Interference-Sensitive Preorders for MANETs

Michele Bugliesi, Lucia Gallina, Andrea Marin, Sabina Rossi, Sardaouna Hamadou

Interference-sensitive Preorders for MANETs

Michele Bugliesi, Lucia Gallina, Andrea Marin, Sabina Rossi
DAIS, Università Ca' Foscari Venezia, Italy
E-mail: {*bugliesi,lgallina,marin,srossi*}*@dais.unive.it*

Sardaouna Hamadou
INRIA Saclay and LIX, École Polytechnique, France
E-mail: *sardaouna.hamadou@polymtl.ca*

Abstract—Connectivity and communication interference are two key aspects in mobile ad-hoc networks (MANETs). We propose a process algebraic model targeted at the analysis of both such aspects of MANETs. The framework includes a probabilistic process calculus and a suite of analytical techniques based on a probabilistic observational congruence and an interference-sensitive preorder. The observational congruence allows us to verify whether two networks exhibit the same behaviour. The preorder makes it possible to evaluate the interference level of different, behaviourally equivalent, networks. We show our framework at work on the analysis of the well-known *Alternating Bit Protocol*, contrasting the behavior of the standard implementation of the protocol against an alternative implementation that exploits an ideal interference cancellation scheme for CDMA transmissions.

Keywords-process algebras; manets; interference.

I. INTRODUCTION

Mobile ad-hoc networks are systems of mobile devices communicating over wireless links without a pre-established connectivity structure. Connectivity and communication interference are two key aspects in such networks. Node mobility is unconstrained: each device in a MANET moves autonomously, thereby seamlessly modifying the underlying topology, and hence creating the need for highly dynamic routing algorithms to ensure the desired level of connectivity among the network devices. Communication interference, in turn, is particularly challenging in MANETs due to the half-duplex nature of wireless channels, which makes it impossible for a transmitter to atomically detect the presence of other, conflicting transmitters on the same channel. As a consequence, any interference among multiple transmissions may only be detected by receivers located at the intersection of the transmission ranges of the emitters. Controlling interference in the presence of node mobility is even more complex due to the dynamic structure of the underlying network topology. While there exist ad-hoc protocols that address these problems, e.g., [1], [2], interferences remain one of the pivotal aspects in the design of MANETs.

Drawing on earlier work on the subject (by the authors [3], [4], and by others [5], [6]), in the present paper we introduce a calculus to provide a formal basis for the analysis of connectivity and the evaluation of interference in MANETs. Like its predecessors [3], [6], 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 as in [3]. Instead, wireless synchronization is nondeterministic, and controlled by (sequential) processes inside the nodes: each transmission broadcasts a message at a given radio frequency and within a given transmission range. Importantly, multiple nodes may simultaneously transmit along the same channel, over overlapping areas: the calculus provides for an explicit representation of the collisions that may occur at the receivers which lie within the transmission range of different senders.

The semantics of the calculus is inspired by Segala's probabilistic automata [7] driven by schedulers to resolve the nondeterministic choice among the probability distributions over target states. We define a probabilistic observational congruence in the style of [8] to equate networks exhibiting the same observable behavior. As in [4], [3], and in contrast to [6], 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 and interference at different areas within a network. We give a coinductive characterization of observational congruence based on a labelled transition semantics. Then, we introduce interference-sensitive preorders over networks to measure the relative interference level of different, but observationally equivalent, networks. We show our framework at work on the analysis of the well-known *Alternating Bit Protocol*.

To our knowledge, the present paper is the first to provide a comprehensive framework for the behavioral analysis and a quantitative assessment of interference for wireless networks in the presence of node mobility. Previous process algebraic models do not deal with interferences [9], [10], [11], [12], [3], or only accounted for them in static networks without providing quantitative metrics [5]. Similarly, existing frameworks based on Petri Nets and queueing networks fall short of accounting for node mobility while maintaining a good accuracy in specifying the protocol designs [13], [14].

Plan of the paper. Section II introduces the calculus and its observational semantics. Section III defines the LTS semantics and the associated notion of probabilistic bisimilarity. Section IV develops a technique for measuring the level of interference. Section V applies that technique to the analysis of the case study. Section VI concludes the paper.
All proofs, omitted here, can be found in [15].

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

Table I: Syntax

II. THE CALCULUS

The peculiarity of the calculus presented here is the non-atomicity of the output and input actions, which we define after [5] to capture the presence of interference caused by the simultaneous transmissions of two (or more) nodes using the same channel in a common transmission area.

We use letters c for *channels*, n for *nodes*, l for *locations*, r for *transmission radii*, x and y 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 $\mathcal{N}$ for the set of networks, and $\mathbf{C}$ for the set of channels. We write $\mathbf{Loc}$ for the set of all locations: as anticipated, while movements may be assumed to be continuous, we identify locations as the countable set of cells that constitute the observing areas within the network. The syntax is shown in Table I.

Networks are collections of nodes, devices running in parallel and using channels to communicate messages. As usual, **0** denotes the empty network and $M_1 \mid M_2$ the parallel composition of two networks. $n[P]_l$ is 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. 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].

Processes are sequential and live within the nodes: **0** is the inactive process; $\mathtt{in}(c, \tilde{x}).P$ is ready to listen to a transmission, while $\mathtt{out}\langle c_{L,r}, \tilde{w}\rangle.P$ is ready to transmit. In $\mathtt{in}(c, \tilde{x}).P$, the variables in $\tilde{x}$ are bound with scope in P. As to the output form, the tag r represents the transmission radius of the sender: the choice of specific transmission ranges may depend on various parameters, and is left to the process running inside the transmitter node. The tag L, in turn, signals the locations from which the transmission will be observed. The remaining syntactic forms are standard: $[w_1 = w_2]P, Q$ behaves as P if $w_1 = w_2$, and as Q otherwise. $A\langle \tilde{w}\rangle$ is the process defined via a (possibly recursive) definition $A(\tilde{x}) \overset{\text{def}}{=} P$, with $|\tilde{x}| = |\tilde{w}|$ where $\tilde{x}$ contains all channels and variables that are free in P.

Two further process forms arise as a result of reduction. In particular, processes that are ready to send or receive evolve into active senders and receivers:

P, Q	::=	...	As in Table I
	$\mid$	$c(\tilde{x}).P$	Active input
	$\mid$	$\bar{c}_{L,r}\langle \tilde{w}\rangle.P$	Active output

Here, $c(\tilde{x}).P$ is actively receiving a tuple $\tilde{w}$ of (closed) values via channel c and continues as $P\{\tilde{w}/\tilde{x}\}$, i.e., as P with $\tilde{w}$ substituted for $\tilde{x}$ (where $|\tilde{x}| = |\tilde{w}|$). Dually, $\bar{c}_{L,r}\langle \tilde{w}\rangle.P$ is transmitting a tuple of values $\tilde{w}$ via channel c and then continues as P. We say that a process P is *active* if it is in prefix form with the prefix denoting an active input or output action. Predicate $\mathtt{Active}(P)$ is $\mathtt{true}$ when P is active, and $\mathtt{A}(M)$ denotes the network composed of all the active nodes in M, i.e., all nodes $n[P]_l$ in M with P active.

In a network of the form $(\nu c)M$, the channel c is said to be bound in M. We denote by $fc(M)$ the set of channels which are not bound in M.

Node connectivity is verified by looking at the physical location and the transmission radius of the sender: a message broadcast by a node is received only by the nodes that lie in the area delimited by the transmission radius of the sender. We presuppose a function $d(\cdot, \cdot)$ which takes two locations and returns the distance separating them (function d can be simply the euclidean distance between two locations, or a more complex function dealing with potential obstacles).

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 $i \in I$. 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, $\sum_{k \in \mathbf{Loc}} \mathbf{J}^n_{lk} = 1$ for all locations $l \in \mathbf{Loc}$. Static nodes are associated with the identity Markov chain, i.e., the identity matrix $\mathbf{J}^n_{ll} = 1$ for all $l \in \mathbf{Loc}$ and $\mathbf{J}^n_{lk} = 0$ for all $l \neq k$. We note by μ^n_l the probability distribution associated with node n located at l, that is, the function over $\mathbf{Loc}$ such that $\mu^n_l(k) = \mathbf{J}^n_{lk}$, for all $k \in \mathbf{Loc}$.

[1]Of course, since channels represent radio frequencies, they 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.

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 $[\![M]\!]_{\mu_l^n}$ the probability distribution over the set of networks induced by μ_l^n and defined as follows: for all network M',

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

Intuitively, $[\![M]\!]_{\mu_l^n}(M')$ 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 $[\![M]\!]_{\mu_l^n}$ if $[\![M]\!]_{\mu_l^n}(M') \neq 0$. 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 $\{\mu_l^n \mid n \text{ is a node and } l \in \mathbf{Loc}\} \cup \{\Delta\}$.

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

The synchronization over a wireless channel is described by the two rules (R-Bgn-Bcast) and (R-End-Bcast). (R-Bgn-Bcast) models the start of a transmission, with node n transiting from ready to active state to transmit message $\tilde{v}$ on channel c with radius r. The state change in n may cause a collision, which the rule captures as follows. We abuse the notation and write $n_h \in H$ to note nodes n_h with $h \in H$, for any index set H). The premise of the rule describes a situation in which nodes $n_k \in K$ and $n_i \in I$ are actively involved in a synchronization, while node n and the $n_j \in J$ are in (output and input, respectively) ready state. Given that all the active transmitters are out of n's range (because $d(l, l_i) > r_i$), n transits into active state: this awakes the $n_j \in J$, as they are now in range of an active transmitter, and at the same time causes a collision at the $n_k \in K$, which also are in range and were already active on input: as a result the $n_k \in K$ exit their active state, receiving the error signal $\perp$. All the remaining active receivers that do not sense a collision, and are in the range of an active sender may conclude the synchronisation (see the R-End-Bcast rule).

As we mentioned earlier, the label L signals the set of locations at which the transmission will be observed. 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.

Rule (R-Move) describes node mobility. 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. We assume that a node can move only if it is not actively involved in

any synchronization: as a result, nodes may move when they are in a ready, but not active state.

All the remaining rules are standard, but a further remark is in order about the (R-Par) rule and its interaction with the rules that govern synchronization. In fact, such interactions may give rise to inconsistent network configurations. To see that, observe that an application of the (R-Par) rule may cause messages to be lost by active receivers located within the range of an active sender, even when there is no interference. Similarly, an application of (R-Par) may exclude any set of active sender and/or receiver from a synchronization: in both cases, the network is left in an inconsistent state, with active senders (dually receivers) and no receiver (sender) in range. While it would be possible to rectify the problem by including conditions to exclude critical pairs for the (R-Par) and synchronization rules, it is technically more convenient to simply disregard any undesired reduction. This is achieved in our framework (to be discussed shortly) by resorting to the notion of "admissible scheduler" to guide the dynamics of networks through "well-formed" executions.

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 [9], an execution for M is a (possibly infinite) sequence of steps $M \rightarrow_{\theta_1} M_1 \rightarrow_{\theta_2} M_2....$. 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 execution $M \rightarrow_{\theta_1} M_1 \dots \rightarrow_{\theta_j} M_j$ of length 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 $e \leq_{prefix} e'$. We write $M \rightarrow^* M'$ if there exists a finite execution $e \in Exec_M$ such that $last(e) = M'$.

We formalize the observational semantics for our calculus in terms of a notion *barb* [8], which is naturally expressed in terms of message transmission.

We denote by $behave(M) = \{[\![M']\!]_\theta \mid M \rightarrow [\![M']\!]_\theta\}$ the set of the possible behaviours of M. In order to solve the nondeterminism in a network execution, we consider each possible probabilistic transition $M \rightarrow [\![M']\!]_\theta$ as arising from a *scheduler* (see [7]). A *scheduler* is a total function F assigning to a finite execution e a distribution $[\![N]\!]_\theta \in behave(last(e))$. We define the set of executions starting from a network M and driven by a scheduler 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 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

$$(\text{R-Bgn-Bcast}) \ \frac{\forall i \in I.d(l,l_i) > r_i \quad \forall i \in I \ \forall j \in J.d(l_i,l_j) > r_i \quad \forall h \in (J \cup K).d(l,l_h) \le r}{n[\text{out}\langle c_{L,r},\tilde{v}\rangle.P]_l \mid M \to [\![n[\bar{c}_{L,r}\langle\tilde{v}\rangle.P]_l \mid M']\!]_\Delta}$$

where $M \equiv \prod_{i \in I} n_i[\bar{c}_{L_i,r_i}\langle\tilde{v}_i\rangle.P_i]_{l_i} \mid \prod_{j \in J} n_j[\text{in}(c,\tilde{x}_j).P_j]_{l_j} \mid \prod_{k \in K} n_k[c(\tilde{x}_k).P_k]_{l_k}$,
$M' \equiv \prod_{i \in I} n_i[\bar{c}_{L_i,r_i}\langle\tilde{v}_i\rangle.P_i]_{l_i} \mid \prod_{j \in J} n_j[c(\tilde{x}_j).P_j]_{l_j} \mid \prod_{k \in K} n_k[P_k\{\bot/\tilde{x}_i\}]_{l_k}$

$$(\text{R-End-Bcast}) \ \frac{\forall j \in J.d(l,l_j) \le r}{n[\bar{c}_{L,r}\langle\tilde{v}\rangle.P]_l \mid \prod_{j \in J} n_j[c(\tilde{x}_j).P_j]_{l_j} \to [\![n[P]_l \mid \prod_{j \in J} n_j[P_j\{\tilde{v}/\tilde{x}_j\}]_{l_j}]\!]_\Delta}$$

$$(\text{R-Res}) \ \frac{M \to [\![M']\!]_\theta}{(\nu c)M \to [\![(\nu c)M']\!]_\theta} \qquad\qquad (\text{R-Move}) \ \frac{\text{Active}(P) = \text{false}}{n[P]_l \to [\![n[P]_l]\!]_{\mu_l^n}}$$

$$(\text{R-Par}) \ \frac{M \to [\![M']\!]_\theta}{M|N \to [\![M'|N]\!]_\theta} \qquad\qquad (\text{R-Struct}) \ \frac{N \equiv M \quad M \to [\![M']\!]_\theta \quad M' \equiv N'}{N \to [\![N']\!]_\theta}$$

Table II: Reduction Semantics

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 note by $Exec_M^F(H)$ the set of executions starting from M and crossing a state in H. Formally $Exec_M^F(H) = \{e \in Exec_M^F \mid last(e^j) \in H \text{ for some } j\}$. We denote the probability for a network M to evolve into a network in H according to the policy given by F as $Prob_M^F(H) = Prob_M^F(Exec_M^F(H))$.

As we anticipated, we restrict the class of all networks (resp. executions) to the class of well-formed networks (resp. executions) where, (1) a transmitter, before transiting in active state checks that, locally, the communication channel is not presently busy with other transmissions, and (2) each active receiver in the network is in the transmission cell of exactly one transmitter. In order to restrict the set of all executions to the set of well-formed executions, we restrict the set of all schedulers to the specific set of *admissible* schedulers. For this purpose, we introduce the following auxiliary operator: $\text{Top}(\cdot)$ over networks. A channel c is at the top level of a network M, denoted $c \in \text{Top}(M)$, if $M \equiv (\nu\tilde{d})(n[P]_l \mid N)$ and P is of the form $\text{in}(c,\tilde{x}).Q$; $c(\tilde{x}).Q$; $\text{out}\langle c_{L,r},\tilde{w}\rangle.Q$; or $\bar{c}_{L,r}\langle\tilde{w}\rangle.Q$.

Definition 2.1: [Well-formed network] A network M is *well-formed* if either $\text{A}(M) \equiv \mathbf{0}$ or $\text{A}(M) \equiv (\nu\tilde{d})\big(\prod_{i \in I} n_i[\bar{c}_{L_i,r_i}\langle\tilde{v}_i\rangle.P_i]_{l_i} \mid \prod_{j \in J} n_j[c(\tilde{x}_j).P_j]_{l_j} \mid \text{A}(N)\big)$ for some N and the following conditions hold:

- $\forall i,i' \in I.d(l_i,l_{i'}) > max(r_i,r_{i'})$,
- $\forall j \in J.\exists!i \in I$ such that $d(l_i,l_j) \le r_i$,
- $c \notin \text{Top}(\text{A}(N))$, and N is well-formed. $\qquad\square$

Definition 2.2: [Admissible scheduler] A scheduler F is *admissible* if for all executions e and for all networks M in the support of $F(e)$, M is well-formed. $\qquad\square$
We denote by $Sched$ the set of admissible schedulers.

We are ready to discuss our notion of observation. We first introduce a notation for *strong barbs*: for any network M,

we write $M\downarrow_{c@K}$ whenever $M \equiv (\nu\tilde{d})(n[\bar{c}_{L,r}\langle\tilde{v}\rangle.P]_l \mid M')$, with $c \notin \tilde{d}$, $K \subseteq L$ and for all k in K, $d(l,k) \le r$. In other words, the strong barb $M\downarrow_{c@K}$ signals that an active transmission from c can be observed in M from some of the intended observation points in L for that transmission. This notion of strong barb generalizes the corresponding notion in related calculi, notably [6]: indeed, taking L to be **Loc** uniformly on all output prefixes, our definition coincides with that in [6].

Definition 2.3: [Probabilistic barb] A well-formed network M has a *barb* with probability p on a channel c at locations in K according to the scheduler F, written $M\Downarrow_p^F c@K$, if $Prob_M^F(H) = p$ with $H = \{M' \mid M \to^* M' \downarrow_{c@K}\}$. $\quad\square$

Now, we introduce a probabilistic observational congruence, in the style of [16], [9], which is defined as the largest relation which satisfies the following properties.

Definition 2.4: Let $\mathcal{R}$ be a relation over networks:

- *Barb preservation*: $\mathcal{R}$ is *barb preserving* if $M\mathcal{R}N$ and $M\Downarrow_p^F c@K$ for some F implies that there exists F' such that $N\Downarrow_p^{F'} c@K$.
- *Reduction closure*: $\mathcal{R}$ is *reduction closed* if $M\mathcal{R}N$ implies that for all F, there exists F' such that for all classes $\mathcal{C} \in \mathcal{N}/\mathcal{R}$, $Prob_M^F(\mathcal{C}) = Prob_N^{F'}(\mathcal{C})$.
- *Contextuality*: $\mathcal{R}$ is *contextual* if $M\mathcal{R}N$ implies that for every context $\mathcal{C}[\cdot]$, it holds that $\mathcal{C}[M] \mathcal{R} \mathcal{C}[N]$, where a context is a network term with a hole $[\cdot]$ defined by the grammar: $\mathcal{C}[\cdot] ::= [\cdot] \mid [\cdot]|M \mid M|[\cdot] \mid (\nu c)[\cdot]$. $\quad\square$

Definition 2.5: [Probabilistic observational congruence] *Probabilistic observational congruence*, written $\cong_p$, is the largest symmetric relation over networks which is reduction closed, barb preserving and contextual. $\quad\square$

III. A BISIMULATION-BASED PROOF TECHNIQUE

We define a LTS semantics for our calculus, which is built upon two sets of rules: one for processes and one for

$$\text{(Beg-Out)} \quad \frac{-}{\mathtt{out}\langle c_{L,r}, \tilde{v}\rangle.P \xrightarrow{\bar{c}_{L,r}} \bar{c}_{L,r}\langle\tilde{v}\rangle.P} \qquad\qquad \text{(End-Out)} \quad \frac{-}{\bar{c}_{L,r}\langle\tilde{v}\rangle.P \xrightarrow{\bar{c}_{L,r}\tilde{v}} P}$$

$$\text{(Beg-In)} \quad \frac{-}{\mathtt{in}(c,\tilde{x}).P \xrightarrow{c} c(\tilde{x}).P} \qquad\qquad \text{(End-In)} \quad \frac{-}{c(\tilde{x}).P \xrightarrow{c\vartheta} P\{\vartheta/\tilde{x}\}}$$

$$\text{(Then)} \quad \frac{P \xrightarrow{\eta} P'}{[v=v]P,Q \xrightarrow{\eta} P'} \qquad \text{(Else)} \quad \frac{Q \xrightarrow{\eta} Q' \quad v_1 \neq v_2}{[v_1=v_2]P,Q \xrightarrow{\eta} Q'} \qquad \text{(Rec)} \quad \frac{P\{\tilde{v}/\tilde{x}\} \xrightarrow{\eta} P' \quad A(\tilde{x}) \overset{\text{def}}{=} P}{A\langle\tilde{v}\rangle \xrightarrow{\eta} P'}$$

Table III: LTS rules for Processes

networks. Table III presents the LTS rules for processes. Transitions are of the form $P \xrightarrow{\eta} P'$, where η ranges over:

$$\eta ::= c \mid c\vartheta \mid \bar{c}_{L,r} \mid \bar{c}_{L,r}\tilde{v} \quad \text{with} \quad \vartheta ::= \tilde{v} \mid \perp.$$

Rules (Beg-Out) and (End-Out) model the beginning and the end of an output action. Rule (Beg-In) models a process beginning listening to a channel in order to receive a value. Rule (End-In) models either the correct reception of a message or the reception of a $\perp$ due to a collision.

Table IV presents the LTS rules for networks. The transitions are of the form $M \xrightarrow{\gamma} [\![M']\!]_\theta$, where M is a network, $[\![M']\!]_\theta$ is a distribution over networks, and γ ranges over:

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

We denote by $\mathtt{A}^s_M(c,l)$ the set of active senders of M on channel c reaching l, i.e., if $\mathtt{A}(M) \equiv (\nu\tilde{d})\big(\prod_{i\in I} n_i[\bar{c}_{L_i,r_i}\langle\tilde{v}_i\rangle.P_i]_{l_i} \mid \prod_{j\in J} n_j[c(\tilde{x}_j).P_j]_{l_j} \mid N\big)$ and $c \notin \mathtt{Top}(N)$ then $\mathtt{A}^s_M(c,l) = \{n_i \mid i \in I, \ d(l,l_i) \leq r_i\}$.

Rules (Beg-Snd) and (End-Snd) model the transmission of a message $\tilde{v}$ through channel c with radius r to the set L of observers. Transmissions are non-atomic actions. (Beg-Rcv) models the beginning of a message reception, while (End-Rcv) models both the successful reception of a message or the reception of a failure message (denoted by $\perp$) due to an interference. Rule (Beg-Bcast) models the beginning of a broadcast message propagation: all the nodes lying within the transmission cell of the sender may begin to receive a message (regardless of the fact that they are in L). Rule (Coll-Bcast) models the collision occurred at the location of a receiver lying within the intersection of the transmission area of different nodes transmitting simultaneously through the same channel. Rule (End-Bcast) models the conclusion of a broadcast message propagation: all the nodes lying within the transmission cell of the sender will successfully receive a message. Rule (Obs) models the observability of a transmission: every transmission may be detected (and hence *observed*) by any recipient located within the transmission cell of one sender and outside the "interference area", that is the intersection of the transmission areas of the active senders of the network. The label $c!\tilde{v}@K \triangleleft R$ represents the transmission of the tuple $\tilde{v}$ of messages via c to the subset K of observers inside the reachable locations R within the transmission cell of the sender. Notice that collisions are not observable and only a correctly ended transmission may be observed. Rules (Lose1) and (Lose2) model 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^n_l, which depends on the Markov chain $\mathbf{J}^n$ statically associated with n. Nodes can move only if they are not executing any active action (i.e., nodes cannot move while transmitting or receiving). Rule (Res) models the standard channel restriction, where $\mathtt{Chan}(\gamma) = c$ if γ is of the form $c?@l$; $c?\vartheta@l$; $c_L![l,r]$; $c_L!\tilde{v}[l,r]$; or $c!\tilde{v}@K \triangleleft R$, and $\mathtt{Chan}(\tau) = \perp$. Rule (Par) is standard.

Based on the LTS semantics, we define a notion of probabilistic labelled bisimilarity which is parametric with respect to a set of schedulers $\mathcal{F}$; then we show that, when $\mathcal{F}$ coincides with the set of all admissible schedulers $Sched$, this is a complete characterisation of our *probabilistic observational congruence*. Let

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

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. A labelled *execution* e of a network M is a finite (or infinite) sequence of steps: $M \xrightarrow{\alpha_1}_{\theta_1} M_1 \xrightarrow{\alpha_2}_{\theta_2} M_2... \xrightarrow{\alpha_k}_{\theta_k} M_k$. With abuse of notation, we define $Exec_M$, $last(e)$, e^j and $e\uparrow$ as for unlabeled executions. We denote by $lbehave(M)$ the set of all possible behaviors 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[2] 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 (admissible) schedulers for the LTS semantics. Given a network M and a scheduler F, we

---

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

$$\text{(Beg-Snd)} \frac{P \xrightarrow{\bar{c}_L, r} P'}{n[P]_l \xrightarrow{c_L![l,r]} [\![n[P']_l]\!]_\Delta} \qquad \text{(End-Snd)} \frac{P \xrightarrow{\bar{c}_L, r\tilde{v}} P'}{n[P]_l \xrightarrow{c_L!\tilde{v}[l,r]} [\![n[P']_l]\!]_\Delta}$$

$$\text{(Beg-Rcv)} \frac{P \xrightarrow{c} P'}{n[P]_l \xrightarrow{c?@l} [\![n[P']_l]\!]_\Delta} \qquad \text{(End-Rcv)} \frac{P \xrightarrow{c\vartheta} P'}{n[P]_l \xrightarrow{c?\vartheta@l} [\![n[P']_l]\!]_\Delta}$$

$$\text{(Beg-Bcast)} \frac{M \xrightarrow{c_L![l,r]} [\![M']\!]_\Delta \quad N \xrightarrow{c?@l'} [\![N']\!]_\Delta \quad d(l,l') \leq r \ \wedge \ \mathtt{A}_N^s(c,l) = \mathtt{A}_N^s(c,l') = \emptyset}{M|N \xrightarrow{c_L![l,r]} [\![M'|N']\!]_\Delta}$$

$$\text{(Coll-Bcast)} \frac{M \xrightarrow{c_L![l,r]} [\![M']\!]_\Delta \quad N \xrightarrow{c?\perp@l'} [\![N']\!]_\Delta \quad d(l,l') \leq r \ \wedge \ \mathtt{A}_N^s(c,l) = \emptyset}{M|N \xrightarrow{c_L![l,r]} [\![M'|N']\!]_\Delta}$$

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

$$\text{(Obs)} \frac{M \xrightarrow{c_L!\tilde{v}[l,r]} [\![M']\!]_\Delta \quad R = \{l' : \ d(l,l') \leq r \ \wedge \ | \ \mathtt{A}_M^s(c,l') | = 1\} \quad K \subseteq R \cap L}{M \xrightarrow{c!\tilde{v}@K \triangleleft R} [\![M']\!]_\Delta}$$

$$\text{(Lose1)} \frac{M \xrightarrow{c_L![l,r]} [\![M']\!]_\Delta}{M \xrightarrow{\tau} [\![M']\!]_\Delta} \qquad \text{(Lose2)} \frac{M \xrightarrow{c_L!\tilde{v}[l,r]} [\![M']\!]_\Delta}{M \xrightarrow{\tau} [\![M']\!]_\Delta}$$

$$\text{(Move)} \frac{\mathtt{Active}(P) = \mathtt{false}}{n[P]_l \xrightarrow{\tau} [\![n[P]_l]\!]_{\mu_l^n}} \quad \text{(Res)} \frac{M \xrightarrow{\gamma} [\![M']\!]_\theta \quad \mathtt{Chan}(\gamma) \neq c}{(\nu c)M \xrightarrow{\gamma} [\![(\nu c)M']\!]_\theta} \quad \text{(Par)} \frac{M \xrightarrow{\gamma} [\![M']\!]_\theta}{M|N \xrightarrow{\gamma} [\![M'|N]\!]_\theta}$$

Table IV: LTS rules for Networks

define $Exec_M^F$ as the set of all labelled executions starting from M and driven by F.

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

Definition 3.1: [Weak action] The definition of weak action is the following:

- $\Longrightarrow$ is the transitive and reflexive closure of $\xrightarrow{\tau}$.
- $\xRightarrow{c?@k}$ denotes $\Longrightarrow \xrightarrow{c?@k} \Longrightarrow$.
- $\xRightarrow{c?\vartheta@k}$ denotes $\Longrightarrow \xrightarrow{c?\vartheta@k} \Longrightarrow$.
- $\xRightarrow{c!\tilde{v}@K\triangleleft R}$ denotes $\Longrightarrow \xrightarrow{c!\tilde{v}@K\triangleleft R} \Longrightarrow$.   □

In the probabilistic setting, while considering a computation with observable content, it is necessary to take into account the actual probability of this computation to ensure that weakly bisimilar systems may not only match one another's transitions but also perform these transitions with matching probabilities. To achieve this, we denote by $Exec_M^F(\xRightarrow{\alpha}, H)$ 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, we define the probability of reaching a network in H from M by performing $\xRightarrow{\alpha}$, according to a scheduler F as $Prob_M^F(\xRightarrow{\alpha}, H) = Prob_M^F(Exec_M^F(\xRightarrow{\alpha}, H))$.

Definition 3.2: [Probabilistic labelled bisimilarity] Let M and N be two networks, and $\mathcal{F}$ be a set of schedulers. A relation $\mathcal{R}^\mathcal{F}$ over networks is a *probabilistic labelled bisimulation relative to $\mathcal{F}$* if $M\mathcal{R}^\mathcal{F}N$ implies: for all scheduler $F \in \mathcal{F}$ there exists a scheduler $F' \in \mathcal{F}$ such that for all α and for all classes $\mathcal{C}$ in $\mathcal{N}/\mathcal{R}^\mathcal{F}$ it holds:

- if $\alpha = \tau$ or $\alpha = c!\tilde{v}@K \triangleleft R$ then $Prob_M^F(\xrightarrow{\alpha}, \mathcal{C}) = Prob_N^{F'}(\Longrightarrow \mathcal{C})$;
- if $\alpha = c?@l$ or $\alpha = c?\tilde{v}@l$ then either $Prob_N^F(\xrightarrow{\alpha}, \mathcal{C}) = Prob_N^{F'}(\xRightarrow{\alpha}, \mathcal{C})$ or $Prob_M^F(\xrightarrow{\alpha}, \mathcal{C}) = Prob_N^{F'}(\Longrightarrow, \mathcal{C})$.

Probabilistic labelled bisimilarity relative to $\mathcal{F}$, $\approx_p^\mathcal{F}$, is the largest symmetric probabilistic labelled bisimulation relative to $\mathcal{F}$ over networks. We denote by $\approx_p$ the relation $\approx_p^{Sched}$ where $Sched$ is the set of all admissible schedulers and simply call it *probabilistic labelled bisimilarity*.   □

Probabilistic labelled bisimilarity is a complete characterisation of probabilistic barbed congruence.

Theorem 3.3: Let M and N be well-formed networks. $M \approx_p N$ if and only if $M \cong_p N$.

IV. INTERFERENCE METRICS

We define a preorder over networks which allows us to compare the average level of interferences of different networks but exhibiting the same connectivity behaviour relative to a specific set of schedulers $\mathcal{F}$. We consider two interference metrics. The first one focuses on the senders and counts

how many currently broadcasting nodes might interfere with each other due to the overlapping communication ranges. The second metric puts the emphasis on the receiver nodes and counts the number of active receivers which are simultaneously reached by two (or more) transmissions.

Sender-based interference: Let M be a network. Given a channel c, we denote by $\mathbf{Overlap^s}(M, c)$ the set of nodes currently broadcasting over c and whose transmission areas are overlapping at some locations. Formally, let $\mathtt{A}(M) \equiv (\nu \tilde{d})\big(\prod_{i \in I} n_i[\bar{c}_{L_i, r_i}\langle \tilde{v}\rangle.P_i]_{l_i} \mid \prod_{j \in J} n_j[c(\tilde{x}_j).P_j]_{l_j} \mid M'\big)$ be the active nodes of M, where $c \notin \mathtt{Top}(M')$, then

$$\mathbf{Overlap^s}(M, c) = \{n_i \mid i \in I, \exists i' \in I.i \neq i' \text{ and } d(l_i, l_{i'}) \leq r_i + r_{i'}\}.$$

For example, consider the following network

$$\begin{aligned} \hat{M} \quad = \quad & n_1[\mathtt{out}\langle c_{L_1, r_1}, \tilde{v}_1\rangle.P_1]_{l_1} \mid n_2[\bar{c}_{L_2, r_2}\langle \tilde{v}_2\rangle.P_2]_{l_2} \\ & \mid n_3[\bar{c}_{L_3, r_3}\langle \tilde{v}_3\rangle.P_3]_{l_3} \mid n_4[\bar{d}_{L, r}\langle \tilde{v}\rangle.P_4]_{l_4} \\ & \mid n_5[c(\tilde{x}).P_5]_{l_5} \mid n_6[\mathtt{in}(c, \tilde{y}).P_6]_{l_6} \end{aligned}$$

where $d(l_i, l_{i'}) > r_i$ for all $i, i' \in \{1, 2, 3\}$, i.e., the nodes n_1, n_2, and n_3 are all far enough away from each other and can broadcast at the same time over the channel c. In this case, function $\mathbf{Overlap^s}(\hat{M}, c)$ is defined as follows: for all $c' \neq c$ (e.g., $c' = d$) $\mathbf{Overlap^s}(\hat{M}, c') = \emptyset$, while

$$\mathbf{Overlap^s}(\hat{M}, c) = \begin{cases} \{n_2, n_3\} & \text{if } d(l_2, l_3) \leq r_2 + r_3 \\ \emptyset & \text{otherwise.} \end{cases}$$

We define the sender-based level of interference induced by a probabilistic action as follows:

$$\mathbf{Interf^s}(M, N) = \begin{cases} |\mathbf{Overlap^s}(N, c)| - |\mathbf{Overlap^s}(M, c)| \\ \quad \text{if } M \xrightarrow{c_L![l, r]} [\![N]\!]_\Delta \text{ for some } L, l, r; \\ 0 \quad \text{otherwise.} \end{cases}$$

Consider again the above network $\hat{M}$. Since $d(l_1, l_i) > r_1$ for $i \in \{2, 3\}$, we have $\hat{M} \xrightarrow{c_{L_1}![l_1, r_1]} [\![\hat{N}]\!]_\Delta$, where

$$\begin{aligned} \hat{N} \quad = \quad & n_1[\bar{c}_{L_1, r_1}\langle \tilde{v}_1\rangle.P_1]_{l_1} \mid n_2[\bar{c}_{L_2, r_2}\langle \tilde{v}_2\rangle.P_2]_{l_2} \\ & \mid n_3[\bar{c}_{L_3, r_3}\langle \tilde{v}_3\rangle.P_3]_{l_3} \mid n_4[\bar{d}_{L, r}\langle \tilde{v}\rangle.P_4]_{l_4} \\ & \mid n_5[P_5']_{l_5} \mid n_6[P_6']_{l_6} \end{aligned}$$

The sender-based level of interference induced by $\hat{M} \xrightarrow{c_{L_1}![l_1, r_1]} [\![\hat{N}]\!]_\Delta$ is, e.g.:

- If n_1 is too far away from both n_2 and n_3, i.e., $d(l_1, l_j) > r_1 + r_j$ for $j \in \{2, 3\}$, then $\mathbf{Overlap^s}(\hat{N}, c) = \mathbf{Overlap^s}(\hat{M}, c)$. Hence, $\mathbf{Interf^s}(\hat{M}, \hat{N}) = 0$.
- If n_2 and n_3 were already overlapping, i.e., $d(l_2, l_3) \leq r_2 + r_3$ and n_1 is not too far away from at least one of them, i.e., $d(l_1, l_2) \leq r_1 + r_2$ or $d(l_1, l_3) \leq r_1 + r_3$ then $\mathbf{Overlap^s}(\hat{N}, c) = \{n_1, n_2, n_3\}$. Therefore, $\mathbf{Interf^s}(\hat{M}, \hat{N}) = 1$.

Receiver-based interference: Hereafter, we denote by $\mathbf{Coll^r}(M, c, l, r)$ the set of nodes in M which are currently listening over channel c and lie in the transmission range of a sender located at l with radius r. Formally, let $\mathtt{A}(M) \equiv (\nu \tilde{d})\big(\prod_{i \in I} n_i[\bar{c}_{L_i, r_i}\langle \tilde{v}\rangle.P_i]_{l_i} \mid \prod_{j \in J} n_j[c(\tilde{x}_j).P_j]_{l_j} \mid M'\big)$ be the active nodes of M, where $c \notin \mathtt{Top}(M')$, then

$$\mathbf{Coll^r}(M, c, l, r) = \{n_j \mid j \in J \text{ and } d(l, l_j) \leq r\}.$$

The number of receiver-based interferences induced by a probabilistic step is:

$$\mathbf{Interf^r}(M, N) = \begin{cases} |\mathbf{Coll^r}(M, c, l, r)| \\ \quad \text{if } M \xrightarrow{c_L![l, r]} [\![N]\!]_\Delta \text{ for some } L; \\ 0 \quad \text{otherwise.} \end{cases}$$

For instance, if we consider again our previous networks $\hat{M}$ and $\hat{N}$, assuming that n_1 can reach both l_5 and l_6 then $P_5' = P_5\{\perp/\tilde{x}\}$ and $P_6' = c(\tilde{y}).P_6$. Then, $\mathbf{Coll^r}(\hat{M}, c, l_1, r_1) = \{n_5\}$. Hence $\mathbf{Interf^r}(\hat{M}, \hat{N}) = 1$.

Now, let $\chi \in \{\mathbf{s}, \mathbf{r}\}$. The χ-type number of interferences induced by an execution $e = M_0 \xrightarrow{\alpha_1}_{\theta_1} M_1 \dots \xrightarrow{\alpha_k}_{\theta_k} M_k$ is

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

Let H be a set of networks, we denote by $Paths_M^F(H)$ the set of all executions from M ending in H and driven by F which are not prefixes of any other execution ending in H. Formally, $Paths_M^F(H) = \{e \in Exec_M^F(H) \mid last(e) \in H \text{ and } \forall e' \text{ such that } e <_{prefix} e', e' \notin Paths_M^F(H)\}$. The average number of interferences is computed by weighting the number of interferences of each execution by its probability according to F and normalized by the overall probability of reaching H.

Definition 4.1: Let H be a set of networks. The average number of interferences to reach H from M according to the scheduler F is

$$\mathbf{Interf}_{M, F}^\chi(H) = \frac{\sum_{e \in Paths_M^F(H)} \mathbf{Interf^\chi}(e) \times P_M^F(e)}{\sum_{e \in Paths_M^F(H)} P_M^F(e)}$$

$\square$

Definition 4.2: Let $\mathcal{H}$ be a countable set of sets of networks and $\mathcal{F}$ a set of schedulers. We say that N is *at least as interference efficient as* M relative to $\mathcal{H}$ and $\mathcal{F}$, denoted

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

if $N \approx_p^{\mathcal{F}} M$ and, for all $H \in \mathcal{H}$ and for all schedulers $F \in \mathcal{F}$, there exists a scheduler $F' \in \mathcal{F}$ such that $\mathbf{Interf}_{N, F'}^\chi(H) \leq \mathbf{Interf}_{M, F}^\chi(H)$. $\square$

V. A CASE STUDY: THE ALTERNATING BIT PROTOCOL

The alternating bit protocol (ABP) is a simple network protocol for reliable transmissions. Messages sent from a transmitter to a receiver include the payload and a 1-bit sequence number. The sender resends the message until it

$SND_j\langle b_j, T_j\rangle$	$=$	$[\texttt{empty}(T) = \texttt{false}](\texttt{out}\langle c_{\{k\},r_j}, (b_j, \texttt{head}(T_j), n_j)\rangle.WAIT\_Ack_j\langle b_j, T_j\rangle), \texttt{out}\langle \texttt{ok}_{\{k\},r_j}, (n_j, END)\rangle$
$WAIT\_Ack_j\langle b_j, T_j\rangle$	$=$	$\texttt{in}(c,(x,y,z)).[y = n_j]([(x = b_j) \wedge (z = ACK)]SND_j\langle \neg b_j, \texttt{dequeue}(T_j)\rangle, SND_j\langle b_j, T_j\rangle),$
		$WAIT\_Ack_j\langle b_j, T_j\rangle$
$RCV\langle b_1, b_2\rangle$	$=$	$\texttt{in}(c,(x,y,z)).[z = n_1](([x = b_1]\texttt{out}\langle c_{\{l_1,l_2,l_3,l_4\},r}, (b_1, n_1, ACK)\rangle.RCV\langle \neg b_1, b_2\rangle,$
		$\texttt{out}\langle c_{\{l_1,l_2,l_3,l_4\},r}, (b_1, n_1, NACK)\rangle.RCV\langle b_1, b_2\rangle),$
		$[z = n_2]([x = b_2]\texttt{out}\langle c_{\{l_1,l_2,l_3,l_4\},r}, (b_2, n_2, ACK)\rangle.RCV\langle b_1, \neg b_2\rangle,$
		$\texttt{out}\langle c_{\{l_1,l_2,l_3,l_4\},r}, (b_2, n_2, NACK)\rangle.RCV\langle b_1, b_2\rangle)),$
		$\texttt{out}\langle c_{\{l_1,l_2,l_3,l_4\},r}, (b_1, n_1, NACK)\rangle.\texttt{out}\langle c_{\{l_1,l_2,l_3,l_4\},r}, (b_2, n_2, NACK)\rangle.RCV\langle b_1, b_2\rangle$
ABP	$=$	$(\nu c)\big(n_1[SND_1\langle 1, T_1\rangle]_{l_1} \mid n_2[SND_2\langle 1, T_2\rangle]_{l_3}\ m[RCV\langle 1, 1\rangle]_k\big)$

Table V: ABP

receives an `ack` from the receiver including the same bit; when that happens, the sender flips the sequence number and starts a new transmission.

We consider a network consisting of two mobile sender nodes, n_1 and n_2, communicating with a static receiver node m. Node n_1 moves back and forth between locations l_1 and l_2 according to the probability distribution defined by the following discrete time Markov chain:

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

Node n_2 moves similarly between l_3 and l_4 according to a discrete time Markov chain with the same transition matrix J. We also assume that the receiver node is always in the transmission range of both senders (and that the senders are always in the range of the receiver) regardless of where the senders are located. This guarantees that m will receive any packet from the senders (unless there is a collision), and that both senders receive any `ack` sent from m.

We assume that the transmission ranges of the senders overlap only when n_1 is at l_1 and n_2 is at l_3. As a result, unless n_1 is at l_1 and n_2 is at l_3, the senders are in the condition to attempt a simultaneous transmission (as they don't sense each other) leading to an interference: in literature, this is known as the *hidden station problem*.

We may describe the sender and receiver processes of the protocol by means of the recursive definitions reported in Table V. SND_j runs inside node n_j, sending a queue of messages T_j with sequence bit b_j; RCV, in turn, runs inside the receiver node m, expecting messages with sequence bits b_1 and b_2 from n_1 and n_2, respectively. We presuppose few auxiliary functions: `empty()`, `dequeue()` and `head()` implement the standard queue operations, while $\neg b$ flips the value of the bit b. Finally, `ok` is a channel name and a location introduced for the purposes of our analysis.

Interference cancellation scheme for CDMA (Code Division Multiple Access): Here, we sketch a simplified successive interference cancellation (SIC) method for CDMA/CA [2] transmission scheme. Assume that nodes n_1 and n_2 cause an interference at m by sending packets

encoded by signals x_A and x_B. Node m receives the signal $y_1 = x_A + x_B$, detects the interference and stores y_1 in memory. In the successive slot, n_1 successfully resends x_A, i.e., m receives $y_2 = x_A$ and sends an `ack` to n_1. Now, x_B may be extracted from y_1 by m without further retransmissions as the result of $y_1 - x_A$. Although in practice this procedure is not always successful, we assume that messages can always be recovered correctly.

In modelling this protocol, the sender processes remain the same as in the simple ABP protocol defined in Table V, while the receiver process is defined as in Table VI.

In order to compare the observational behaviours of the protocols, we assume that a successful end of transmission of the packets by a sender, indicated by broadcasting the message "END" over the channel `ok`, is observable for any observer node located at k. Thus the protocols are equivalent if for each scheduler F driving one of the protocols, there exists a scheduler F' driving the other one such that both protocols correctly transmit both sequences of messages with the same probabilities. In this analysis, we are only interested in the levels of interference due to the internal nodes of the protocols. Therefore, we restrict communications over the channel c to the internal nodes of the protocols.

The following result states that both protocols exhibit the same observable behaviour.

Proposition 5.1: $ABP \approx_p SIC\_ABP$.

Measuring the interference level of the protocols: Schedulers constitute an essential feature for modeling communication protocols as they provide freedom in modeling implementation and incomplete knowledge of the system. However, many schedulers could be in fact unrealistic or useless. Indeed, schedulers giving priority to communications over movements will, for instance, cancel the two-state nature of the sender nodes, while those giving priority to end broadcasting actions over begin broadcasting actions will prevent any interference. Therefore, we consider the following set $\mathcal{F}_{\mathbf{fas}}$ of *fair alternating schedulers* which:

1) always alternate between sending packets and node movements so that at each interaction of the transmitters with the receiver, the formers could be far enough

$RCV_{SIC}\langle b_1, b_2\rangle$	$= \mathtt{in}(c, (x_1, x_2, x_3))[x_3 = n_1]\big(([x_1 = b_1]\mathtt{out}\langle c_{\{l_1,l_2,l_3,l_4\},r}, (b_1, n_1, ACK)\rangle.RCV_{SIC}\langle \neg b_1, b_2\rangle,$
	$\mathtt{out}\langle c_{\{l_1,l_2,l_3,l_4\},r}, (b_1, n_1, NACK)\rangle.RCV_{SIC}\langle b_1, b_2\rangle),$
	$[x_3 = n_2]([x_1 = b_2]\mathtt{out}\langle c_{\{l_1,l_2,l_3,l_4\},r}, (b_2, n_2, ACK)\rangle.RCV_{SIC}\langle b_1, \neg b_2\rangle,$
	$\mathtt{out}\langle c_{\{l_1,l_2,l_3,l_4\},r}, (b_2, n_2, NACK)\rangle.RCV_{SIC}\langle b_1, b_2\rangle)),$
	$\mathtt{out}\langle c_{\{l_1,l_2,l_3,l_4\},r}, (b_1, n_1, NACK)\rangle.WAIT\langle \perp_{x_1,x_2,x_3}, b_1, b_2\rangle$
$WAIT\langle \perp_{p_1,p_2,p_3}, b_1, b_2\rangle$	$= \mathtt{in}(c, (x_1, x_2, x_3))[x_3 = n_1]([x_1 = b_1](\mathtt{out}\langle c_{\{l_1,l_2,l_3,l_4\},r}, (b_1, n_1, ACK)\rangle.$
	$[f(x_3, p_3) = n_2][b_2 = f(x_1, p_1)](\mathtt{out}\langle c_{\{l_1,l_2,l_3,l_4\},r}, (b_2, n_2, ACK)\rangle.RCV_{SIC}\langle \neg b_1, \neg b_2\rangle),$
	$\mathtt{out}\langle c_{\{l_1,l_2,l_3,l_4\},r}, (b_2, n_2, NACK)\rangle.RCV_{SIC}\langle \neg b_1, b_2\rangle),$
	$\mathtt{out}\langle c_{\{l_1,l_2,l_3,l_4\},r}, (x_1, n_1, NACK)\rangle.WAIT\langle \perp_{x_1,x_2,x_3}, b_1, b_2\rangle$
SIC_ABP	$= (\nu c)\big(n_1[SND_1\langle 1, T_1\rangle]_{l_1} \mid n_2[SND_2\langle 1, T_2\rangle]_{l_3} \mid m[RCV_{SIC}\langle 1, 1\rangle]_k\big)$

Table VI: SIC_ABP

away from each other to cause interference or not;

2) give priority to acknowledgment actions (ACK and $NACK$) to model our assumption of an error-free feedback channel;

3) give priority to begin broadcasting actions (Beg-Bcast) over end broadcasting actions (End-Bcast).

The two protocols exhibit the same observable behaviour relative to $\mathcal{F}_{\mathbf{fas}}$.

Proposition 5.2: $ABP \approx_p^{\mathcal{F}_{\mathbf{fas}}} SIC\_ABP$.

Now let T_1 and T_2 be the queues of messages to be transmitted by the senders. We compare the interference efficiency of the protocols in the context of the set $\mathcal{H}(T_1, T_2) = \{H_\rho(T_1, T_2) \,|\, \rho \leq \max(|T_1|, |T_2|)\}$ where $H_\rho(T_1, T_2)$ means that all the packets up to ρ have been correctly transmitted by both senders and is defined as $H_\rho(T_1, T_2) = H_\rho^1(T_1, T_2) \cup H_\rho^2(T_1, T_2)$ where

$$H_\rho^1(T_1, T_2) = \{M \,|\, M \equiv (\nu c)\big(n_1[SND_1\langle b_1, \mathtt{dequeue}^\rho(T_1)\rangle]_{l'}$$
$$\mid n_2[SND_2\langle b_2, \mathtt{dequeue}^\rho(T_2)\rangle]_{k'} \mid m[RCV\langle b_1, b_2\rangle]_k) \}$$

with the assumption that $\mathtt{dequeue}(\emptyset) = \emptyset$, and $b_1, b_2 \in \{0, 1\}$. Similarly

$$H_\rho^2(T_1, T_2) = \{N \,|\, N \equiv (\nu c)\big(n_1[SND_1\langle b_1, \mathtt{dequeue}^\rho(T_2)\rangle]_{l''}$$
$$\mid n_2[SND_2\langle b_2, \mathtt{dequeue}^\rho(T_2)\rangle]_{k''} \mid m[RCV_{SIC}\langle b_1, b_2\rangle]_k)\}$$

with b_1 and b_2 in $\{0, 1\}$, l', l'' in $\{l_1, l_2\}$, and k', k'' in $\{l_3, l_4\}$. Then, we compute the interference level of the protocols assuming that we start by a move action for each sender node so that their first transmissions could create an interference if they move too far away from each other[3]. The results are summarized in the following propositions.

Proposition 5.3: For all F in $\mathcal{F}_{\mathbf{fas}}$ and for all $\rho \leq \max(|T_1|, |T_2|)$ we have:

$$\mathbf{Interf}_{ABP,F}^{\mathbf{s}}(H_\rho(T_1, T_2)) =$$
$$2 \times \mathbf{Interf}_{ABP,F}^{\mathbf{r}}(H_\rho(T_1, T_2)) =$$
$$2 \times \left(\frac{(p+q)^2}{q^2} - 1\right) \times \min(\rho, |T_1|, |T_2|)$$

with $0 < p, q < 1$. $\qquad \square$

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

The proof relies on the observation that correct packets are sent only when the mobile nodes are in the locations l_1 and l_3. Hence, by exploiting the independence between the stochastic processes underlying the node movements, the result follows by standard analysis of absorbing Markov chains.

Note that our sender-based interference metric coincides with the number of lost packets. For the ABP with SIC, we have:

Proposition 5.4: For all F in $\mathcal{F}_{\mathbf{fas}}$ and each $\rho \leq \max(|T_1|, |T_2|)$ we have:

$$\mathbf{Interf}_{SIC\_ABP,F}^{\mathbf{s}}(H_\rho(T_1, T_2)) =$$
$$2 \times \mathbf{Interf}_{SIC\_ABP,F}^{\mathbf{r}}(H_\rho(T_1, T_2)) =$$
$$2 \times \frac{p}{(p+q)^3}\Big(n(p+q)(p+2q) -$$
$$\frac{((1-p-q)^n - 1)(p+q-1)(p^2 - p(1-p-q)^{n+1} - 4q + 3pq + 2q^2 - p)}{p+q-2}\Big) \times$$
$$\min(\rho, |T_1|, |T_2|)$$

with $0 < p, q < 1$.

Indeed, the n-th steps transition probability matrix J^n is:

$$J^n = \begin{vmatrix} \frac{p(1-p-q)^n + q}{p+q} & \frac{p - p(1-p-q)^n}{p+q} \\ \frac{q - q(1-p-q)^n}{p+q} & \frac{p + q(1-p-q)^n}{p+q} \end{vmatrix}$$

According to the SIC specification, nodes need only to send one packet for a successful packet transmission if they are in the locations l_1 and l_3. All the other location combinations require one of the nodes to send two packets for each successful transmission (while the other sends just one). Starting from states l_1 and l_3, the probability of being still in the same state after $i > 0$ steps is given by $(p(1-p-q)^i + q)^2 / (p+q)^2$ (by independence). We derive the expression given by Proposition 5.4 as the closed expression of the following sum which represents the expected number of observed interferences for sending n packets:

$$\sum_{i=1}^{n} \left(1 - \left(\frac{p(1-p-q)^i + q}{p+q}\right)^2\right).$$

Finally, from Propositions 5.1, 5.3, and 5.4, we can conclude that the SIC-based ABP protocol is much more

interference efficient than its simple version.

Theorem 5.5: $SIC\_ABP \sqsubseteq^{\chi}_{\langle \mathcal{F}_{\mathbf{fas}}, \mathcal{H}(T_1,T_2) \rangle} ABP.$

VI. Related Work and Conclusion

Probabilistic and stochastic models are nowadays widely used in the design and verification of complex systems. In [17] Song and Godskesen propose a probabilistic broadcast calculus for mobile and wireless networks whose connections are unreliable. Palamidessi et al. in [9] define an extension of the applied pi-calculus with nondeterministic and probabilistic choice operators. Priami [10] introduces a stochastic extension of the pi-calculus. In the context of performance evaluation, Hillston et al. [11] introduce the process algebra PEPA which is used for modelling systems composed of concurrently active components which co-operate and share work. Bernardo et al. introduce the Markovian process algebra EMPA$_{gr}$ [12]. All those calculi are built upon atomic actions and do not allow multiple devices to transmit at the same time. The problem of interference is considered by Sangiorgi et al. [5] who propose a calculus to detect collisions due to the simultaneous transmissions of two or more devices. However, their work does not support node mobility and no interference metric is proposed. Finally, existing frameworks based on Petri Nets and queueing networks fall short of accounting for node mobility while maintaining a good accuracy in specifying the protocol designs [13], [14].

To our knowledge, the present paper is the first to provide a comprehensive framework for the behavioral analysis and a quantitative assessment of interference for wireless networks in the presence of node mobility.

Acknowledgments

References

[1] X. Wang and K. Kar, "Throughput modelling and fairness issues in CSMA/CA based ad-hoc networks," in *Proc. of the 24th Annual Joint Conf. of the IEEE Computer and Communications Societies (INFOCOM'05)*. IEEE, 2005, pp. 23–34.

[2] A. Muqattash and M. Krunz, "CDMA-based MAC protocol for wireless ad hoc networks," in *Proc. of the 4th ACM International Symposium on Mobile ad hoc Networking & Computing (MobiHoc'03)*. ACM, 2003, pp. 153–164.

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

[4] L. Gallina and S. Rossi, "Sender- and receiver-centered interference in wireless ad hoc networks," in *Proc. of IFIP Wireless Days 2010 (WD'10)*. IEEE, 2010.

[5] I. Lanese and D. Sangiorgi, "An operational semantics for a calculus for wireless systems," *Theoretical Computer Science (TCS'10)*, vol. 411, no. 19, pp. 1928 – 1948, 2010.

[6] M. Merro, "An observational theory for mobile ad hoc networks," *Information and Computation*, vol. 207, no. 2, pp. 194–208, 2009.

[7] R. Segala and N. Lynch, "Probabilistic simulations for probabilistic processes," in *Proc. of the 5th International Conference on Concurrency Theory (CONCUR'94)*, ser. LNCS, vol. 836. Springer-Verlag, 1994, pp. 481–496.

[8] R. Milner and D. Sangiorgi, "Barbed bisimulation," in *Proc. of International Colloquium on Automata, Languages and Programming (ICALP'92)*, ser. LNCS, vol. 623. Springer-Verlag, 1992, pp. 685–695.

[9] J. Goubault-Larrecq, C. Palamidessi, and A. Troina, "A probabilistic applied pi-calculus," in *Proc. of the 5th Asian Symposium on Programming Languages and Systems (APLAS'07)*, ser. LNCS, vol. 4807. Springer-Verlag, 2007, pp. 175–190.

[10] C. Priami, "Stochastic π-calculus," *The Computer Journal*, vol. 38, no. 7, pp. 578–589, 1995.

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

[12] M. Bernardo and M. Bravetti, "Performance measure sensitive congruences for markovian process algebras," *Theoretical Computer Science*, vol. 290, no. 1, pp. 117 – 160, 2003.

[13] G. Mohimani, F. Ashtiani, A. Javanmard, and M. Hamdi, "Mobility Modeling, Spatial Traffic Distribution, and Probability of Connectivity for Sparse and Dense Vehicular Ad Hoc Networks," *IEEE Trans. on Vehicular Technology*, vol. 58, no. 4, May, 2009.

[14] M. Beccuti, M. D. Pierro, A. Horvàth, A. Horvàth, and K. Farkas, "A Mean Field Based Methodology for Modeling Mobility in Ad Hoc Networks," in *Proc. of 73rd IEEE Vehicular Technology Conference (VTC Spring)*. Budapest, HU: IEEE, 2011, pp. 1–5.

[15] M. Bugliesi, L. Gallina, S. Hamadou, A. Marin, and S. Rossi, "Interference-sensitive preorders for manets," Department of Computer Science, University Ca' Foscari of Venice, Research Report DAIS-2011-10, 2011.

[16] E. Bandini and R. Segala, "Axiomatizations for probabilistic bisimulation," in *Proc. of the 28th International Colloquium on Automata, Languages and Programming (ICALP'01)*, ser. LNCS, vol. 2076. Springer-Verlag, 2001, pp. 370–381.

[17] L. Song and J. Godskesen, "Probabilistic mobility models for mobile and wireless networks," in *Theoretical Computer Science*, ser. IFIP Advances in Information and Communication Technology. Springer Boston, 2010, vol. 323, pp. 86–100.