exec_M^F$ *s.t.* $last(e) = M$ $F(e) = [\![n[P]_l \mid N]\!]_\Delta$

Definition 2 (Probabilistic Barb). *We say that a network M has a* probabilistic barb *with probability p on a channel c according to the scheduler F, written $M \Downarrow_p^F c$, if $Prob_M^F(H) = p$ where $H = \{M' \mid M \rightarrow^* M',\ M' \downarrow_c^F\}$.*

Intuitively, for a given network M and scheduler F, if $M \Downarrow_p^F c$ then there is a positive probability that M, driven by F, performs a transmission on channel c and at least one of the intended recipients is able to correctly listen to it.

Following, we introduce a probabilistic observational congruence, in the style of [1, 4, 10], which is parametric with respect to a set of schedulers $\mathcal{F} \subseteq \mathit{Sched}$ and is defined as the largest $\mathcal{F}$-relation which satisfies the following properties. Let $\mathcal{R}^{\mathcal{F}}$ be a relation over networks:

Barb preservation. $\mathcal{R}^{\mathcal{F}}$ is *barb preserving* if $M \mathcal{R}^{\mathcal{F}} N$ and $M \Downarrow_p^F c$ for some $F \in \mathcal{F}$ implies that there exists $F' \in \mathcal{F}$ such that $N \Downarrow_p^{F'} c$.

Reduction closure. $\mathcal{R}^{\mathcal{F}}$ is *reduction closed* if $M \mathcal{R}^{\mathcal{F}} N$ implies that for all $F \in \mathcal{F}$, there exists $F' \in \mathcal{F}$ such that for all classes $\mathcal{C} \in \mathcal{N}/\mathcal{R}^{\mathcal{F}}$ $Prob_M^F(\mathcal{C}) = Prob_N^{F'}(\mathcal{C})$.

Contextuality. $\mathcal{R}^{\mathcal{F}}$ is *contextual* if $M \mathcal{R}^{\mathcal{F}} N$ implies that for every context $\mathcal{C}[\cdot]$, it holds that $\mathcal{C}[M] \, \mathcal{R}^{\mathcal{F}} \, \mathcal{C}[N]$, where a context is a network term with a hole $[\cdot]$ defined by the grammar: $\mathcal{C}[\cdot] ::= [\cdot] \mid [\cdot] || M \mid M || [\cdot]$.

Definition 3 (Probabilistic observational congruence w.r.t. $\mathcal{F}$). Probabilistic observational congruence w.r.t. a set $\mathcal{F}$ of schedulers*, written* $\cong_p^{\mathcal{F}}$*, is the largest symmetric* $\mathcal{F}$*-relation over networks which is reduction closed, barb preserving and contextual.*

Two networks are related by $\cong_p^{\mathcal{F}}$ if they exhibit the same probabilistic behaviour (communications) relative to the corresponding sets of intended recipients and the fixed set of schedulers $\mathcal{F}$. Hereafter we develop a bisimulation-based proof technique for $\cong_p^{\mathcal{F}}$. It provides an efficient method to check whether two networks are related by $\cong_p^{\mathcal{F}}$.

3 LTS 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 $c(\tilde{v})$ and $\bar{c}_{L,r}\langle \tilde{v} \rangle$, respectively.

Rules for processes are simple and they do not need deeper explanations. Note that such rules do not rely on any probabilistic notion since processes executions are simply deterministic sequences of actions.

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 are used to model the mobility of nodes. The tag γ is as follows:

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

Rule (Snd) models the sending, with transmission radius r, of the tuple $\tilde{v}$ through channel c to a specific set L of recipients, while rule (Rcv) models the reception of $\tilde{v}$ at l via channel c. Rule (Bcast) models the broadcast message

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

$$\text{(Then)}\ \frac{P \xrightarrow{\eta} P'}{[\tilde{v} = \tilde{v}]P, Q \xrightarrow{\eta} P'} \qquad \text{(Else)}\ \frac{Q \xrightarrow{\eta} Q' \quad \tilde{v_1} \neq \tilde{v_2}}{[\tilde{v_1} = \tilde{v_2}]P, Q \xrightarrow{\eta} Q'}$$

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

Table 4: LTS rules for Processes

propagation: all the nodes lying within the transmission cell of the sender may receive the message, regardless of the fact that they are in L. Rule (Obs) models the observability of a transmission: every transmission may be detected (and hence *observed*) by any intended recipient located within the transmission cell of the sender. The label $c!\tilde{v}@K$ represents the transmission of the tuple $\tilde{v}$ of messages via c to a set K of recipients in L, located within the transmission cell of the sender. Rule (Lose) models both message loss and a local activity of the network which an observer is not party to. 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 to n.
Based on the LTS semantics, we define a probabilistic labelled bisimilarity, which we show is a complete characterisation of our *probabilistic observational congruence*. It is built upon the following actions:

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

Again, we write $M \xrightarrow{\alpha}_\theta N$ if $M \xrightarrow{\alpha} [\![M']\!]_\theta$ and N is in the support of $[\![M']\!]_\theta$. Moreover we write $M \xrightarrow{\alpha} N$ if $M \xrightarrow{\alpha}_\theta N$ for some θ. An *execution* e of a network M is a finite (or infinite) sequence of steps: $M \xrightarrow{\alpha_1}_{\theta_1} M_1 \xrightarrow{\alpha_2}_{\theta_2} M_2 ... \xrightarrow{\alpha_k}_{\theta_k} M_k$. With abuse of notation, we define $Exec_M$, $last(e)$, e^j and $e\uparrow$ as for unlabelled executions. We denote by $lbehave(M)$ the set of all possible behaviors of M, that is, $lbehave(M) = \{(\alpha, [\![M']\!]_\theta) \mid M \xrightarrow{\alpha} [\![M']\!]_\theta\}$. Executions arise by resolving the non-determinism of both α and $[\![M]\!]_\theta$. As a consequence, a scheduler for the labelled semantics is a function[1] F assigning to a finite labelled execution e a pair $(\alpha, [\![M]\!]_\theta) \in lbehave(last(e))$. By *LSched*, we denote the set of schedulers for the LTS semantics. Given a network M and a scheduler F, we define $Exec_M^F$ as the set of executions starting from M and driven by F.

Since we are interested in weak observational equivalences, that abstract over τ-actions, we introduce the notion of *weak action*.

[1] We still use F to denote a scheduler for the LTS semantics. When it is not clear from the context, we will explicitly state if it refers to the LTS semantics or not.

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

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

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

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

$$\text{(Par)}\ \frac{M \xrightarrow{\gamma} [\![M']\!]_\theta}{M|N \xrightarrow{\gamma} [\![M'|N]\!]_\theta}$$

Table 5: LTS rules for Networks

Definition 4 (Weak Action). *We write $M \stackrel{\alpha}{\Longrightarrow} M'$ if:*

- *$\alpha \neq c!\tilde{v}@K$ and there exists a labelled execution $e \in Exec_M$ such that $e = M \xrightarrow{\alpha_1}_{\theta_1} M_1 ... \xrightarrow{\alpha_k}_{\theta_k} M'$, $\alpha_j = \alpha$ for one $j \in \{1,\ldots,k\}$ and $\alpha_i = \tau$ for all $i \in \{1,\ldots,k\}$ with $i \neq j$.*
- *$\alpha = c!\tilde{v}@K$ and there exists a labelled execution $e \in Exec_M$ such that $e = M \xrightarrow{\alpha_1}_{\theta_1} M_1 ... \xrightarrow{\alpha_k}_{\theta_k} M'$ where for all $i \in \{1,\ldots,k\}$ either $\alpha_i = \tau$ or $\alpha_i = c?\tilde{v}@l$ or $\alpha_i = c!\tilde{v}@H$ and $K = \cup_{\alpha_i = c!\tilde{v}@H} H$ and $K \cap \{l \mid \alpha_i = c?\tilde{v}@l\} = \emptyset$.*

We denote by $\stackrel{\hat{\alpha}}{\Longrightarrow}$ either $\stackrel{\alpha}{\Longrightarrow}$ or a possibly empty sequence of τ-transitions when $\alpha \neq c!\tilde{v}@K$; otherwise $\stackrel{\hat{\alpha}}{\Longrightarrow}$ denotes $\stackrel{\alpha}{\Longrightarrow}$. Notice that $\stackrel{c!\tilde{v}@K}{\Longrightarrow}$ means that a distributed observer receiving an instance of message $\tilde{v}$, at each location in K, in several computational steps, cannot assume that those messages belong to the same broadcast transmission, but they may be different transmissions of the same message. The presence of the weak input actions $\stackrel{c?\tilde{v}@l}{\Longrightarrow}$ is due to the fact that we want to ignore all the inputs executed by each location which is not included in the set of the intended recipients.

We denote by $Exec_M^F(\stackrel{\alpha}{\Longrightarrow}, H)$ the set of executions that, starting from M, according to the scheduler F, lead to a network in the set H by performing $\stackrel{\alpha}{\Longrightarrow}$. Moreover, we define $Prob_M^F(\stackrel{\alpha}{\Longrightarrow}, H) = Prob_M^F(Exec_M^F(\stackrel{\alpha}{\Longrightarrow}, H))$.

Definition 5 (Probabilistic labelled bisimilarity w.r.t. $\mathcal{F}$). *Let M and N be two networks. A relation $\mathcal{R}^{\mathcal{F}}$ over networks is a* probabilistic bisimulation w.r.t. $\mathcal{F}$ *if $M \mathcal{R}^{\mathcal{F}} N$ implies:*

1. *for all* $F \in \mathcal{F}$*, there exists* $F' \in \mathcal{F}$ *such that for all classes* $\mathcal{C}$ *in* $\mathcal{N}/\mathcal{R}^{\mathcal{F}}$ $Prob_M^F(\mathcal{C}) = Prob_N^{F'}(\mathcal{C})$*;*
2. *for all* $F \in \mathcal{F}$ *there exists* $F' \in \mathcal{F}$ *such that for all* α *and for all classes* $\mathcal{C}$ *in* $\mathcal{N}/\mathcal{R}^{\mathcal{F}}$ *it holds* $Prob_M^F(\stackrel{\hat{\alpha}}{\Longrightarrow}, \mathcal{C}) = Prob_N^{F'}(\stackrel{\hat{\alpha}}{\Longrightarrow}, \mathcal{C})$*.*

Probabilistic labelled bisimilarity w.r.t. $\mathcal{F}$*, written* $\approx_p^{\mathcal{F}}$*, is the largest symmetric probabilistic labelled bisimulation w.r.t.* $\mathcal{F}$ *over networks.*

Now, we will show that the bisimulation above is a complete characterisation of the reduction barbed congruence. For this purpose, we define the following auxiliary function Φ that maps each scheduler for the LTS semantics to the unique scheduler of the reduction semantics that mimics its behaviour. Φ is the function $\Phi : \mathit{LSched} \longrightarrow \mathit{Sched}$ such that if $F(e) = (\alpha, [\![N]\!]_\theta) \in \mathit{lbehave}(\mathit{last}(e))$ then $\Phi(F)(\hat{e}) = [\![N]\!]_\theta \in \mathit{behave}(\mathit{last}(e))$. Where $e = M \xrightarrow{\alpha_1}_{\theta_1} M_1 \xrightarrow{\alpha_2}_{\theta_2} M_2 ... \xrightarrow{\alpha_k}_{\theta_k} M_k$ is a labelled execution and $\hat{e} = M \rightarrow_{\theta_1} M_1 ... \rightarrow_{\theta_k} M_k$ the unique unlabelled execution associated to e. With the notation above, the following theorem holds. Its proof is given in Appendix B based on some auxiliary results presented in Appendix A.

Theorem 1. *Given a set of schedulers* $\mathcal{F} \subseteq \mathit{Sched}$*:*

$$M \cong_p^{\mathcal{F}} N \text{ if and only if } M \approx_p^{\Phi^{-1}(\mathcal{F})} N.$$

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. For this purpose we associate an energy cost to LTS transitions as follows:

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

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[2] r regardless of the message being transmitted is observable or not (or even lost). In the same way, if $e = M_0 \xrightarrow{\alpha_1}_{\theta_1} M_1 \xrightarrow{\alpha_2}_{\theta_2} M_2 ... \xrightarrow{\alpha_k}_{\theta_k} M_k$ is an execution then

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

Now 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

[2] Note that considering the radius of the communication channel as the energy cost of the transmitted data is standard.

other execution ending in H. More formally, $Paths^F_M(H) = \{e \in Exec^F_M(H) \mid last(e) \in H$ and $\forall e'$ such that $e <_{prefix} e'$, $e' \notin \mathbf{Paths}^F_M(H)\}$.

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

Definition 6. *Let H be a set of networks. The average energy cost of reaching H from M according to the scheduler F is*

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

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

Definition 7. *Let $\mathcal{H}$ be a countable set of sets of networks and $\mathcal{F}$ be a set of schedulers. We say that N is* more energy efficient than *M w.r.t. $\mathcal{F}$ and $\mathcal{H}$,*

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

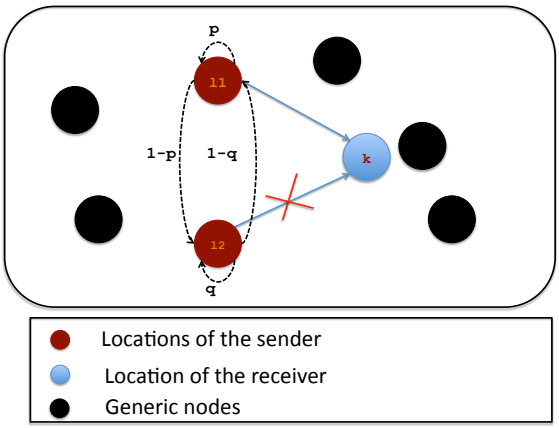
if $N \cong^{\mathcal{F}}_p M$ and, for all schedulers $F \in \mathcal{F}$ and for all $H \in \mathcal{H}$, there exists a scheduler $F' \in \mathcal{F}$ such that $\mathbf{Cost}^{F'}_N(H) \leq \mathbf{Cost}^F_M(H)$.

5 Analysing the SW-ARQ and GBN-ARQ Protocols

High speed data transmission is rapidly becoming an essential requirement of today's wireless networks. Consequently, adaptive modulation and coding (AMC) techniques are increasingly being used in most of 2.5/3g wireless networks in order to increase the transmission rate. At the same time, a wireless channel is error prone due to fading and other propagation impairments. To address this issue, many control schemes have been proposed. In particular the automatic repeat request (ARQ)-based error control is considered as very attractive to counteract the residual errors without using costly error correction codes at the physical layer (see, e.g., [14, 6]). However, portable communication devices must rely on batteries with limited energy to conduct communication. There are three main ARQ protocols: *stop-and-wait (SW)*, *go-back-N (GBN)* and *selective repeat (SR)*. In this section, we use our framework to analyse both SW-ARQ and GBN-ARQ protocols. First we show that the protocols exhibit the same observational behaviour, that is they are bisimilar. Then, we compute and compare their energy consumption under various scenarios depending on the stability of the wireless channel.

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 [6] since correctly received packets may be required to be resent.

Fig. 1: Topology of the network and mobility of s



Modelling of the Protocols.

We consider a single transmitter node using ARQ-based error recovery to communicate with a receiver node over a wireless channel. Transmissions occur in fixed-size time slots. Moreover, we restrict to a one time slot. The SW-ARQ-based protocol transmits one packet per slot while the GBN-ARQ-based one transmits n packets (i.e., the full capacity of the channel). For both protocols, the transmitter continuously sends packets until it detects a transmission error through a NACK feedback. Here, we consider an error-free feedback channel[3] and assume that the acknowledgment (ACK) or negative acknowledgment (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 in our framework both SW-ARQ and GBN-ARQ-based protocols for a communication channel of capacity $n = 3$. We consider a unique stationary receiver. In order to take into account the two states nature of the channel, we model the transmitter as a mobile node *send* ($\langle r, J^s \rangle$) 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 1). The mobility of the sender is modelled by the following two states Markov chain:

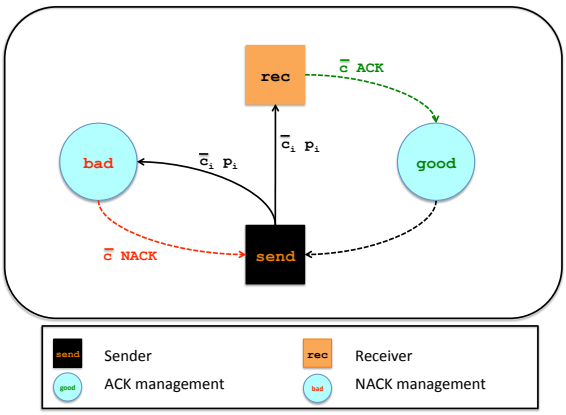
$$J^s = \begin{vmatrix} p & 1-p \\ 1-q & q \end{vmatrix}$$

[3] A very standard assumption [6].

where p and q are the probabilities of the stability of the node in its good and bad states respectively.

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 stationary 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 consuming r energy for each fired packet.

Fig. 2: Structure of the communications s



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, then 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:

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

which upon receiving packet p_i over the channel c_i, sends $NACK(i)$ over the channel c.

GBN-ARQ. 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:

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

where the process $SEND$ is defined as follows.

$$SEND\langle i, x, y\rangle = \bar{c}_{i\emptyset,r}\langle p_i\rangle.c(z)[x = NACK(i-3)]GB\langle i-3\rangle, SEND\langle i+1, y, z\rangle$$

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_1 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. Thus 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 *loose* ($\langle 0, I\rangle$) located at l_1 and executing the process:

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

The full model of GBN protocol is as follows.

$$GBN = send[GB\langle 1\rangle]_{l_1} \mid rec[REC\langle 1\rangle]_{l_1} \mid loose[WAST]_{l_1} \mid \prod_{i\geq 1} b_i[BAD\langle i\rangle]_{l_2}.$$

SW-ARQ. 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.

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

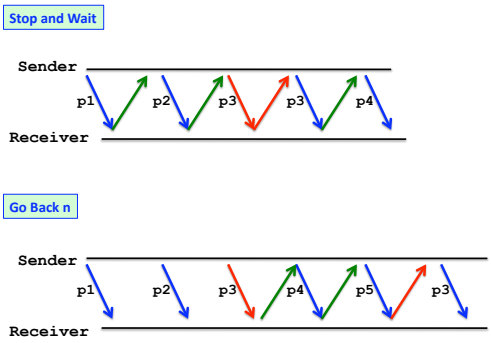
The full protocol is then modelled as the network

$$SW = send[SW\langle 1\rangle]_{l_1} \mid rec[REC\langle 1\rangle]_{l_1} \mid \prod_{i\in I} b_i[BAD\langle i\rangle]_{l_2}.$$

Measuring the Energy Cost of the Protocols.

This section presents 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 w.r.t. 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

Fig. 3: An example of the behaviour of GBN and SW

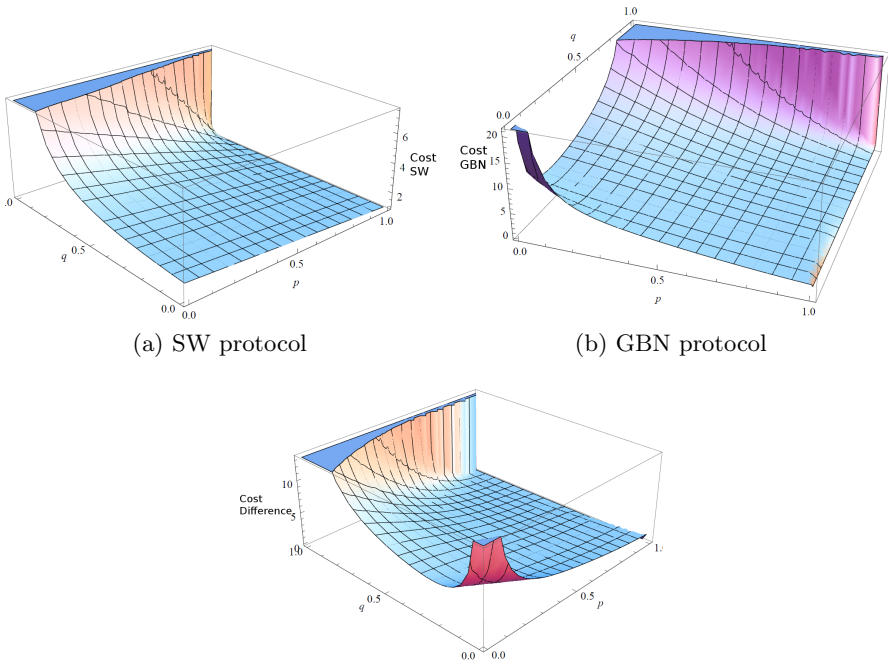


transmit the same packets with the same probabilities. Schedulers constitute an essential feature for modelling communication protocols as they provide freedom in modelling implementation and incomplete knowledge of the system. However, many schedulers could be in fact unrealistic. Consider for example schedulers giving priority to communication actions over movements of the nodes. Such schedulers cancel the two states nature of the communication channel since the latter remains in the same state until there is no longer available communication action. Thus, if the network started with a good channel then all the messages will be transmitted correctly without enduring any lost. In contrast, if it started with a bad channel, then it will be retransmitting indefinitely the first packet since the channel remains always bad. Though, that under such schedulers, both SW-ARQ and GBN-ARQ protocols behave exactly the same way in terms of our observability, they represent however unrealistic implementation scenarios. 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.

Under these assumptions, we can prove the following result which shows that both protocols exhibit the same observable behaviour.

Proposition 1. $GBN \approx_p^{\mathcal{F}_{alt}} SW$



(a) SW protocol (b) GBN protocol

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

Fig. 4: Energy cost functions for SW and GBN protocols and their comparison.

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

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

for some process P and

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

We then 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[4]. The results are summarized in the following propositions and illustrated in Figure 4.

Proposition 2. *If $q \neq 1$ then for all $F \in \mathcal{F}_{alt}$*

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

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

Proposition 3. *If $q \neq 1$ then for all $F \in \mathcal{F}_{alt}$*

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

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 4-(a) and 4-(b)), the GBN protocol deteriorates faster. Indeed, as illustrated by Figure 4-(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.

6 Conclusion

We presented the Probabilistic E-BUM calculus for modeling both connectivity and energy-aware properties of mobile ad-hoc networks.

As a future work we plan to develop an observational preorder which, in one bisimulation step, checkes both observational equivalence and energy-aware preordering. We also plan to extend the model with different metrics and apply it for measuring the level of both sender- and receiver-centered interference.

References

1. E. Bandini and R. Segala. Axiomatizations for probabilistic bisimulation. In *Proc. of the 28th International Colloquium on Automata, Languages and Programming (ICALP '01)*, volume 2076 of *Lecture Notes in Computer Science*, pages 370–381. Springer-Verlag, 2001.
2. M. Bernardo and M. Bravetti. Performance measure sensitive congruences for markovian process algebras. *Theoretical Computer Science*, 290(1):117 – 160, 2003.
3. L. Gallina and S. Rossi. Sender- and receiver-centered interference in wireless ad hoc networks. In *Proc. of IFIP Wireless Days 2010*. IEEE Computer Society Press, 2010.
4. J. Goubault-Larrecq, C. Palamidessi, and A. Troina. A probabilistic applied pi-calculus. In *Proc. of the 5th Asian Symposium on Programming Languages and Systems (APLAS '07)*, volume 4807/2009 of *Lecture Notes in Computer Science*, pages 175–190. Springer-Verlag, 2007.

5. J. Hillston. *A Compositional Approach to Performance Modelling*. Cambridge University Press, 1996.
6. L.B. Le, E. Hossain, and M. Zorzi. 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.
7. C. Priami. Stochasticπ-calculus. *The Computer Journal*, 38(7):578–589, 1995.
8. R.Milner and D. Sangiorgi. Barbed bisimulation. In *Proc. of International Colloquium on Automata, Languages and Programming (ICALP'92)*, volume 623 of *Lecture Notes in Computer Science*, pages 685–695. Springer-Verlag, 1992.
9. S. M. Ross. *Stochastic Processes*. John Wiley & Sons, 2nd edition, 1996.
10. R. Segala. Modeling and verification of randomized distributed real -time systems. Technical report, Massachusetts Institute of Technology, Cambridge, MA, USA, 1996.
11. R. Segala and N.A. Lynch. Probabilistic simulations for probabilistic processes. In *Proc. of the 5th International Conference on Concurrency Theory (CONCUR '94)*, volume 836 of *Lecture Notes in Computer Science*, pages 481–496. Springer-Verlag, 1994.
12. S. Singh, M. Woo, and C.S. Raghavendra. Power-aware routing in mobile ad hoc networks. In *Proc. of the 4th annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '98)*, pages 181–190. ACM, 1998.
13. L. Song and J. Godskesen. Probabilistic mobility models for mobile and wireless networks. In *Theoretical Computer Science*, volume 323 of *IFIP Advances in Information and Communication Technology*, pages 86–100. Springer Boston, 2010.
14. M. Zorzi and R. R. Rao. Error control and energy consumption in communications for nomadic computing. *IEEE Transactions on Computers*, 46(3):279 – 289, 1997.

A Barb and Harmony theorem

In order to prove Theorem 1, we establish the following auxiliary results. Lemma 1 below emphasizes the relationships between the LTS and the reduction semantics.

Lemma 1. *Let* $M \equiv \prod_{i\in I} n_i[P_i]_{l_i}$ *be a network.*

1. *If* $M \xrightarrow{c?\tilde{v}@l} [\![M']\!]_\Delta$, *then* $M \xrightarrow{c?\tilde{v}@l}_\Delta M'$ *and there exists* $j \in I$ *and* P *s.t.* $P_j \equiv c(\tilde{x}).P$ *and* $M' \equiv n_j[P_j\{\tilde{v}/\tilde{x}\}]_{l_j} | \prod_{i\in I-\{j\}} n_i[P_i]_{l_i}$.
2. *If* $M \xrightarrow{c_L!\tilde{v}[l,r]} [\![M']\!]_\Delta$, *then* $M \xrightarrow{c_L!\tilde{v}[l,r]}_\Delta M'$ *and there exists* $j \in I$ *and* P *s.t.* $P_j \equiv \bar{c}_{L,r}\langle\tilde{v}\rangle.P$, $I - \{j\} = I_1 \cup I_2$,
$M \equiv n_j[\bar{c}_{L,r}\langle\tilde{v}\rangle.P]_{l_j} | \prod_{i\in I_1} n_i[c(\tilde{x_i}).P_i]_{l_i} | \prod_{i\in I_2} n_i[P_i]_{l_i}$ *and*
$M' \equiv n_j[P]_{l_j} | \prod_{i\in I_1} n_i[P_i\{\tilde{v}/\tilde{x}\}]_{l_i} | \prod_{i\in I_2} n_i[P_i]_{l_i}$.

Proof. The proof is obtained via an induction on the transition rules of Table 5.
Case 1: $M \xrightarrow{c?\tilde{v}@l} [\![M']\!]_\Delta$

(Rcv): Assume that $M \xrightarrow{c?\tilde{v}@l} [\![M']\!]_\Delta$ is inferred by rule (Rcv), then $M \xrightarrow{c?\tilde{v}@l}_\Delta M'$ and there exists n, P, l, r such that $M \equiv n[P]_l$, $M' \equiv n[P']_l$, $P = c(\tilde{v}).Q$ and $P' = \{\tilde{v}/\tilde{x}\}Q$. Hence the result holds since by considering the empty set I, we have:
$M \equiv n[c(\tilde{v}).Q]_l | \prod_{i\in I} n_i[P_i]_{l_i}$ and $M' \equiv n[\{\tilde{v}/\tilde{x}\}Q]_l | \prod_{i\in I} n_i[P_i]_{l_i}$.

(Par): Now if $M|N \xrightarrow{c?\tilde{v}@l} [\![M|N']\!]_\Delta$ is inferred by rule (Par), where $M \xrightarrow{c?\tilde{v}@l}_\Delta M'$, then $M|N \xrightarrow{c?\tilde{v}@l}_\Delta M'|N$. And by induction hypothesis we have

$$M \equiv n_j[c(\tilde{x}).P']_{l_j} | \prod_{\imath \in I-\{j\}} n_i[P_i]_{l_i}$$

and

$$M' \equiv n_j[P'\{\tilde{v}/\tilde{x}\}]_{l_j} | \prod_{\imath \in I-\{j\}} n_i[P_i]_{l_i}.$$

Let $N \equiv \prod_{\imath \in I'} n_i[P_i]_{l_i}$. By applying rule (Struct -Cxt- Par) we obtain

$$M|N \equiv n_j[c(\tilde{x}).P']_{l_j} | \prod_{i \in I \cup I'-\{j\}} n_i[P_i]_{l_i}$$

and

$$M'|N \equiv n_j[P'\{\tilde{v}/\tilde{x}\}]_{l_j} | \prod_{i \in I \cup I'-\{j\}} n_i[P_i]_{l_i}.$$

The other cases follow straightforwardly from congruence rules of the reduction relation.

Case 2: $M \xrightarrow{c_L!\tilde{v}[l,r]} [\![M']\!]_\Delta$

(Snd): If $M \xrightarrow{c_L!\tilde{v}[l,r]} [\![M']\!]_\Delta$ is inferred by the rule (Snd), then $M \xrightarrow{c_L!\tilde{v}[l,r]}_\Delta M'$ and $M \equiv n[\bar{c}_{L,r}\langle\tilde{v}\rangle.P]_l$. Since $\bar{c}_{L,r}\langle\tilde{v}\rangle.P \xrightarrow{c_{\bar{L},r}\tilde{v}} P'$, then the lemma holds if I_1 and I_2 are empty, because

$$M \equiv n[\bar{c}_{L,r}\langle\tilde{v}\rangle.P]_l | \prod_{i \in I_1} n_i[P_i\{\tilde{v}/\tilde{x}\}]_{l_i} | \prod_{i \in I_2} n_i[P_i]_{l_i}$$

and

$$M' \equiv n[P]_l | \prod_{i \in I_1} n_i[P_i\{\tilde{v}/\tilde{x}\}]_{l_i} | \prod_{i \in I_2} n_i[P_i]_{l_i}$$

(Bcast): If $M|N \xrightarrow{c_L!\tilde{v}[l,r]} [\![M'|N']\!]_\Delta$, then because $M \xrightarrow{c_L!\tilde{v}[l,r]} [\![M']\!]_\Delta$ and $N \xrightarrow{c?\tilde{v}@l'} [\![N']\!]_\Delta$, with $d(l,l') \leq r$, $M \xrightarrow{c_L!\tilde{v}[l,r]}_\Delta M'$ and $N \xrightarrow{c?\tilde{v}@l'}_\Delta N'$. By induction hypothesis we have:

$$M \equiv n[\bar{c}_{L,r}\langle\tilde{v}\rangle.P]_l | \prod_{i \in I_1} n_i[c(\tilde{x_i}).P_i]_{l_i} | \prod_{i \in I_2} n_i[P_i]_{l_i}$$

and

$$M' \equiv n[P]_l | \prod_{i \in I_1} n_i[P_i\{\tilde{v}/\tilde{x}\}]_{l_i} | \prod_{i \in I_2} n_i[P_i]_{l_i}$$

for some n, P, $\tilde{v}$, l and some (possibly empty) sets I_1 and I_2. Similarly we have:

$$N \equiv n_j[c(\tilde{x_j}).P_j]_{l_j} | \prod_{i \in I_3} n_i[P_i]_{l_i}$$

and

$$N' \equiv n_j[P_j\{\tilde{v}/\tilde{x_j}\}]_{l_j} | \prod_{i \in I_3} n_i[P_i]_{l_i}$$

for some n_j, P_j, l_j, and a (possibly empty) set I_3 . By applying rule (Struct - Par - Cxt):

$$M|N \equiv n[\bar{c}_{L,r}\langle\tilde{v}\rangle.P]_l|\prod_{i\in I_1\cup\{j\}} n_i[c(\tilde{x_i}).P_i]_{l_i}|\prod_{i\in I_2\cup I_3} n_i[P_i]_{l_i}$$

and

$$M'|N' \equiv n[P]_l|\prod_{i\in I_1\cup\{j\}} n_i[P_i\{\tilde{v}/\tilde{x}\}]_{l_i}|\prod_{i\in I_2\cup I_3} n_i[P_i]_{l_i}$$

The proofs of the remaining cases are similar to the proof of the first part of the lemma.

Lemma 2 (≡ respects transitions). *If $M \equiv N$ then $lbehave(N) = lbehave(N)$.*

Proof. We proceed by induction on the depth of the inference $M \xrightarrow{\gamma} [\![M']\!]_\theta$ since if $lbehave(M) = lbehave(N)$ and $M \xrightarrow{\gamma} [\![M']\!]_\theta$ then $N \xrightarrow{\gamma} [\![M']\!]_\theta$.

Following we prove the result in the special case that $M \equiv N$ is due to a single application of structural rules. The general case follows just by iterating the special case.

(Par) Suppose that $M \xrightarrow{\gamma} [\![M']\!]_\theta$ is inferred by an application of rule (Par), with $M \equiv M_1|M_2$ and $M_1 \xrightarrow{\gamma} [\![M_1']\!]_\theta$. Now there are two cases in which $M_1|M_2 \equiv N$ could holds.
- $N \equiv M_2|M_1$. In this case we can again use rule (Par) to deduce $N \xrightarrow{\gamma} [\![M_2|M_1']\!]_\theta$ which is equivalent to $[\![M_1' \mid M_2]\!]_\theta$ as required.
- Now suppose that a single rule of structural congruence is used within M_1, so that $M_1 \equiv N_1$ and N is $N_1|M_2$. Then, since $M_1 \xrightarrow{\gamma} [\![M']\!]_\theta$ is inferred by a shorter inference, by inductive hypotesis we have $N_1 \xrightarrow{\gamma} [\![M']\!]_\theta$. No, by applying rule (Par) we obtain $N_1|M_2 \xrightarrow{\gamma} [\![M']\!]_\theta$ as required.

(Bcast) Suppose $M \xrightarrow{\gamma} [\![M']\!]_\Delta$ has been inferred by an application of rule (Bcast), where $\gamma = c_L!\tilde{v}[l,r]$. $M \equiv M_1|M_2$ where $M_1 \xrightarrow{c_L!\tilde{v}[l,r]} [\![M_1']\!]_\Delta$ and $M_2 \xrightarrow{c?\tilde{v}@l'} [\![M_2']\!]_\Delta$. Again there are many ways in which $M_1|M_2 \equiv N$.
- For example suppose that N is $M_2|M_1$. In this case we can again use rule (Bcast) to deduce $N \xrightarrow{\gamma} [\![M_2'|M_1']\!]_\Delta$ which is equivalent to $[\![M']\!]_\Delta$ as required.

The proofs for the remaining cases are similar to the case of the rule (Par) above.

Theorem 2 (Harmony theorem).

1. *If $M \xrightarrow{\tau} [\![M']\!]_\theta$ then $M \longrightarrow [\![M']\!]_\theta$.*
2. *If $M \longrightarrow [\![M']\!]_\theta$ then $M \xrightarrow{\tau} [\![M']\!]_\theta$.*
3. *if $M \downarrow_c^F$ then there exists $K \subseteq Loc$, a message $\tilde{v}$ and a scheduler F' such that $F'(e) = (\xrightarrow{c!\tilde{v}@K}, F(e))$ for all $e \in Exec_M^{F'}$ s.t. $last(e) = M$.*

Proof. We prove the first two points of the theorem by induction on the derivation $M \xrightarrow{\tau} [\![M']\!]_\theta$.

1. Suppose that the τ-action has been generated by an application of the rule (Lose). In this case we have $\dfrac{M \xrightarrow{c_L!\tilde{v}[l,r]} [\![M']\!]_\Delta}{M \xrightarrow{\tau} [\![M']\!]_\Delta}$, then, by invoking Lemma 1 we have:

$$M \equiv n_j[\bar{c}_{L,r}\langle\tilde{v}\rangle.P]_{l_j} \,|\, \prod_{i\in I_1} n_i[c(\tilde{x_i}).P_i]_{l_i} \,|\, \prod_{i\in I_2} n_i[P_i]_{l_i}$$

and

$$M' \equiv n_j[P]_{l_j} \,|\, \prod_{i\in I_1} n_i[P_i\{\tilde{v}/\tilde{x}\}]_{l_i} \,|\, \prod_{i\in I_2} n_i[P_i]_{l_i}$$

for some (possibly empty) sets I_1 and I_2. By applying rules (R-Bcast) and (R-Par) we have:

$$n_j[\bar{c}_{L,r}\langle\tilde{v}\rangle.P]_{l_j} \,|\, \prod_{i\in I_1} n_i[c(\tilde{x_i}).P_i]_{l_i} \,|\, \prod_{i\in I_2} n_i[P_i]_{l_i} \longrightarrow [\![n_j[P]_{l_j} \,|\, \prod_{i\in I_1} n_i[P_i\{\tilde{v}/\tilde{x}\}]_{l_i} \,|\, \prod_{i\in I_2} n_i[P_i]_{l_i}]\!]_\Delta$$

and, by applying (R-Struct), we obtain $M \longrightarrow [\![M']\!]_\Delta$, as required.
Suppose now that the τ-action is generated by the rule (Move), i.e.,

$$\frac{}{n[P]_l \xrightarrow{\tau} [\![n[P]_l]\!]_{\mu_l^n}}$$

then, thanks to the rule (R-Move) we have:

$$\frac{}{n[P]_l \longrightarrow [\![n[P]_l]\!]_{\mu_l^n}}.$$

The other cases (e.g. rule (Move)) follow straightforwardly from congruence rules of the reduction relation.

2. If we consider the rules where a τ-action in LTS semantics corresponds to a reduction (for example (Move) and (R-Move)), then the proof is obvious. We therefore prove the other cases.
Now Suppose that the derivation $M \longrightarrow [\![M']\!]_\theta$ is generated by the rule (R-Bcast), i.e.,

$$\frac{\forall i \in I.d(l,l_i) \le r,\ |\tilde{x_i}| = |\tilde{v}|}{n[\bar{c}_{L,r}\langle\tilde{v}\rangle.P]_l \mid \prod_{i\in I} n_i[c(\tilde{x_i}).P_i]_{l_i} \rightarrow [\![n[P]_l \mid \prod_{i\in I} n_i[P_i\{\tilde{v}/\tilde{x_i}\}]_{l_i}]\!]_\Delta}$$

Then, by applying successively rules (Snd), (Rcv) and (Bcast) we obtain:

$$\frac{\dfrac{\bar{c}_{L,r}\langle\tilde{v}\rangle.P \xrightarrow{\bar{c}_{L,r}\tilde{v}} P}{n[\bar{c}_{L,r}\langle\tilde{v}\rangle.P]_l \xrightarrow{c_L!\tilde{v}[l,r]} [\![n[P]_l]\!]_\Delta} \quad \dfrac{c(\tilde{x_1}).P_1 \xrightarrow{c\tilde{v}} P_1\{\tilde{v}/\tilde{x_1}\}}{n_1[c(\tilde{x_1}).P_1]_{l_1} \xrightarrow{c?\tilde{v}@l_1} [\![n_1[P_1\{\tilde{v}/\tilde{x_1}\}]_{l_1}]\!]_\Delta} \ d(l,l_1) \le r}{n[\bar{c}_{L,r}\langle\tilde{v}\rangle.P]_l | n_1[c(\tilde{x_1}).P_1]_{l_1} \xrightarrow{c_L!\tilde{v}[l,r]} [\![n[P]_l | n_1[P_1\{\tilde{v}/\tilde{x_1}\}]_{l_1}]\!]_\Delta}.$$

By applying $|I| - 1$ times rule (Bcast) and one time rule (Lose) we get

$$n[\bar{c}_{L,r}\langle\tilde{v}\rangle.P]_l|\prod_{i\in I} n_i[c(\tilde{x_i}).P_i]_{l_i} \xrightarrow{\tau} [\![n[P]_l|\prod_{i\in I} n_i[P_i\{\tilde{v}/\tilde{x_i}\}]_{l_i}]\!]_\Delta$$

as required.
If the derivation $M \longrightarrow [\![M']\!]_\theta$ is instead generated by the rule (R-Struct), i.e.,

$$\frac{M \equiv N \; N \rightarrow [\![M']\!]_\theta}{M \rightarrow [\![M']\!]_\theta}$$

then by induction hypothesis $N \xrightarrow{\tau} [\![M']\!]_\theta$, and from Lemma 2 (since $M \equiv N$) we obtain $M \xrightarrow{\tau} [\![M']\!]_\theta$.

3. The last point of the theorem follows from the definition of barb and Lemma 1.

B Proof of Theorem 1

Now back to the proof of our main theorem, i.e.,

$$M \cong_p^{\mathcal{F}} N \text{ if and only if } M \approx_p^{\Phi^{-1}(\mathcal{F})} N.$$

We start by showing that bisimulation is:

1. reduction colosed
2. contextual
3. barb preserving

The reduction closure follows from the definition of bisimulation, since for all classes $\mathcal{C}$ in $\mathcal{N}/\mathcal{R}$ and for all $F \in \Phi^{-1}(\mathcal{F})$, there exists $F' \in \Phi^{-1}(\mathcal{F})$ such that $Prob_M^F(\mathcal{C}) = Prob_N^{F'}(\mathcal{C})$.

Now for the contextuality, we need to show that for each context $C[\cdot]$ and for each $\mathcal{G} \in \mathit{LSched}$:

$$\mathcal{R} = \{(C[M], C[N]) \mid M \approx_p^{\mathcal{G}} N\}$$

is a probabilistic bisimulation.

We proceed by induction on the structure of $C[\cdot]$. We start by defining:

$$Exec_M^F(n, H) = \{e \in Exec_M^F \mid last(e^j) \in H \text{ for some } j \leq n\}$$

as the set of executions from M and crossing a state in H after at most n execution steps. We denote $Prob_M^F(n, H)$ as the probability $Prob_M^F(Exec_M^F(n, H))$. Similarly we define $Exec_M^F(\stackrel{\hat{\alpha}}{\Longrightarrow}, n, \mathcal{C})$ and $Prob_M^F(\stackrel{\hat{\alpha}}{\Longrightarrow}, n, \mathcal{C})$ where the lenght of the $\alpha-$executions are bound by n.

Now let us write $M \approx_{p,n}^{\mathcal{G}} N$ when $\forall F \, \exists F'$ s.t. $\forall \mathcal{C} \in \mathcal{N}/\mathcal{R}$ we have:

1. $Prob_M^F(n, \mathcal{C}) = Prob_N^{F'}(n, \mathcal{C})$

2. $\forall \alpha$, $Prob^F_M(\stackrel{\hat{\alpha}}{\Longrightarrow}, n, \mathcal{C}) = Prob^{F'}_N(\stackrel{\hat{\alpha}}{\Longrightarrow}, n, \mathcal{C})$

Since

$$Prob^F_M(\mathcal{C}) = \lim_{n\to\infty} Prob^F_M(n, \mathcal{C})$$

and

$$Prob^F_M(\stackrel{\hat{\alpha}}{\Longrightarrow}, \mathcal{C}) = \lim_{n\to\infty} Prob^F_M(\stackrel{\hat{\alpha}}{\Longrightarrow}, n, \mathcal{C})$$

we need just to show, via induction on n, that for all $n \geq 0$, for all N and M s.t. $M \approx^{\mathcal{G}}_{p,n} N$ we have:

$$\text{for all context } C[\cdot] \text{ , } C[M] \approx^{\mathcal{G}}_{p,n} C[N].$$

Following we prove the first point of $\approx^{\mathcal{G}}_{p,n}$, that is $\forall F\, \exists F'$ s.t.

$$Prob^F_{C[M]}(1, \mathcal{C}) = Prob^{F'}_{C[N]}(1, \mathcal{C})$$

$\forall M$, N s.t. $M \approx^{\mathcal{G}}_{p,n}$, context $C[\cdot]$ and $\mathcal{C} \in \mathcal{N}/\mathcal{R}$.

We prove the inductive hypothesis for $n = 1$ since the case when $n = 0$ is trivial.

1. if $C[\cdot] = \cdot \mid \mathbf{0}$ (empty network), then the result follows straightforwardly from the fact that $M \approx^{\mathcal{G}}_{p} N$.
2. if $C[\cdot] = \cdot \mid T$ for some $T \neq \mathbf{0}$

We proceed by induction on the structure of actions. Since no transition rule in the operational semantics eliminates the operator $\mid$ we can confine our attention to $\mathcal{C}$ of the form $\mathcal{C}_1 \mid \mathcal{C}_2$.

(a) $\alpha = \tau$

There are two cases to consider:

i α is a τ-action representing a node's movement: if the moving node belongs to M (respectively N) then the result follows from the fact that $M \approx^{\mathcal{G}}_{p} N$, that means if $M \stackrel{\tau}{\rightarrow}_{\mu^n_k} M'$ then $\mathcal{C} \equiv [M']_{\approx^{\mathcal{G}}_{p}} \mid [O]_{\approx^{\mathcal{G}}_{p}}$ where $[M]_{\approx^{\mathcal{G}}_{p}}$ means the equivalence class of M according to $\approx^{\mathcal{G}}_{p}$. Clearly $\forall F$, $\exists F'$ s.t.

$$Prob^F_{M|O}(1, \mathcal{C}) = Prob^{F'}_{N|O}(1, \mathcal{C}) = [M']_{\mu^n_k}(M')$$

since $M \approx^{\mathcal{G}}_{p} N$.

If the moving node belongs to O then $\mathcal{C} \equiv [M]_{\approx^{\mathcal{G}}_{p}} \mid [O']_{\approx^{\mathcal{G}}_{p}}$ where $O \stackrel{\tau}{\rightarrow}_{\mu^n_k} O'$. Again the same argumentation allows us to conclude.

ii α is a τ-action representing a loose action. In this case $\mathcal{C} \equiv [M']_{\approx^{\mathcal{G}}_{p}} \mid [O']_{\approx^{\mathcal{G}}_{p}}$ where $M \stackrel{\alpha_1}{\longrightarrow}_{\Delta} M'$ and $O \stackrel{\alpha_2}{\longrightarrow}_{\Delta} O'$ and
1) $\alpha_1 = \tau$ and α_2 is an input action, if the sender node belongs to M
2) $\alpha_2 = \tau$ and α_1 is an input action, if the sender node belongs to O.
In both cases, since $M \approx^{\mathcal{G}}_{p} N$, then $\forall F\, \exists F'$ s.t.

$$Prob^F_{M|O}(1, \mathcal{C}) = Prob^{F'}_{N|O}(1, \mathcal{C}) = 1.$$

(b) $\alpha = c?\tilde{v}@l$ for some channel c, some message $\tilde{v}$ and some location l. The same argument as in (a) ii allows to conclude.

(c) $\alpha = c!\tilde{v}@K$ for some channel c, some message $\tilde{v}$ and some set of locations K There are two cases to consider.

i α belongs to an interaction between M and O In this case $\mathcal{C} \equiv [M']_{\approx_p^{\mathcal{G}}} \mid [O']_{\approx_p^{\mathcal{G}}}$ where $M \xrightarrow{\alpha_1}_{\Delta} M'$ and $O \xrightarrow{\alpha_2}_{\Delta} O'$ and
1) $\alpha_1 = c!\tilde{v}@K$ and $\alpha_2 = c?\tilde{v}@l$ for some $l \in \mathbf{Loc}$, if the sender node belongs to M
2) $\alpha_2 = c!\tilde{v}@K$ and $\alpha_1 = c?\tilde{v}@l$ for some $l \in \mathbf{Loc}$, if the sender node belongs to O.
In both cases, since $M \approx_p^{\mathcal{G}} N$, then $\forall F\, \exists F'$ s.t.

$$Prob^{F}_{M|O}(1, \mathcal{C}) = Prob^{F'}_{N|O}(1, \mathcal{C}) = 1.$$

ii M and O do not interact. If α belongs to M, then $C \equiv [M']_{\approx_p^{\mathcal{G}}} \mid [O]_{\approx_p^{\mathcal{G}}}$, where $M \xrightarrow{\alpha}_{\Delta} M'$. Since $M \approx_p^{\mathcal{G}} N$, $\forall F\, \exists F'$ s.t. $Prob^{F}_{M|O}(1, \mathcal{C}) = Prob^{F'}_{N|O}(1, \mathcal{C}) = 1$. If the output action belongs to O then $C \equiv [M]_{\approx_p^{\mathcal{G}}} \mid [O']_{\approx_p^{\mathcal{G}}}$, where $O \xrightarrow{\alpha}_{\Delta} O'$ and the same argumentation allows us to conclude.

Induction step

Since we estabilished our induction hypothesis, we proceed now with the induction step.

Suppose that for all $k \leq n$, and $\forall F\, \exists F'$ s.t.

$$Prob^{F}_{M|O}(k, \mathcal{C}) = Prob^{F'}_{N|O}(k, \mathcal{C})\ \forall \mathcal{C} \equiv \mathcal{C}_1 \mid \mathcal{C}_2$$

we need to prove that

$$Prob^{F}_{M|O}(n+1, \mathcal{C}) = Prob^{F'}_{N|O}(n+1, \mathcal{C})$$

but

$$Prob^{F}_{M|O}(n{+}1, \mathcal{C}) = \sum_{k=1}^{n} \sum_{\mathcal{C}'_1|\mathcal{C}'_2} Prob^{F}_{M|O}(k, \mathcal{C}'_1 \mid \mathcal{C}'_2) \cdot Prob^{F}_{Rep_{\mathcal{C}_1}|Rep_{\mathcal{C}_2}}(n{+}1{-}k, \mathcal{C}_1 \mid \mathcal{C}_2)$$

where $Rep_{\mathcal{C}_1}$ and $Rep_{\mathcal{C}_2}$ are representatives of the respective classes. But, since both k and $n+1-k$ are $\leq n$, the induction hypothesis allows us to conclude.

Finally for the barb preserving we must prove that if $M \approx_p^{\Phi^{-1}(\mathcal{F})} N$, then for all $\hat{F} \in \mathcal{F}$ such that $M \Downarrow_p^{\hat{F}} c$ there exists $\hat{F}' \in \mathcal{F}$ such that $N \Downarrow_p^{\hat{F}'} c$.

We know that $M \Downarrow_p^{\hat{F}} c$ means that $Prob^{\hat{F}}_{M}(H) = p$, where $H = \{M' \mid M \rightarrow^{*} M' \downarrow_c^{\hat{F}}\}$ and, by the Harmony theorem, $\exists F \in \Phi^{-1}(\mathcal{F})$ s.t. $\forall e \in Exec^{F}_{M}(\mathcal{C})$,

$$\hat{F}(\hat{e}) = (\tau, F(e)) \text{ and } last(e) = last(\hat{e}).$$

Then $Prob^{\hat{F}}_M(H) = Prob^F_M(\Longrightarrow, H) = p$. We know that there exist $\tilde{v_1}, ..., \tilde{v_n}$ messages and $K_1, ..., K_n$ sets of locations such that $\forall M' \in H \ M' \xrightarrow{c!\tilde{v_i}@K_i}$ for some $i \in [1-n]$. Now we can find a set of equivalence classes $\mathcal{C}_1$ ... $\mathcal{C}_n$ such that for each $M' \in H$ if $M' \xrightarrow{c!\tilde{v_i}@K_i}$ then $M' \in \mathcal{C}_i$, and $H \subseteq \cup_{i\in[1-n]}\mathcal{C}_i$. Since M and N are bisimilar there exists F' in $\Phi^{-1}(\mathcal{F})$ such that, for each equivalence class $\mathcal{C}_i$, $Prob^F_M(\Longrightarrow, \mathcal{C}_i) = Prob^{F'}_N(\Longrightarrow, \mathcal{C}_i)$, and

$$\sum_{i\in[1-n]} Prob^F_M(\Longrightarrow, \mathcal{C}_i) = p = \sum_{i\in[i-n]} Prob^{F'}_N(\Longrightarrow, \mathcal{C}_i).$$

Since $\mathcal{C}_1, ...\mathcal{C}_n$ are equivalence classes with respect to the bisimulation, for each $N' \in \mathcal{C}_i$ such that $N \Longrightarrow N'$ there exists $M' \in H$ such that $M' \xrightarrow{c!\tilde{v_i}@K_i} [\![M'']\!]_\Delta$ driven by F, then, since $M' \approx^{\mathcal{F}}_p N'$, $N' \xrightarrow{c!\tilde{v_i}@K_i} [\![N'']\!]_\Delta$ according F'. Now consider the scheduler $\hat{F'} = \Phi(F')$. Then

i $\hat{F'}(\hat{e}) = [\![N']\!]_\theta \ \forall e$ s.t. $last(e) \in \mathcal{C}_i$ and $F'(e) = (\alpha, [\![N']\!]_\theta)$
ii $last(\hat{e}) \not\downarrow^{\hat{F'}}_c \ \ \forall c$ and $\forall e \in Exec^{F'}_N$ s.t. $last(e) \notin \mathcal{C}_i$

If $H' = \{N' \mid N \Longrightarrow N' \downarrow^{\hat{F'}}_c\}$, it follows that:
$p = \sum_{i\in[1-n]} Prob^F_M(\Longrightarrow, \mathcal{C}_i) = \sum_{i\in[i-n]} Prob^{F'}_N(\Longrightarrow, \mathcal{C}_i)$
$= \sum_{i\in[i-n]} Prob^{\hat{F}}_N(H') = Prob^{\hat{F'}}_N(H') \Rightarrow N \Downarrow^{\hat{F'}}_p c$, as required. Hence, we just showed $\approx^{\Phi^{-1}(\mathcal{F})}_p \Rightarrow \cong^{\mathcal{F}}_p$.

Now, we prove the reverse implication (i.e. $\cong^{\mathcal{F}}_p \Rightarrow \approx^{\Phi^{-1}(\mathcal{F})}_p$). Let M and N be two networks such that $M \cong^{\mathcal{F}}_p N$, then we have to prove that:

1. for each scheduler F in $\Phi^{-1}(\mathcal{F})$ there exists F' in $\Phi^{-1}(\mathcal{F})$ such that, for all equivalence classes $\mathcal{C}$
$$Prob^F_M(\mathcal{C}) = Prob^{F'}_M(\mathcal{C}).$$
2. for all schedulers F in $\Phi^{-1}(\mathcal{F})$ there exists F' in $\Phi^{-1}(\mathcal{F})$ such that, for each equivalence class $\mathcal{C}$ and for each action α
$$Prob^F_M(\overset{\hat{\alpha}}{\Longrightarrow}, \mathcal{C}) = Prob^{F'}_M(\overset{\hat{\alpha}}{\Longrightarrow}, \mathcal{C}).$$

The first point follows straightforwardly from the definition of bisimulation and harmony theorem.

Now onto the second point. Let F in $\Phi^{-1}(\mathcal{F})$ be a scheduler, we must prove that there exists F' in $\Phi^{-1}(\mathcal{F})$ such that for all α and for all $\mathcal{C}$ in $\mathcal{N}/_{\cong^{\mathcal{F}}_p}$, $Prob^F_M(\overset{\hat{\alpha}}{\Longrightarrow}, \mathcal{C}) = Prob^{F'}_N(\overset{\hat{\alpha}}{\Longrightarrow}, \mathcal{C})$.

Consider the scheduler $\hat{F} = \Phi(F)$ mimicking F. It is clear that for all α and for all $\mathcal{C}$ s.t. $M \overset{\hat{\alpha}}{\Longrightarrow} M'$ for some $M' \in \mathcal{C}$:

$$Prob^{\hat{F}}_M(\mathcal{C}) = Prob^F_M(\overset{\hat{\alpha}}{\Longrightarrow}, \mathcal{C}) \tag{1}$$

and, since $M \cong_p^{\mathcal{F}} N$, there exists $\hat{F}'$ s.t.:

$$Prob_N^{\hat{F}'}(\mathcal{C}) = Prob_M^{\hat{F}}(\mathcal{C}). \tag{2}$$

If $\alpha = c!\tilde{v}@K$ where $K = \{l_1, .., l_h\}$ and $Prob_M^{\hat{F}}(\mathcal{C}) = Prob_M^{F}(\stackrel{\hat{\alpha}}{\Longrightarrow}, \mathcal{C})$, we can build a context allowing us to prove that also N eventually performs α with the same probability $Prob_M^{\hat{F}}(\mathcal{C})$. Indeed, consider the following context

$$C[\cdot] = \cdot \mid \prod_{i \in [1-h]} n_i[c(\tilde{x}).\bar{d}_{i\,k,r_i}\langle\tilde{x}\rangle]_{l_i} \mid m[d_1(\tilde{x_1})...d_h(\tilde{x_h}).\bar{ok}_{k,r}\langle\tilde{x}\rangle]_k$$

where all the nodes n_i and m are stationary, and d_i and ok are fresh channels for both M and N. Since $Prob_M^{F_1}(\stackrel{\hat{\alpha}}{\Longrightarrow}, \mathcal{C}) = Prob_M^{\hat{F}}(\mathcal{C})$, there exists $\hat{F}_1$ s.t. $C[M]\Downarrow_p^{\hat{F}_1} ok$ where $p = Prob_{C[M]}^{\hat{F}_1}(\hat{M}) = Prob_M^{\hat{F}}(\mathcal{C})$, $\hat{M} = M' \mid m[\bar{ok}_{k,r}\langle\tilde{x}\rangle]_k$ and $M' \in \mathcal{C}$.

Now since barbed congruence is a contextual relation, there exists $\hat{F}_1'$ s.t. $C[N]\Downarrow_p^{\hat{F}_1'} ok$ and $Prob_{C[N]}^{\hat{F}_1'}(\hat{N}) = p$ where $\hat{N} = N' \mid m[\bar{ok}_{k,r}\langle\tilde{x}\rangle]_k$. The presence of the barb on channel ok ensures that each node located at l_1, ..., l_h have been able to received the message sent by N on the channel c. Hence $N \stackrel{c!\tilde{v}@K}{\Longrightarrow} N'$.

Moreover, since the nodes n_i and m are stationary and their actions do not change the probability. Clearly, there exists F' in $\Phi^{-1}(\{\hat{F}'\})$ s.t.

$$Prob_N^{F'}(\stackrel{\hat{\alpha}}{\Longrightarrow}, \mathcal{C}) = p. \tag{3}$$

Finally, we can conclude by combining (1), (2), and (3).