Evaluating Resistance to Jamming and Casual Interception in Mobile Wireless Networks*

Lucia Gallina, Gian-Luca Dei Rossi, Andrea Marin, Sabina Rossi
DAIS, Università Ca' Foscari Venezia, Italy
{lgallina,deirossi,marin,srossi}@dais.unive.it

ABSTRACT

Mobile ad-hoc and sensor networks play an important role in several application fields. The usage of wireless links and the node mobility make the networks prone to security attacks; among these, jamming attacks are insidious and they consist of one or more nodes continuously transmitting dummy packets to keep some wireless links busy. The goal is to destroy the network connectivity or highly reduce its throughput. In this paper we propose a probabilistic formal method, based on a process algebraic approach, targeted at the analysis of connectivity and the evaluation of interference in mobile networks. We show our framework at work on the analysis of an indoor wireless communication scenario. We evaluate the vulnerability of a network to jamming and casual interception with two different routing strategies for the Hybrid Wireless Mesh Protocol (HWMP).

Categories and Subject Descriptors

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

General Terms

Theory, Reliability

Keywords

MANETs, Jamming, Process Algebra, Routing

1. INTRODUCTION

Mobile ad-hoc networks (MANETs) consist of a set of wireless communicating devices that independently move in a given space. The mobility of the devices is one of the salient features of MANETs because it implies the impossibility of applying well-known protocols for networks based on static infrastructures. In practice, nodes that form a MANET may be laptops, smart-phones but even simpler devices such as sensors. Due to their characteristics, MANETs are the best solution for various applications, ranging from the monitoring of herds of animals to supporting communications in military battlefields and civilian disaster recovery scenarios. The design of MANETs must face a set of research challenges regarding functional and non-functional aspects. Among the functional requirements we mention the *connectivity* that requires the development of reliable protocols on a dynamic infrastructure and unreliable wireless links. Networks' connectivity is a key-factor also when analysing MANETs' protocol robustness to *jamming* attacks. These are interferences generated by malicious users that constantly occupy some frequencies with dummy transmissions in the area surrounding their locations. Their goal is to destroy the connectivity of the MANET or highly reduce its throughput. Although the only possible counter-measure to a jamming attack is the physical removal of the malicious nodes from the network, different protocols may exhibit different robustness. Non-functional requirements are usually expressed in terms of indices for measuring the Quality of Service (QoS), e.g., the system's throughput, latency and overall energy consumption. This latter aspect is particularly relevant since mobile devices rely on a limited battery and several research efforts have been devoted to compare protocols in terms of energy efficiency (see, e.g., [13]). In this paper we resort to the formal calculus introduced in [10] that allows for the formal specification of MANETs in terms of modular composition of its nodes. Routing and communication protocols can be encoded and both functional and non-functional properties can be analysed in a completely algorithmic way. More technically, our calculus is a probabilistic and non-deterministic process algebra whose semantics is inspired by Segala's automata [14]. In the network definition, nodes' mobility is modelled probabilistically; the region where the MANET works is discretised into arbitrary small parts, and the mobility behaviour of the nodes is modelled by a Discrete Time Markov Chain (DTMC). The state of wireless links and the behaviour of the protocols are non-deterministic. The analysis of this model allows one to derive several functional properties of the MANETs, e.g., the observation equivalence of two protocols for a reliable communication over an unreliable channel can be proved [4]. In order to obtain quantitative indices of the network (e.g., throughput, response time, utilisation, energy consumption) the model must be transformed into a purely probabilistic one. This is done by means of schedulers in a similar fashion to [5]. The main

*Work partially supported by the MIUR Project IPODS "Interacting Processes in Open-ended Distributed Systems".

contribution of this paper is the development of a methodology for the analysis of the resilience of MANETs' routing protocols to jamming attack. The calculus proposed in [10] is simplified in order to focus only on the features needed for the connectivity analysis. Roughly speaking, we say that a routing protocol that governs a network N is resilient to the jamming attacks if the functional behaviour of N without the jammers is equivalent - according to some observation semantics - to the functional behaviour of N when the jammers are introduced. It is worth of notice that the peculiarity of this approach is the analysis of a protocol in a scenario, i.e., the outcomes of the connectivity analysis depend on the protocol definition *and* the mobility properties of the network's nodes. For instance, one may decide to analyse the connectivity of a network in which highly mobile nodes have the roles of routing packets that could be destroyed by jammers. The mobility of these nodes could ensure the resilience up to a given number of jammers. We also apply our methodology to study the resilience of two MANETs' routing protocols to jamming attacks. In particular, we will study if under different scenarios and spatial distribution of the jammers, the protocols are able to ensure the network connectivity among the nodes that are not directly disturbed by the attackers. We consider two different routing strategies for the Hybrid Wireless Mesh Protocol (HWMP), the first based on the reactive Ad-hoc On-Demand Distance Vector (AODV) protocol and the second using a proactive tree-based approach. This case study aims at showing the methodology that we propose at work in a relatively simple case. It is worth of notice the fact that our calculus can be encoded into the probabilistic calculi used by automatic verification tools, such as the PRISM model checker [8]. As a consequence, with the implementation of our verifying method, more complicated scenarios than the one considered here could be studied.

Related Works. Probabilistic and stochastic models are nowadays widely used in the design and verification of complex systems. Palamidessi et al. in [5] define an extension of the applied pi-calculus with non-deterministic and probabilistic choice operators. In the context of performance evaluation, Hillston [7] introduces the Performance Evaluation Process Algebra (PEPA) that is used for modelling systems consisting of concurrently active components which co-operate and share work. Bernardo et al. introduce EMPA$_{gr}$ [1], an extended Markovian process algebra including probabilities, priority and exponentially distributed durations. All these calculi abstract out the interferences, since they consider only atomic actions, and they do not allow multiple devices to transmit at the same time. The problem of interference is considered by Sangiorgi et al. [9] who propose a calculus to detect collisions due to the simultaneous transmissions of two or more devices. Differently from our work, their calculus does not support mobility of nodes. In the literature jamming attacks has been clustered into different classes, starting from *proactive jammers*, which continually emits radio signals maintaining the channel constantly busy, to the *reactive jammers*, which remain listening to the channel and then start to interfere only when they sense activity on the channel. In [16] the authors give a detailed classification, as well as a way of distinguishing simple interferences, due to the networks collisions during the communication, and the jamming caused by malicious attackers. In [15] the authors discuss the effect of reactive jammers and show that they are extremely difficult to distinguish from the simple network congestions, and most of the measures used for proactive jamming are not able to detect this kind of attacks.

Plan of the paper. Section 2 introduces our calculus. Section 3 defines the methodology based on the calculus aimed at evaluating the resilience of a MANET to a jamming attack. Section 4 introduces a case study. A network is defined and the routing protocols AODV and tree-based are encoded. Section 5 details the proof technique that underlines the decision of the network's resiliency to a specific jamming attack and in Section 6 we apply it to the case study. Finally, Section 7 concludes the paper.

2. THE CALCULUS

The framework we propose to reason about jammings and casual interferences is based on the probabilistic calculus presented in [10]. We introduce here a slightly simplified version of this calculus by focusing on its main peculiarity, the non-atomicity of the output and input actions 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.

Syntax. The syntax of our calculus is shown in Table 1. 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 all networks, and **Loc** for the set of all locations.
Networks are collections of nodes, running in parallel and using channels to communicate messages. As usual, $\mathbf{0}$ denotes the empty network and $M_1|M_2$ the parallel composition of two networks. $n[P]_l$ 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. Notice that 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: $\mathbf{0}$ is the inactive process; $\mathtt{in}(c, \tilde{x}).P$ is ready to listen to a transmission, while $\mathtt{out}\langle c_r, \tilde{w}\rangle.P$ 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, 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 remaining syntactic forms are standard: $[w_1 = w_2]P, Q$ behaves as P if $w_1 = w_2$, and as Q otherwise. $A\langle\tilde{w}\rangle$ is the process defined via a (possibly recursive) definition $A(\tilde{x}) \stackrel{\text{def}}{=} P$, with $|\tilde{x}| = |\tilde{w}|$ where $\tilde{x}$ contains all channels and variables that are free in P.
Two further process forms arise in the dynamic of the calculus. In particular, processes that are ready to send or receive evolve into active senders and receivers:

P, Q	::=	$\ldots$	As in Table 1
	$\mid$	$c(\tilde{x}).P$	Active input
	$\mid$	$\bar{c}_r\langle\tilde{w}\rangle.P$	Active output

[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 as we will see later.

Networks		**Processes**	
M, N ::= $\mathbf{0}$	Empty network	P, Q ::= $\mathbf{0}$	Inactive process
$\mid n[P]_l$	Node (or device)	$\mid \mathtt{in}(c, \tilde{x}).P$	Input
$\mid (\nu c)M$	Channel restriction	$\mid \mathtt{out}\langle c_r, \tilde{w}\rangle.P$	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 1: Syntax

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, $\bar{c}_r\langle\tilde{w}\rangle.P$ 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.

Each node n is associated with a pair $\langle r_n, \mathbf{J}^n\rangle$, 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 $l \in \mathbf{Loc}$ and $\mathbf{J}^n_{lk} = 0$ for $l \neq k$. 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 $k \in \mathbf{Loc}$.

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' in n and by $[\![M]\!]_{\mu^n_l}$ the probability distribution over networks induced by μ^n_l and defined by: for all networks M',

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

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

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).

Reduction semantics. The dynamics of the calculus is specified by the *probabilistic reduction relation* ($\rightarrow$), described in Table 2: $M \rightarrow [\![M']\!]_\theta$ 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 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 $n_i \in I$ and $n_k \in K$ 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, as described by the (R-End-Bcast) rule.

Rule (R-Move) describes node mobility. A node n located at l and executing a move action will reach a location with probability described by the distribution μ^n_l 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 before starting a synchronization (when they are in a ready, but not active, state). This is a fairly common assumption in wireless networks, and a very reasonable one for all practical situations, in which wireless synchronization may be assumed to be orders of magnitude faster than node mobility.

All the remaining rules are standard, but a further remark is in order about the (R-Par) rule which may give rise to inconsistent network configurations. Indeed, 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.

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 [5], an execution for M is a (possibly infinite) sequence of steps $M \rightarrow_{\theta_1} M_1 \rightarrow_{\theta_2} M_2 \ldots$. We write $Exec_M$ 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 $M \rightarrow_{\theta_1} M_1 \ldots \rightarrow_{\theta_j} M_j$ of length j of the execution $e = M \rightarrow_{\theta_1} M_1 \cdots \rightarrow_{\theta_j} M_j \rightarrow_{\theta_{j+1}} M_{j+1} \cdots$,

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

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

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

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

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

Table 2: Reduction Semantics

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 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 [14]).
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:

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

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 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, $Exec_M^F(H) = \{e \in Exec_M^F \mid last(e^j) \in H$ 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 $Prob_M^F(H) = Prob_M^F(Exec_M^F(H))$.
In the rest of the paper, we restrict our attention 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. Formally, we restrict the set of all schedulers to the specific set of *admissible* schedulers. For this purpose, we introduce the following auxiliary operator: $\mathtt{Top}(\cdot)$ over networks. A channel c is at the top level of a network M, denoted $c \in \mathtt{Top}(M)$, if $M \equiv (\nu\tilde{d})(n[P]_l \mid N)$ and P is of the form $\mathtt{in}(c,\tilde{x}).Q$; $c(\tilde{x}).Q$; $\mathtt{out}\langle c_{L,r}, \tilde{w}\rangle.Q$; or $\bar{c}_{L,r}\langle\tilde{w}\rangle.Q$. Also remember that $\mathtt{A}(M)$ returns a network composed only by its currently active nodes.
A network M is *well-formed* if either $\mathtt{A}(M) \equiv \mathbf{0}$ or $\mathtt{A}(M) \equiv (\nu\tilde{d})\big(\prod_{i\in I} n_i[\bar{c}_{L_i,r_i}\langle\tilde{v}_i\rangle.P_i]_{l_i} \mid \prod_{j\in J} n_j[c(\tilde{x}_j).P_j]_{l_j} \mid \mathtt{A}(N)\big)$ 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 $d(l_i,l_j) \leq r_i$,
- $c \notin \mathtt{Top}(\mathtt{A}(N))$, and N is well-formed.

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.
We shall denote the set of admissible schedulers by $Sched$.

3. OBSERVATION EQUIVALENCES AND RESISTANCE TO JAMMING

We present a methodology, based on observation equivalences, for evaluating resistance to jammings of mobile wireless communications. Roughly speaking, we say that a network N, governed by a specific routing protocol, is resilient to a jamming attack if the behaviour of N in a context without the jammer is observational equivalent to the behaviour of N in the presence of the jammer.
Following a standard practice, we formalize the observational semantics of our calculus in terms of a notion *barb* [11]. We first introduce a notation for *strong barb*: for a network M, we write $M\downarrow_{c@K}$ when $M \equiv (\nu\tilde{d})(n[\bar{c}_r\langle\tilde{v}\rangle.P]_l \mid M')$, with $c \notin \tilde{d}$, $K \subseteq \mathbf{Loc}$ and for all $k \in K$, $d(l,k) \leq r$. We generalize this notion of barb to the general case of observations made dynamically along (well-formed) executions.

Definition 3.1. *[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 $H = \{M' \mid M \rightarrow^* M' \downarrow_{c@K}\}$.*

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. Consider, e.g., schedulers giving priority to movements over communications which possibly prevent any node transmission, or schedulers giving priority to end broadcasting

over begin broadcasting which will prevent any interference. In this paper we specialize the observational semantics given in [10] in order to compare the behaviour of networks relative to a restricted set of schedulers. Since our semantics is contextual, we need to ensure that the set of schedulers we consider allows the specific networks we analyze to interact with any possible context. Hence, for a given a set $\mathcal{F}$ of schedulers and a set $\mathcal{M} = \{M_1, M_2, ...\}$ of networks, we define the *contextual superset* $\mathcal{F}_{\mathcal{M}}$ of $\mathcal{F}$ relative to $\mathcal{M}$, as the largest set of schedulers allowing M_1, M_2, ... to interact with any possible context in the model even when driven by $\mathcal{F}$. Hereafter a context is a network term with a hole $[\cdot]$ defined by the grammar: $\mathcal{C}[\cdot] ::= [\cdot] \mid [\cdot]|M \mid M|[\cdot] \mid (\nu c)[\cdot]$.

DEFINITION 3.2. *Given a set of networks $\mathcal{M}$ and a set $\mathcal{F} \subseteq Sched$ of admissible schedulers, the* contextual superset *$\mathcal{F}_{\mathcal{M}}$ of $\mathcal{F}$ relative to $\mathcal{M}$ is defined as the largest subset of $Sched$, satisfying the following properties: $\forall F \in \mathcal{F}_{\mathcal{M}}$, either $F \in \mathcal{F}$ or $\exists F' \in \mathcal{F}$ such that, $\forall$ context $C[\cdot]$, $\forall e \in Exec^{F}_{C[O]}$, where $O \equiv M \in \mathcal{M}$, $C[O] = C_0[O_0]$ and $e = C_0[O_0] \rightarrow_{\theta_1} C_1[O_1] \rightarrow_{\theta_2} C_2[O_2] ... \rightarrow_{\theta_k} C_k[O_k]$, $\exists e' \in Exec^{F'}_{M}$ and $[j_0 - j_h]$ such that $e' = M_{j_0} \rightarrow_{\theta'_{j_1}} M_{j_1} \rightarrow_{\theta'_{j_2}} M_{j_2} ... \rightarrow_{\theta'_{j_h}} M_{j_h}$, and $\exists$ a monotonic surjective function f from $[0-k]$ to $[j_0 - j_h]$ such that for all $j \in [1-k]$, $O_{j-1} \equiv M_{f(j-1)}$, $O_j \equiv M_{f(j)}$ and for all $i \in [1-h]$, $\theta_{j_i} = \theta_j$ when $M_{f(j-1)} \rightarrow_{\theta'_{j_i}} M_{f(j)}$.*

Now, we introduce our probabilistic observational congruence relative to a specific set of schedulers.

DEFINITION 3.3. *Let $\mathcal{M}$ be a set of networks, $\mathcal{F} \subseteq Sched$, and $\mathcal{R}$ be a relation over networks.*

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

DEFINITION 3.4. *[Probabilistic observational congruence] Let $\mathcal{M}$ be a set of networks and $\mathcal{F} \subseteq Sched$.* Probabilistic observational congruence relative to $\mathcal{F}_{\mathcal{M}}$, *noted $\cong^{\mathcal{F}_{\mathcal{M}}}_{p}$, is the largest symmetric relation over networks which is reduction closed and barb preserving relative to $\mathcal{F}_{\mathcal{M}}$, and contextual.*

Two networks are related by $\cong^{\mathcal{F}_{\mathcal{M}}}_{p}$ if they exhibit the same probabilistic behaviour relative to the set of schedulers $\mathcal{F}_{\mathcal{M}}$. By abuse of notation, we write $N \cong^{\mathcal{F}}_{p} M$ when $\mathcal{M} = \{N, M\}$. In the next section we show how to use our framework for evaluating the resistance to jamming of a network N governed by a specific routing protocol. This is done by comparing the observational behaviour of N with the behaviour of the same network in the jamming context.

DEFINITION 3.5. *[Jamming-resistance] Let N be a network and $C[\cdot]$ be a context consisting of a malicious jamming node. Let $\mathcal{F}$ be a set of schedulers and $\tilde{c}$ be a set of channels. We say that N is* jamming-resistant *in the context $C[\cdot]$, relative to the set of schedulers $\mathcal{F}$ and independently from communications over channels $\tilde{c}$, if*

$$(\nu\tilde{c})N \cong^{\mathcal{F}}_{p} (\nu\tilde{c})C[N].$$

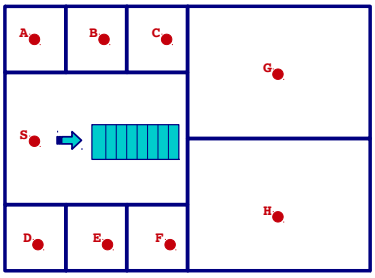

Figure 1: Topology of a building floor

The hiding operator $(\nu\tilde{c})$ is used here to specialize the verification method to some specific class of communications. It is worth of notice that the peculiarity of this approach is the analysis of a network governed by a specific protocol in a scenario: the outcomes of the analysis depend on the protocol definition and the mobility properties of the network's nodes. For instance, one may decide to analyse the connectivity of a network in which highly mobile nodes have the roles of routing packets that could be destroyed by jammers. The mobility of these nodes could ensure the resilience up to a given number of jammers.

4. A CASE STUDY

We apply the proposed methodology to a simple case study. We compare the resilience to jamming attacks of a wireless network governed by two different routing protocols.

Scenario. Let us consider an in-door MANET operating in a building of 25×25 meters with three floors whose height is 3 meters. For the sake of simplicity, we assume each floor to have the same topology: eight rooms connected by a central corridor as shown in Figure 1. A location is denoted by a pair $\langle f, r \rangle$ where $f \in \{1, 2, 3\}$ indicates the floor, while $r \in \{A, B, C, D, E, F, G, H, S\}$ indicates the room. Rooms A, B, C, D, E, F have dimension 5×5 meters, while G, H, S have dimension 10×15 meters.

The MANET consists of seven devices: three stationary nodes (n_1, n_2, n_3), located respectively in the first, second and third floor, which represent the servers inside the building, and four mobile nodes (n_4, n_5, n_6, n_7). Nodes can transmit with three transmission radii: $r_1 = 5m$, $r_2 = 10m$ and $r_3 = 15m$. We assume the network uses omnidirectional antennas, and adopt the euclidean function to evaluate the distance between two locations. Hence, a node sending a message with radius r_1 can reach only those nodes lying in the adjacent rooms, or in the same room of an adjacent floor, while using radius r_2 the successfulness of a reception depends on the kind of rooms the sender and the receivers are occupying. In order to simplify the computation of the distance between senders and receivers, we assume that each transmission always begins exactly from the centre of a room. In practice, if n_1 is located at $l = \langle 1, A \rangle$ and n_2 is located at $k = \langle 2, A \rangle$ then their distance is $d(l, k) = 3m$. Hence, they are able to mutually communicate by using any of the transmission radii in the set $\{r_1, r_2, r_3\}$. If n_1 is located at $l' = \langle 1, C \rangle$, then the Euclidean distance between l' and k is $d(l', k) = \sqrt{109}$. Since $10 < \sqrt{109} < 15$, n_1 and n_2 can communicate only using radius r_3. Note that these assumptions are intended to simplify the model. However, our cal-

culus may also deal with non-euclidean distances allowing us to take into account the effect of walls on the transmissions.

The HWMP protocol. The Hybrid Wireless Mesh Protocol (HWMP) is the routing protocol used by the standard IEEE 802.11s [6]. It offers a variety of routing strategies, including some optional ones. HWMP can be configured to operate in two modes: on-demand reactive mode and tree-based proactive mode.
The reactive mode is based on the Ad-hoc On-demand Distance Vector Routing protocol (AODV) [3, 12]. It uses three types of control packets whose forwarding allows each node to update its information about the best path to reach a specific destination. Each node maintains information about its neighbours in its *route table*, where each entry contains the following data associated with a given destination d.

- $seq\#_d$: the sequence number of the destination: it is incremented before sending a request, or (only if the node is the destination) before broadcasting the response;
- $\texttt{nexthop}_d$: the next intermediate node towards the destination d;
- $\texttt{hopcount}_d$: the number of hops from the source to the destination d;
- `lifetime`: the life-time of the record in the route table;
- $\texttt{List of precursors}_d$: the list of nodes using the actual node as `nexthop` to reach d.

The protocol exchanges three types of control packets:
- **RREQ** (Route Request): when a node needs to find a path, it broadcasts a RREQ message and then, when it receives the response, it chooses the cheapest path (in terms of energy costs, delays, number of hops, etc.);
- **RREP** (Route Response): When a node receives a RREQ message, it controls if it is the destination. In this case it immediately sends back the response, otherwise it searches a valid path in its *route table* to send back. If there are no valid paths, it propagates the RREQ packet;
- **RERR** (Route Error): the error message informing the network of a link breakage.

The tree-based proactive mode used by the HWMP protocol is based on a proactive protocol in which the network topology is built statically, forming a tree rooted in a chosen node. As in the previous mode, packet types, apart from root announcements, are RREQ, RREP and RERR. The difference lays in the fact that, in the tree-based approach, network topology is built statically from the start, and each node selects as the next hop the neighbour node that is nearest to the root according to its hop count, but it also maintains a table that contains the hop count to the root of its neighbours. The final result of these operations is that a spanning tree is created and used for the subsequent message forwarding. Whenever a node detects that the link with the upstream node, i.e., the node to which it would normally forward packets, is broken, it selects a new upstream node from its neighbours using the table of the hop counts, updates its own count and then broadcasts a RRER message to the downstream nodes. Then, these can select their own alternative paths if the hop count of the upstream node is no longer the shortest one. If a new path is chosen, the RERR message is propagated in turn to the downstream nodes. The effect of this action is to build a new spanning subtree. In what follows we assume that the root node, e.g., a gateway between a wired network and the wireless one, is chosen statically and that it never moves.

Modelling the system. We model our case study into the framework of Section 2. The set of feasible locations is:

$$\mathbf{Loc} = \{\langle f, r\rangle : f \in \{1,2,3\} \wedge r \in \{A,B,C,D,E,F,G,H,S\}\}\,.$$

Each node n_i is statically characterised by a pair $\langle r_{n_i}, \mathbf{J}^{n_i}\rangle$, where r_n is the maximum transmission radius, i.e., in our scenario $r_{n_i} = r_3$ for all n_i, with $i = 1, \ldots, 7$, and $\mathbf{J}^{n_i}$ is the transition matrix of a discrete time Markov chain denoting the probability of movements.
We model the HWMP protocol into our probabilistic calculus with the aim of comparing the resilience to jamming of both its proactive and reactive modes. In its simplest definition, the routing table maintained by each node has the form:

$$\langle d, seq\#_d, \texttt{nexthop}_d, \texttt{hopcount}_d, LP_d\rangle$$

where d is the primary key of the table and identifies the node name of the destination associated with the entry.
The packets used by HWMP are:
- $(\texttt{rreq}, \texttt{Bid}_{s,d}, (s, seq\#_s), (d, seq\#_d), \texttt{hopcount}_{s,d})$ for the request of a path from s to d;
- $(\texttt{rrep}, (s, seq\#_s), (d, seq\#_d), \texttt{hopcount}_{s,d})$ for the response to a path request from s to d;
- $(\texttt{rerr}, s, d, seq\#_d)$ denotes an error message;
- $(\texttt{ack}, d, seq\#_d)$ to confirm the correct reception of a route response.

For the sake of simplicity we abstract out the AODV packet management, and we consider the behaviour of the network assuming that a node needing a path to a given destination simply obtains it by calling the procedure $\texttt{find\_path}(s, d)$ where s and d are the source and destination nodes. The effect of this procedure is to update the node's routing table.
We consider the following network, where nodes n_4 and n_6 want to communicate, and the paths are the results of the routing protocol execution:

$$N = \prod_{i=1}^{7} (n_i[P_i]_{l_i})$$

where $\forall i \in [1-7], l_i \in \mathbf{Loc}$, and:

$$P_4 = \texttt{find\_path}(n_4, n_6).\texttt{out}\langle c_{r_2}, (msg, \texttt{nexthop}_{n_6}, n_6)\rangle.\\ \texttt{in}(c, (x_1, x_2, x_3)).[x_3 = n_4]\texttt{out}\langle \texttt{ok}_{r_2}, OK\rangle.P_4,$$

$$P_6 = \texttt{in}(c, (x_1, x_2, x_3)).[x_2 = n_6](\texttt{find\_path}(n_6, n_4).\\ \texttt{out}\langle c_{r_2}, (ack, \texttt{nexthop}_{n_4}, n_4)\rangle).P_6$$

and $\forall i \in \{1,2,3,5,7\}$:

$$P_i = \texttt{in}(c, (x_1, x_2, x_3)).[x_2 = n_i](\\ [x_3 = n_4](\texttt{find\_path}(n_i, n_4).\\ \texttt{out}\langle c_{r_2}, (x_1, \texttt{nexthop}_{n_4}, n_4)\rangle).P_i,\\ [x_3 = n_6](\texttt{find\_path}(n_i, n_6).\\ \texttt{out}\langle c_{r_2}, (x_1, \texttt{nexthop}_{n_6}, n_6)\rangle).P_i).$$

The processes depicted above describe the communication between n_4 and n_6. Node n_4 uses the function `find_path` to discover the path to n_6, and then forwards the packet. When it receives the acknowledgment it fires the message OK through the channel `ok`. Node n_6 waits for the packet from n_4 and, when it receives it, it sends back the acknowledgment. Each intermediate node executes a simple process forwarding both the packet sent by n_4 and the ack of n_6.
These processes will be used to study both the *AODV* and the *tree-based* protocols, the only difference is the way the

$$\text{(Beg-Out)}\ \frac{-}{\mathtt{out}\langle c_r, \tilde{v}\rangle.P \xrightarrow{\bar{c}_r} \bar{c}_r\langle\tilde{v}\rangle.P} \qquad \text{(End-Out)}\ \frac{-}{\bar{c}_r\langle\tilde{v}\rangle.P \xrightarrow{\bar{c}_r\tilde{v}} P}$$

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

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

Table 3: LTS rules for Processes

function `find_path` behaves: while in the *AODV* the path is discovered after the RREQ, RREP and RERR packages exchange, in the *tree-based* protocol, in order to find the best path, the process simply follows the predetermined routing spanning tree, built at the moment of the initial network setup through the algorithm informally described above. Thus, HWMP packets are sent only during the network setup or whenever a broken link is detected, causing the corresponding operations to happen.

We consider different jamming attacks. We focus our attention on proactive jammers, executing the following process: $P = \mathtt{out}\langle c_{r_2}, JAM\rangle.P$ which continuously broadcasts the dummy message JAM with radius r_2 on channel c. We consider two malicious nodes: a static jamming attacker m_1 identified by $\langle r_2, \mathbf{I}\rangle$, with $\mathbf{I}$ being the identity matrix, located at $k = \langle 1, H\rangle$, and a mobile node m_2 identified by $\langle r_2, \mathbf{J}^{m_2}\rangle$ whose initial location is $k' = \langle 3, G\rangle$. Note that the jammer m_1 is a node blocking the activity of each node lying within

$$\mathbf{Loc}_{m_1} = \{\langle 1, H\rangle, \langle 2, H\rangle, \langle 3, H\rangle, \langle 1, F\rangle, \langle 2, F\rangle, \langle 3, F\rangle\}.$$

We study the behaviour of the protocol in a context C_1 consisting of node m_1, and in a context C_2 consisting of both m_1 and m_2. We consider the set of schedulers $\mathcal{F}$ such that:

- during the forwarding of the packets (x_1, x_2, x_3) where $x_1 \in \{msg, ack\}$, movements of nodes n_i, $i \in [4-7]$ occurs at each collision or JAM-message reception,
- the beginning of output actions have priority on the ending of the output actions.

The first constraint allows us to model the fact that mobile nodes react to interference by moving away from their current location, while the second constraint is necessary since we are considering proactive jammers, which continue to send packets to provoke collisions.

Let $C_1[\cdot] = \cdot \mid m_1[P]_k$. The robustness of the network using the HMWP protocol against the malicious node m_1 can be verified by checking if the observational behaviour of N is independent of the presence of the jammer inside the building. Formally, we have to prove that:

$$(\nu c)N \cong_p^{\mathcal{F}} (\nu c)C_1[N].$$

The restriction operator νc is due to the fact that we want to observe only the correctness of the communication between the nodes n_4 and n_6, without considering the different paths they may choose for the message forwarding. Then, since c is hidden, the only observable action in $(\nu c)N$ and $(\nu c)C_1[N]$ is $\xrightarrow{\mathtt{ok}!OK@K}$ for some $K \subseteq \mathbf{Loc}$.

Moreover, we consider the dynamic context:

$$C_2[\cdot] = \cdot \mid m_1[P]_k \mid m_2[P]_{k'}$$

where m_2 is a mobile node, meaning that the *jamming area* may change in time. We also consider the same restricted set of schedulers $\mathcal{F}$ as above and we aim at observing the successfulness of the communication between n_4 and n_6. Hence, we have to prove that:

$$(\nu c)N \cong_p^{\mathcal{F}} (\nu c)C_2[N].$$

5. A BISIMULATION PROOF TECHNIQUE

In this section we define a bisimulation-based proof technique for $\cong_p^{\mathcal{F}\mathcal{M}}$. It provides an efficient method to check whether two networks are related by $\cong_p^{\mathcal{F}\mathcal{M}}$.

We define a LTS semantics for our calculus, which is built upon two sets of rules: one for processes and one for networks. Table 3 presents the LTS rules for processes. Transitions are of the form $P \xrightarrow{\eta} P'$, where η ranges over input and output actions, defined as follows:

$$\eta ::= c \mid c\vartheta \mid \bar{c}_r \mid \bar{c}_r\tilde{v} \quad \text{with} \quad \vartheta ::= \tilde{v} \mid \bot.$$

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 $\bot$ due to a collision.

Table 4 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 the labels:

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

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

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

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

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

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

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

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

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

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

Table 4: LTS rules for Networks

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". 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 locations 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 $c?@l$; $c?\vartheta@l$; $c![l,r]$; $c!\tilde{v}[l,r]$; 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 bisimulation relative to a set of schedulers $\mathcal{F}_\mathcal{M}$. It is built upon the following actions:

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

Again, we write $M \xrightarrow{\alpha}_\theta N$ if $M \xrightarrow{\alpha} [\![M']\!]_\theta$ and N is in the support of $[\![M']\!]_\theta$. 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 define $Exec^F_M$ 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* as:

- $\Longrightarrow$ 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 $\Longrightarrow \xrightarrow{c!\tilde{v}@K \triangleleft R} \Longrightarrow$.
- $\xRightarrow{\hat{\alpha}}$ denotes $\Longrightarrow$ when $\alpha = \tau$ and $\xRightarrow{\alpha}$ when $\alpha \neq \tau$.

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

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

transitions but also perform these transitions with matching probabilities. To achieve this, we denote by $Exec^F_M(\stackrel{\alpha}{\Longrightarrow}, H)$ 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 the probability of reaching a network in H from M by performing $\stackrel{\alpha}{\Longrightarrow}$, according to a scheduler F as $Prob^F_M(\stackrel{\alpha}{\Longrightarrow}, H) = Prob^F_M(Exec^F_M(\stackrel{\alpha}{\Longrightarrow}, H))$. We are interested in a notion of bisimilarity which is a complete characterisation of our observational congruence. Since the latter is relative to a set of schedulers $\mathcal{F}_{\mathcal{M}}$, we need to define the corresponding set $\hat{\mathcal{F}}_{\mathcal{M}}$ for the LTS semantics (see [2] for a formal definition).
Now we introduce our probabilistic labelled bisimilarity relative to a given set of schedulers.

DEFINITION 5.1. *[Probabilistic labelled bisimilarity] Let M and N be two networks. An equivalence relation $\mathcal{R}$ over networks is a* probabilistic labelled bisimulation *relative to $\hat{\mathcal{F}}_{\mathcal{M}}$ if $M\mathcal{R}N$ implies: for all scheduler $F \in \hat{\mathcal{F}}_{\mathcal{M}}$ there exists a scheduler $F' \in \hat{\mathcal{F}}_{\mathcal{M}}$ such that for all α and for all classes $\mathcal{C}$ in $\mathcal{N}/\mathcal{R}$ it holds:*

1. *if $\alpha = \tau$ or $\alpha = c!\tilde{v}@K \triangleleft R$ then $Prob^F_M(\stackrel{\alpha}{\rightarrow}, \mathcal{C}) = Prob^{F'}_N(\stackrel{\hat{\alpha}}{\Longrightarrow} \mathcal{C})$;*
2. *if $\alpha = c?@l$ or $\alpha = c?\vartheta@l$ then either $Prob^F_M(\stackrel{\alpha}{\rightarrow}, \mathcal{C}) = Prob^{F'}_N(\stackrel{\alpha}{\Longrightarrow}, \mathcal{C})$ or $Prob^F_M(\stackrel{\alpha}{\rightarrow}, \mathcal{C}) = Prob^{F'}_N(\Longrightarrow, \mathcal{C})$.*

Probabilistic labelled bisimilarity *relative to $\hat{\mathcal{F}}_{\mathcal{M}}$, written $\approx_p^{\hat{\mathcal{F}}_{\mathcal{M}}}$, is the largest symmetric probabilistic labelled bisimulation relative to $\hat{\mathcal{F}}_{\mathcal{M}}$ over networks.*

Probabilistic labelled bisimilarity is a complete characterisation our probabilistic observational congruence.

THEOREM 5.2. *Let $\mathcal{F}$ be a set of schedulers and $\mathcal{M}$ be a set of networks. Let M and N be two well-formed networks. $M \approx_p^{\hat{\mathcal{F}}_{\mathcal{M}}} N$ if and only if $M \cong_p^{\mathcal{F}_{\mathcal{M}}} N$.*

6. RESILIENCE OF THE REACTIVE AND PROACTIVE MODES TO JAMMING

In order to verify resilience of our network case study with respect to the jamming context C_1, we check whether

$$(\nu c)N \cong_p^{\mathcal{F}} (\nu c)C_1[N]. \quad (1)$$

By using the proof technique presented in Section 5, it is sufficient to find a probabilistic bisimulation containing the pair $((\nu c)N, (\nu c)C_1[N])$. Let us consider the relation

$$\mathcal{R} = \{((\nu c)\bar{N}, (\nu c)\bar{C}_1[\bar{N}]) : (\nu c)\bar{C}_1[\bar{N}] \in \bar{\mathcal{M}}\}$$

where $\bar{\mathcal{M}} = \{(\nu c)\bar{C}_1[\bar{N}] : (\nu c)C_1[N] \rightarrow^* (\nu c)\bar{C}_1[\bar{N}]\}$. In order to prove (1), it is enough to prove that $\mathcal{R}$ is a bisimulation relative to $\hat{\mathcal{F}}_{\mathcal{M}}$ with $\mathcal{M} = \{(\nu c)N, (\nu c)C_1[N])\}$. Since in $\bar{\mathcal{M}}$ channel c is hidden, the only actions that $\bar{N}$ can do are τ actions, or output through the channel `ok`, while the jamming context $\bar{C}_1[\cdot]$ can only make τ actions. Let us consider $(\nu c)\bar{N} \stackrel{\tau}{\rightarrow} [\![(\nu c)\bar{N}']\!]_\theta$ driven by $F \in \hat{\mathcal{F}}_{\mathcal{M}}$. Then, $\forall \mathcal{C} \in \mathcal{N}/\mathcal{R}$, $Prob^F_{(\nu c)\bar{N}}(\stackrel{\tau}{\rightarrow}, \mathcal{C}) =$

$$\sum_{(\nu c)\bar{N}'' \in \mathcal{C} \text{ in the support of } [\![(\nu c)\bar{N}']\!]_\theta} [\![(\nu c)\bar{N}']\!]_\theta((\nu c)\bar{N}'').$$

If the τ action is due to the application of the rule (Move), we conclude by applying rule (Par) to m_1.

If $(\nu c)\bar{N} \stackrel{\tau}{\rightarrow} (\nu c)[\![\bar{N}']\!]_\Delta$, because of an application of rule (Beg-Bcast) then $\bar{N} \stackrel{c![l,r_2]}{\longrightarrow} [\![\bar{N}']\!]_\Delta$ for some location l. If $\bar{C}_1[\mathbf{0}] \equiv C_1[\mathbf{0}]$, or $d(l,k) > r_2$ we are done, because it is enough to apply rule (Par), since both $\bar{C}_1[\bar{N}]$ and $\bar{C}_1[\bar{N}']$ are well-formed networks, and $\exists F' \in \hat{\mathcal{F}}_{\mathcal{M}}$ such that $Prob^F_{(\nu c)\bar{N}}(\stackrel{\tau}{\rightarrow}, \mathcal{C}) = Prob^{F'}_{(\nu c)\bar{C}_1[\bar{N}]}(\Longrightarrow, \mathcal{C})$, as required. If $\bar{C}_1[\mathbf{0}] \equiv C'_1[\mathbf{0}]$ and $d(l,k) \leq r_2$, we can not apply rule (Par) because $C'_1[\bar{N}']$ is not well-formed (i.e., there are two active senders whose distance is smaller of their transmission radius). Now, in order to prove bisimulation, we have to find in $\hat{\mathcal{F}}_{\mathcal{M}}$ a scheduler allowing $C'_1[\bar{N}]$ to reach $\bar{C}_1[\bar{N}']$ with probability 1, where $\bar{C}_1[\cdot] \in \{C_1[\cdot], C'_1[\cdot]\}$. In this case, there exists F' allowing m_1 to finish its communication, i.e., $C_1[\bar{N}] \stackrel{\tau}{\rightarrow} [\![C_1[\bar{N}']]\!]_\Delta$.
The most interesting case is when $d(k,l_h), d(l,l_h) \leq r_2$ for some $h \in [1{-}7]$, where $\bar{N} \equiv \prod_{i=1}^7 n_i[Q_i]_{l_i}$ and $Q_h = c(\tilde{x}).Q'_h$. This means that n_h is in the jammer's transmission area, and it receives the correct packet in $\bar{N}$, while it receives a collision in $C'_1[\bar{N}]$ avoiding it to immediately reach $\bar{C}_1[\bar{N}']$ with probability 1. If the node receiving the collision ($\perp$) is an intermediate node in the path from n_4 to n_6, we are done, since, when n_4 does not receive the acknowledge, it searches another path to reach n_6 (by re-executing `find_path`) and it sends again the message. If the node receiving the collision is the source or the destination of the communication (n_4, n_6), the bisimulation depends on the transition matrix modelling its mobility: if the node, with a finite number h of steps, goes far away from the jammer with probability 1, i.e., the probability to end up in an ergodic set in which all states represents locations inside the jammed area is 0, then the bisimulation is proved, otherwise the probability for $C'_1[\bar{N}]$ to reach a state in $\mathcal{C}$ is $1-p$, where $p = \sum_{l' \in \mathbf{Loc}_{jam}} \mathbf{J}^{n_i\ (h)}_{ll'}$ and $\mathbf{J}^{n_i}$ being the transition matrix associated with the node n_i.
Moreover, when the protocol uses the tree-based strategy, assuming that root nodes are static (in our example nodes n_1, n_2 and n_3), e.g., when those nodes are bridges between wireless and wired networks, or access points, if the node receiving the collision is the root itself, it can never be possible to detect an alternative route, thus the bisimulation does not hold.

Let us now consider now the dynamic context:

$$C_2[\cdot] = \cdot \mid m_1[P]_k \mid m_2[P]_{k'}.$$

We have to prove that:

$$(\nu c)N \cong_p^{\mathcal{F}} (\nu c)C_2[N]. \quad (2)$$

Again, we check whether the relation

$$\mathcal{S} = \{((\nu c)\bar{N}, (\nu c)\bar{C}_2[\bar{N}]) : (\nu c)\bar{C}_2[\bar{N}] \in \mathcal{M}\},$$

where $\mathcal{M} = \{(\nu c)\bar{C}_2[\bar{N}] : (\nu c)C_2[N] \rightarrow^* (\nu c)\bar{C}_2[\bar{N}]\}$, is a bisimulation. The proof proceeds as for the previous context (the behaviours of N and $C_2[N]$ only differ for the collisions caused by the jammers attacks), but we have to take into account that the second jammer is mobile: if in the previous case we had only to verify the positions of n_4 and n_6, while the jamming area was known statically, now we have to analyse all the possible locations occupied by the mobile jammer.
Let the Markov chain $\mathbf{J}^{n_i m_2}$ be the joint process between $\mathbf{J}^{n_i}$ and $\mathbf{J}^{m_2}$, in which states are pairs (l,k) of states of the

chains $\mathbf{J}^{n_i}$ and $\mathbf{J}^{m_2}$, which are associated with nodes n_i and m_2 (the mobile jammer), respectively. Hence, each pair (l, k) of $\mathbf{J}^{n_i m_2}$ means that n_i is located at l and m_2 is located at k. The state space of $\mathbf{J}^{n_i m_2}$ is thus the Cartesian product of the state spaces of $\mathbf{J}^{n_i}$ and $\mathbf{J}^{m_2}$. We consider a subset $\mathbf{S}^{n_i m_2}$ of $\mathbf{J}^{n_i m_2}$, such that $(l, k) \in \mathbf{S}^{n_i m_2} \Leftrightarrow d(l, k) \geq r_{m_2}$, i.e., all combinations of states in which the jammer can interfere with the node n_i. n_i can always communicate successfully (and then the bisimulation is proved) only if the probability to end up in an ergodic set in which all states are member of $\mathbf{S}^{n_i m_2}$ is 0.

Again, when using the tree-based strategy, if the root nodes are stationary, even if n_4 and n_6, when jammed, are able to reach a safe location with probability 1, the success of the communication depends on the root nodes, and if the root nodes are jammed, n_4 and n_6 may not be able to find a valid path to complete their communication.

Even if both relations $\mathcal{R}$ and $\mathcal{S}$ are proved to be bisimulations only under particular conditions depending on the mobility of nodes, with our proof technique we have been able to show that the reactive approach used by the HWMP protocol is more robust against jamming than the proactive one, since, when using the tree-based routing, the bisimulations are proved under more restricted conditions, which depend on the mobility behaviour of the root nodes.

7. CONCLUSION

In this paper we addressed the problem of proving the resilience of a MANET to jamming attacks. To this aim, we presented a probabilistic calculus that allows the evaluation of the level of interference within a MANET and the decision of the connectivity of a network under different scenarios. In particular, we proposed a formal approach to the evaluation of the robustness of a MANET in case of jamming attacks, both when the attackers is a static node and when it is mobile. We showed our calculus at work for the analysis of a case study. Specifically, we encoded two protocols used for the routing in MANETs and modelled a simplified scenario with both static a mobile nodes. Applying the proof methodology presented in Section 5, we compared the robustness of the network to jamming attacks when each of the two routing protocols are adopted. As concerns future works, a tool based on PRISM [8] for the analysis of mobile ad-hoc networks based on our calculus is currently being developed.

8. REFERENCES

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

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

[3] I. D. Chakeres and E. M. Belding-Royer. Aodv routing protocol implementation design. In *Proc. of 24th International Conference on Distributed Computing Systems Workshops - W7: EC (ICDCSW'04)*, volume 7, pages 698–703. IEEE, 2004.

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

[5] 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 of *LNCS*, pages 175–190. Springer-Verlag, 2007.

[6] G. Hiertz, D. Denteneer, S. Max, R. Taori, J. Cardona, L. Berlemann, and B. Walke. Ieee 802.11s: the wlan mesh standard. *Wireless Communications, IEEE*, 17(1):104–111, 2010.

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

[8] M. Kwiatkowska and D. Parker. Advances in probabilistic model checking. In *Proc. 2011 Marktoberdorf Summer School: Tools for Analysis and Verification of Software Safety and Security*. IOS Press, 2012. To appear.

[9] I. Lanese and D. Sangiorgi. An operational semantics for a calculus for wireless systems. *Theoretical Computer Science (TCS'10)*, 411(19):1928 – 1948, 2010.

[10] M.Bugliesi, L.Gallina, S. Hamaodu, A. Marin, and S. Rossi. Interference-sensitive preorders for manets. In *Proc. 9th International Conference on Quantitative Evaluation of SysTems*. IEEE, 2012. To appear.

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

[12] E. M. Royer and C. E. Perkins. Multicast operation of the ad-hoc on-demand distance vector routing protocol. In *Proc. of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '99)*, pages 207–218. ACM, 1999.

[13] A. Safwat, H. Hassanein, and H. Mouftah. Energy-aware routing in MANETs: analysis and enhancements. In *Proc. of the 5th ACM int. workshop on Modeling analysis and simulation of wireless and mobile systems*, MSWiM '02, pages 46–53, Atlanta, Georgia, USA, 2002. ACM.

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

[15] M. Strasser, B. Danev, and S. Čapkun. Detection of reactive jamming in sensor networks. *ACM Trans. Sen. Netw.*, 7:16:1–16:29, 2010.

[16] W. Xu, W. Trappe, Y. Zhang, and T. Wood. The feasibility of launching and detecting jamming attacks in wireless networks. In *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pages 46–57, New York, NY, USA, 2005. ACM.