Modelling probabilistic wireless networks (Extended Abstract)

Andrea Cerone and Matthew Hennessy

Department of Statistics and Computer Science
Trinity College Dublin
ceronea@scss.tcd.ie, Matthew.Hennessy@scss.tcd.ie,

Abstract. We propose a process calculus to model distributed wireless networks. The calculus focuses on high-level behaviour, emphasising local broadcast communication and probabilistic behaviour.
Our formulation of such systems emphasises their *interfaces*, through which their behaviour can be observed and tested, although this complicates their contextual analysis. Nevertheless we propose a novel operator with which networks can be decomposed into components. Using this operator we define probabilistic generalisations of the well-known may-testing and must-testing preorders.
We define an extensional probabilistic labelled transition system in which actions represent particular interactions networks support via their interfaces. We show that novel variations on probabilistic simulations support compositional reasoning for these networks which are sound with respect to the testing preorders. Finally, and rather surprisingly, we show that these simulations turn out not to be complete.

1 Introduction

There is growing interest in the development of formal methods for the analysis of wireless systems, and a number of process calculi have been suggested for describing and analysing their behaviour, [10,11,13]. Our proposal focuses on descriptions at a high-level of abstraction, where for example network nodes use protocols at the *MAC level* [7] to implement reliable communication between nodes; thus we are abstracting from collision prone behaviour. For us a wireless system will take the form $\mathcal{M} = \Gamma \rhd M$ where Γ describes the network topology, a connected undirected graph of station nodes; some nodes will contain running code, while others will be in the system interface, $\mathsf{Int}(\mathcal{M})$, through which the system may be tested, or indeed composed with peers to form larger systems. The running code at individual stations is described in the component M, using essentially a broadcast version of CCS, [12,15].

However the range of a broadcast from a given station node is determined by the underlying connectivity graph Γ. Further, we allow the code at stations to behave probabilistically. We also assume that a fixed number of communication channels are available to stations to broadcast to their neighbours; it is well-known that multiple access techniques such as *TDMA* and *FDMA* [17] can

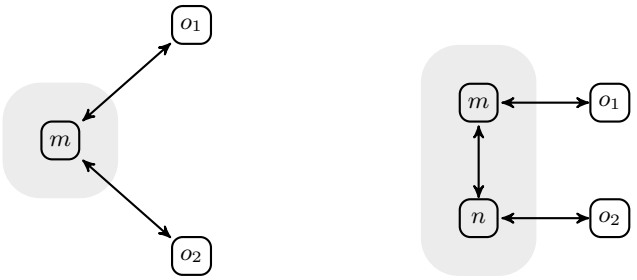


Fig. 1. Example networks

be used to implement such virtual channels. In the literature other calculi for modelling wireless systems have been proposed; in particular, our calculus has been inspired by [10,11,8,13]. Recently, there has been also a growing interest in modeling networks with probabilistic behaviour [9,5,6].

Two example systems in our calculus are given in Figure 1; here and henceforth we use shading to denote nodes running code in a network, with the remaining being in the interface. Our goal is to develop behavioural theories for such systems, and associated proof technologies.

Using standard process-calculi techniques we can give an intensional semantics to the set of such (well-formed) networks Nets thereby endowing it with the structure of a probabilistic labelled transition system [16], a pLTS; see Section 3. This is a significant step towards our goal as in [3] behavioural testing preorders have been defined for arbitrary pLTSs, and forms of (probabilistic) simulations have been shown to be both sound and complete with respect to them. However significant problems arise when trying to adapt this approach to Nets.

In [3] a total binary operator $|$ is assumed to exist for arbitrary systems; a system $\mathcal{S}$ is tested by running the combined system $\mathcal{S}\,|\,\mathcal{T}$ and observing the effect on the testing system $\mathcal{T}$; indeed all standard process calculi come equipped with such an operator. However there is no definitive manner in which arbitrary pairs of wireless systems from Nets can be combined, so as to maintain consistency. For example both $\mathcal{S}$ and $\mathcal{T}$ might expect to run their own code at a particular station they have in common; or in the combined system natural well-formedness conditions might be violated. Our intention with such an operator is to implement *black-box testing* of $\mathcal{S}$ by $\mathcal{T}$; so for example $\mathcal{T}$ should have no access to, or indeed knowledge of, the internal stations in $\mathcal{S}$. Instead interaction between $\mathcal{T}$ and the system $\mathcal{S}$ will be restricted to what we will call *the interface* of $\mathcal{S}$.

With this in mind, in Section 4 we propose a novel asymmetric combinator for (well-formed) wireless networks, $\mathcal{S} \Vdash \mathcal{T}$; this in turn leads to formulations of the standard testing pre-orders to Nets, $\mathcal{S}_1 \sqsubseteq_{may} \mathcal{S}_2$ and $\mathcal{S}_1 \sqsubseteq_{must} \mathcal{S}_2$. The asymmetry is necessary; in Theorem 2 we show that if any *reasonable* symmetric combinator were used then the resulting pre-orders would be degenerate.

We then give, in Section 5, an extensional pLTS in which the probabilistic simulations are sound with respect to $\mathcal{S}_1 \sqsubseteq_{may} \mathcal{S}_2$. Here the extensional actions are defined in terms of behaviour, broadcasts and reception of values, which can be detected at the interface of systems, $\mathsf{Int}(\mathcal{S})$. However again this is not simply a straightforward application of the simulations from [3]. A problem arises because in certain situations the broadcast of a message to a set of interface nodes, as might happen in $\mathcal{M}$ of Figure 1 from m to the pair $\{o_1, o_2\}$, can be simulated by a multicast of copies of the message through a series of nodes. In $\mathcal{N}$ from Figure 1 this might happen by a broadcast from m, with reaches the interface node o_1 and the internal node n, followed by a broadcast by n to the second interface node o_2.

We also provide a variation on simulations, in the same extensional pLTS, which are sound with respect to $\mathcal{S}_1 \sqsubseteq_{must} \mathcal{S}_2$. Again results in [3] can not be relied upon. Instead we define a novel notion of *deadlock simulation* for this purpose.

We already know that our notion of simulations are not complete for wireless systems; an example is given at the end of Section 5. Nevertheless we believe that they are powerful enough to treat non-trivial case studies. In Section 6 we provide one example which shows the way in which they can be used. In future we intend to evaluate more fully our proposed methodology.

In this extended abstract we omit all proofs, and some technical definitions are also elided. Full details are available in the accompanying technical report [1], together with some more illustrative examples.

2 Background

Recall that a probability distribution Δ over a set S is a function $\Delta : S \to [0, 1]$ such that $\sum_{s \in S} \Delta(s) = 1$. Given a set S, we use $\mathcal{D}(S)$ to denote the set of probability distributions over S.

Definition 1. A *probabilistic labelled transition system* (pLTS) is a 4-tuple $\langle S, \mathsf{Act}_\tau, \to, \omega \rangle$, where

(i) S is a set of states,
(ii) Act_τ is a set of transition labels with a distinguished label τ,
(iii) the relation $\to$ is a subset of $S \times \mathsf{Act}_\tau \times \mathcal{D}(S)$,
(iv) $\omega : S \mapsto \{ \text{ true }, \text{ false } \}$ is a (success) predicate over the states S.

As usual, we will write $s \xrightarrow{\mu} \Delta$ in lieu of $(s, \alpha, \Delta) \in \longrightarrow$. It is finitary if S is finite and for every $s \in S$, the set $\{ \Delta \mid s \xrightarrow{\mu} \Delta \text{ for some } \mu \in \mathsf{Act}_\tau \}$ is finite. □

This definition of pLTS is slightly different from that provided in [3], for we have introduced a success predicate ω over states, which will be used when testing processes.

We use standard notation, borrowed from [3], for distributions and operations on them. We use Δ, Θ to range over probability distributions; $\lceil \Delta \rceil$ represents the *support* of Δ, that is all states such that $\Delta(s) > 0$ while $\overline{s}$ denotes the one point

distribution for an $s \in S$. We will also have a minor need for *sub-distributions*, with $\mathcal{D}_{sub}(S)$ representing the sub-distributions over S; for $\Delta \in \mathcal{D}_{sub}(S)$, the quantity $\sum_{s \in S} \Delta(s)$, called the size of the sub-distribution, may be strictly less then 1.

Definition 2 (Lifted Relations). *Let $\mathcal{R} \subseteq S \times \mathcal{D}_{sub}(S)$ be a relation from states to subdistributions. Then $\overline{R} \subseteq \mathcal{D}_{sub}(S) \times \mathcal{D}_{sub}(S)$ is the smallest relation which satisfies*

- *$s \,\mathcal{R}\, \Delta$ implies $s \,\overline{\mathcal{R}}\, \Delta$*
- *If I is a finite index set and $\Delta_i \;\overline{\mathcal{R}}\; \Theta_i$ for each $i \in I$ then $(\sum_{i \in I} p_i \cdot \Delta_i) \,\mathcal{R}\, (\sum_{i \in I} p_i \cdot \Theta_i)$ whenever $\sum_{i \in I} p_i \leq 1$.* □

Lifting of relations can also be defined for full distributions, by simply requiring $\sum_{i \in I} p_i = 1$ in the last constraint of the definition above.

In a pLTS $\langle S, \mathsf{Act}_\tau, \rightarrow, \omega \rangle$, each transition relation $\xrightarrow{\mu} \subseteq S \times \mathcal{D}(S)$ can be lifted to $\overline{(\xrightarrow{\mu})} \subseteq \mathcal{D}(S) \times \mathcal{D}(S)$. With an abuse of notation, the latter is still denoted as $\xrightarrow{\mu}$.

Lifted transition relations allow us to reason about the behaviour of pLTSs in terms of sequences of transitions. We also need to formalise internal computations, indefinite sequences of τ actions. We employ the infinitary version $\Delta \Longrightarrow \Delta'$ from [3], in which states from the support of Δ may at any point decide to stop performing τ actions; in general Δ' may turn out to be a sub-distribution, rather than a distribution; this is because part of a distribution may never stop performing τ-actions. A minor variation, $\Delta \Longrightarrow\!\!\!\!\succ \Delta'$, insists that states must continue performing τ actions so long as they are able; intuitively $\Delta \Longrightarrow\!\!\!\!\succ \Delta'$ may be viewed as a probabilistic version of a maximal computation from Δ. The formal definitions are relegated to the appendix; note the presence of the success predicate ω in a pLTS means that our formulation is a slight generalisation from that in [3].

3 Networks and their computations

As explained in the Introduction a wireless system is represented by a pair $\Gamma \rhd M$ where Γ is an undirected graph representing the connectivity in the underlying network between the wireless stations and M the code running in the individual stations. The language for code is given in Figure 2 and uses standard syntax from process calculi. Basically a system consists of a collection of named nodes at each of which there is some running code, $n[\![s]\!]$. The set of nodes appearing in a system M is denoted by $\mathrm{nodes}(M)$.

The process calculus operators $c!\langle v \rangle.p$, $c?(x).p$ will represent the broadcast and reception of values respectively; the latter is a binding operator for the variable x, and the standard notions of free occurrences of a variable as well as closed system terms arise. As usual, given a list of variables $\tilde{x}$ and a list of closed values $\tilde{v}$ of the same length, we use the notation $p\{\tilde{v}/\tilde{x}\}$ to denote process p where

$M,\ N ::=$		**Systems**
	$n[\![s]\!]$	Nodes
	$M \mid N$	Composition
	$\mathbf{0}$	Identity

$p,\ q ::=$		(probabilistic) Processes
	s	
	$p\ {}_p\!\oplus\ q$	probabilistic choice

$s,\ t ::=$		States
	$c!\langle e\rangle\,.p$	broadcast
	$c?(x)\,.p$	receive
	$\omega.\,\mathbf{0}$	test
	$s + t$	choice
	$\mathsf{if}\ b\ \mathsf{then}\ s\ \mathsf{else}\ t$	branch
	$\tau.p$	internal activities
	$A(\tilde{x})$	calls
	$\mathbf{0}$	terminate

Fig. 2. Syntax

the free occurrences of a variable x appearing in $\tilde{x}$ is replaced with the respective closed value v appearing in $\tilde{v}$. We also assume a set of process definitions of the form $A(\tilde{x}) \Leftarrow p$, meaning that the definition $A(\tilde{v})$ can be unfolded in $p\{\tilde{v}/\tilde{x}\}$.

The effect of a broadcasts is determined by the underlying network Γ. For example if a value is broadcast from the station n then it can only be received at stations m connected to n in Γ; that is those m such that $(n, m) \in \Gamma_E$ where Γ_E is the set of edges of Γ. For this, and similar concepts, we tend to use more graphic notation such as $\Gamma \vdash n \leftrightarrow m$.

We only consider the sublanguage of well-formed terms, in which each node name has at most one occurrence, and we use sSys to denote the set of all well-formed terms which are closed, meaning that they have no free occurrences of a free variables. Nodes appearing in $\mathrm{nodes}(M)$ in a network $\Gamma \rhd M$ are called internal, in contrast with nodes in $\Gamma_V \setminus \mathrm{nodes}(M)$ which are called external. The set $\Gamma_V \setminus \mathrm{nodes}(M)$ is also called the interface of the network, denoted as $\mathsf{Int}(\Gamma \rhd M)$. A network $\Gamma \rhd M$ is well-formed if:

(i) $M \in \mathsf{sSys}$
(ii) $\mathrm{nodes}(M) \subseteq \Gamma_V$, where Γ_V denotes the set of nodes in Γ
(iii) whenever $k \in \mathsf{Int}(\Gamma \rhd M)$, there exists some $m \in \mathrm{nodes}(M)$ such that $\Gamma \vdash k \leftrightarrow m$
(iv) whenever $k_1, k_2 \in \mathsf{Int}(\Gamma \rhd M)$, $\Gamma \vdash k_1 \not\leftrightarrow k_2$.

Most of these conditions are natural; in particular, requirements (iii) and (iv) establish that internal nodes (that is, nodes running code) in a network have

knowledge of the nodes in the external environment to which they are connected, but they have no information about how these nodes are interconnected. Requirement (iv) is also necessary for the soundness of our proof methodologies. We use Nets to denote the set of well-formed networks, in the sequel ranged over by $\mathcal{M}, \mathcal{N}, \ldots$. We will also use some obvious notation, such as nodes($\mathcal{M}$) to denote the set of nodes running code in $\mathcal{M}$.

Example 1. Consider $\mathcal{M} = \Gamma_M \rhd M$ described in Figure 1, where M denotes the code $m[\![\tau.(c!\langle v\rangle \,.\, \mathbf{0}\,{}_{0.81}\!\oplus\, \mathbf{0})]\!]$. Intuitively in this network, the station m, after performing some internal computation can broadcast a value v along channel c with probability 0.81. This message can be detected by the interface nodes o_1 and o_2.

Consider now network $\mathcal{N} = \Gamma_N \rhd N$, in the same figure, where N denotes $m[\![\tau.(c!\langle v\rangle\,{}_{0.9}\!\oplus\, \mathbf{0})]\!] \mid n[\![P]\!]$, and $P \Leftarrow c?(x)\,.(c!\langle x\rangle\,{}_{0.9}\!\oplus\, \mathbf{0}) + c?(x)\,.P$. Here station m broadcasts the value v along channel c with probability 0.9; this message can be detected by the interface node o_1 and the internal station n. Station n, upon receiving the message, decides to forward it with probability 0.9; since the nodes in the range of n are m and o_2, these are the nodes which will detect the value broadcast by n. Therefore, the probability of the original broadcast performed by station m reaching both the interface nodes o_1 and o_2 is 0.81, the same as in the network $\mathcal{M}$.[1] Note also that node n can non-deterministically decide to ignore broadcasts along channel c which can be received either by node m or by the interface node o_2. This ensures that the network $\mathcal{N}$ has a computation in which its behaviour is not affected by the external nodes o_1, o_2. Informally speaking, the network $\mathcal{N}$ is more reliable than the network $\mathcal{M}$, when the latter is viewed optimistically. □

Judgements in the formal intensional semantics of networks take the form $\Gamma \rhd M \xrightarrow{\mu} \Delta$, where Δ is a distribution over sSys and μ can take one of the forms: $n.\tau$ internal computation at node n, $c.n?v$ reception from node n of value v or $n.c!v$, transmission from node n.

The rules for inferring judgements are given in Figure 3 and they rely on a pre-semantics for the states s from Figure 4. These in turn take the form $s \xrightarrow{\mu} p$, where s is a closed state, p is a process and μ is one of the forms $c!v$, $c?v, \tau$ or ω. The deductive rules for inferring these judgements are given in Figure 4 and should be self-explanatory. The main rules in Figure 3 also use some standard notation from [3] for interpreting processes p from Figure 2 as distributions over states, $[\![p]\!]$; this has the obvious definition, namely $[\![s]\!] = \overline{s}$ and $[\![p_1\,{}_p\!\oplus\, p_2]\!] = p \cdot [\![p_1]\!] + (1-p) \cdot [\![p_2]\!]$.

Rule (B-BROAD) models the evolution of a node n which broadcasts value v along channel c. Here the term $n[\![\Delta]\!]$ represents a distribution over sSys, obtained by extending the function $n[\![\cdot]\!]$ in the standard way to distributions. This technique is also used in subsequent rules, for example extending the operator $\mid$ from one on system terms to distributions over system terms.

[1] Also, the probability of message v being only by node o_1 in network $\mathcal{N}$ is 0.9 and 0.81 in network $\mathcal{M}$.

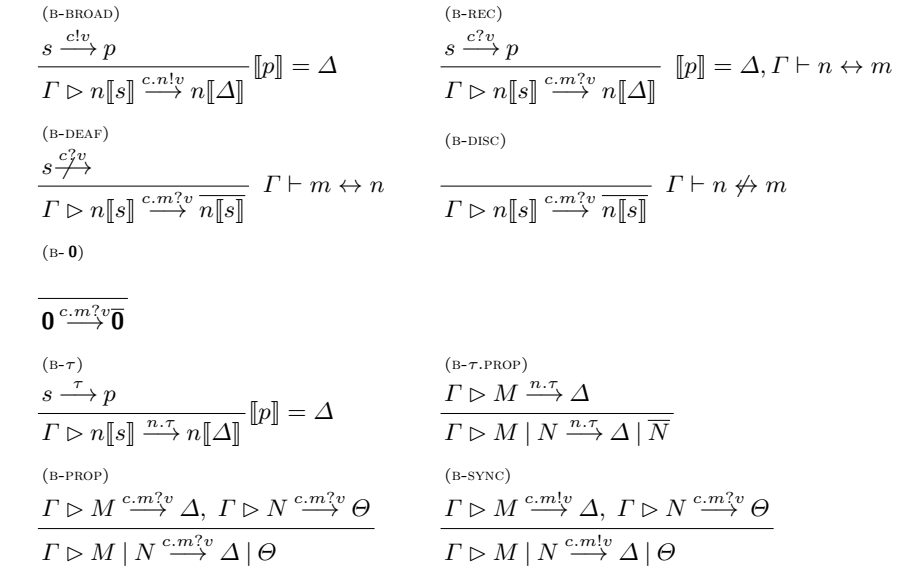


Fig. 3. Intensional semantics of networks

Rules (B-REC), (B-DEAF) and (B-DISC) express how a node n reacts when a message is broadcast by a sender node m; if the former is in the range of transmission of the sender, and it is waiting to receive a value along the same channel used by the sender to broadcast, then it will receive the message correctly. In all the other cases the behaviour of node n is not affected by the broadcast performed by m.

The rules (B-τ) and (B-τ.PROP) model internal activities performed by some node of a system term. Finally, rules (B-SYNC) and (B-PROP) describe how communication between nodes of a network is handled; these rules have been defined to model broadcast communication. See [1] for more discussion and some sanity checks on the rules. For example one can show that if $\Gamma \rhd M \xrightarrow{\mu} \Delta$ can be inferred from the rules then every N in the support of Δ has exactly the same set of node station names as M.

4 Testing Networks

As discussed in the Introduction, in order to test networks we need to be able to compose the network to be tested, say $\mathcal{M}$, with the network performing the test, say $\mathcal{N}$. A natural definition would be to define

$$(\Gamma_M \rhd M) \,|||\, (\Gamma_N \rhd N) = (\Gamma_M \cup \Gamma_N) \rhd (M \mid N) \tag{1}$$

where the combined connectivity graph $\Gamma_M \cup \Gamma_N$ is obtained set theoretically, by the point-wise union of the individual node sets and edge sets. However in

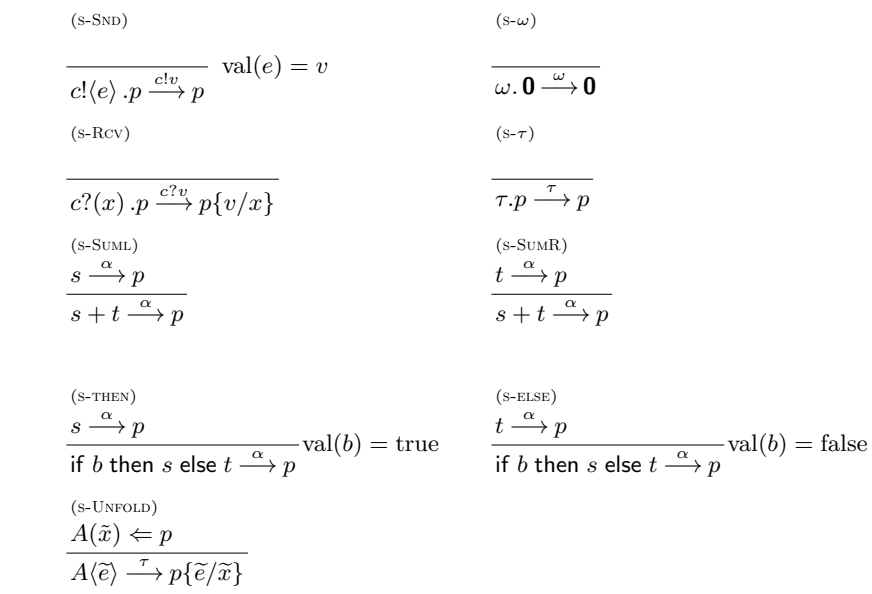


Fig. 4. Pre-semantics of states

general this will lead to ill-defined networks. Therefore we have to be satisfied by partial composition operators; moreover we should only use a composition operators which reflect in some way the practical manner in which the tester $\mathcal{N}$ can realistically interact with the testee $\mathcal{M}$.

Definition 3 (Network Extension). *The operator $\Vdash$ is the partial operator between pairs of networks defined by letting $(\Gamma_M \rhd M) \Vdash (\Gamma_N \rhd N) = (\Gamma_M \cup \Gamma_N) \rhd (M \mid N)$ if $nodes(M) \cap (\Gamma_N)_V = \emptyset$, undefined otherwise.* □

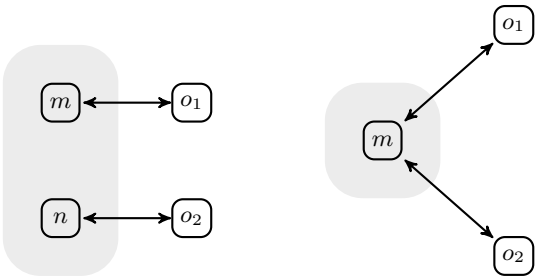
This operator is associative but in general not symmetric. In $\mathcal{M} \Vdash \mathcal{N}$ the system $\mathcal{N}$ is only allowed to place code at interface of $\mathcal{M}$. In particular it has no access to the internal nodes of $\mathcal{M}$; on the other hand $\mathcal{M}$ can place no code at any node in $\mathcal{N}$. These restrictions are natural if we view $\mathcal{N}$ as testing $\mathcal{M}$ in a black-box manner.

Proposition 1 (Interface Preservation). *If $\mathsf{Int}(\mathcal{M}) = \mathsf{Int}(\mathcal{N})$, and $\mathcal{L}$ is a network such that both $\mathcal{M} \Vdash \mathcal{L}$ and $\mathcal{N} \Vdash \mathcal{L}$ are defined then $\mathsf{Int}(\mathcal{M} \Vdash \mathcal{L}) = \mathsf{Int}(\mathcal{N} \Vdash \mathcal{L})$.* □

The operator $\Vdash$ is also a universal constructor for (well-formed) networks:

Proposition 2. *Every well-formed network $\Gamma \rhd M$ such that M is different from $\mathbf{0}$ can be written*[2] *in the form $\Gamma \rhd M = (\Gamma' \rhd M') \Vdash (\Gamma'' \rhd n[\![s]\!])$.* □

[2] modulo a simple structural equivalence



$\mathcal{M} = \Gamma_M \rhd m[\![c!\langle v\rangle \,.\, \mathbf{0}]\!] \mid n[\![c!\langle v\rangle \,.\, \mathbf{0}]\!]$ $\quad$ $\mathcal{N} = \Gamma_N \rhd m[\![c!\langle v\rangle]\!]$

Fig. 5. Broadcast vs. Multicast

We test network $\mathcal{M}$ by considering maximal computations of the composite systems $\mathcal{M} \Vdash \mathcal{T}$, where $\mathcal{T}$ is a system designed to elicit certain behaviour from $\mathcal{M}$. This composite testing harness should run in isolation from its environment, for example by ignoring possible broadcasts either $\mathcal{M}$ or $\mathcal{T}$ might receive at their interfaces. So we define a reduction relation $\twoheadrightarrow$ for networks, by letting $(\Gamma \rhd M) \twoheadrightarrow \Delta$ whenever $(\Gamma \rhd M) \xrightarrow{m.\tau} \Delta$ or $\Gamma \rhd M \xrightarrow{c.m!v} \Delta$.

We also define a success predicate $\omega(\cdot)$ for networks by letting $\omega(\mathcal{M}) = \text{true}$ whenever $\mathcal{M} = \Gamma \rhd (n[\![s]\!] \mid M)$ for some state s such that $s \xrightarrow{\omega}$. Thus we have defined a particular pLTS, as in Definition 1, with a unique transition action $\xrightarrow{\tau}$, which we take to be $\twoheadrightarrow$; such simple pLTSs we refer to as *testing structures*, TSs.

Definition 4 (Tabulating results). *The* value *of a sub-distribution in a TS is given by the function $\mathcal{V} : \mathcal{D}_{sub}(S) \to [0,1]$, defined by $\mathcal{V}(\Delta) = \sum\{\, \Delta(s) \mid \omega(s) = \mathsf{true} \,\}$. Then the set of possible results from a state s is given by $\mathcal{R}(s) = \{\, \mathcal{V}(\Delta') \mid \Delta \Longrightarrow \Delta' \,\}$. Recall from page 4 that $\overline{s} \Longrightarrow \Delta'$ represents a (probabilistic) maximal computation from s.* □

Definition 5 (Testing networks). *We write*

(i) $\mathcal{M} \sqsubseteq_{may} \mathcal{N}$ *if for every system $\mathcal{T}$ such that both $\mathcal{M} \Vdash \mathcal{T}$ and $\mathcal{N} \Vdash \mathcal{T}$ are well-defined, and every outcome $p \in \mathcal{R}(\mathcal{M} \Vdash \mathcal{T})$ there exists $p' \in \mathcal{R}(\mathcal{N} \Vdash \mathcal{T})$ such that $p \leq p'$.*

(ii) $\mathcal{M} \sqsubseteq_{must} \mathcal{N}$ *if for every $\mathcal{T}$ such that both $\mathcal{M} \Vdash \mathcal{T}$ and $\mathcal{N} \Vdash \mathcal{T}$ are well-defined and for every $p' \in \mathcal{R}(\mathcal{M} \Vdash \mathcal{T})$ there exists $p \in \mathcal{R}(\mathcal{N} \Vdash \mathcal{T})$ such that $p \leq p'$.* □

Example 2 (Broadcast vs Multicast). Consider the networks $\mathcal{M}$ and $\mathcal{N}$ in Figure 5. Intuitively in $\mathcal{N}$ the value v is (simultaneously) broadcast to both nodes o_1 and o_2 while in $\mathcal{M}$ there is a multicast. More specifically o_1 receives v from mode m while in an independent broadcast o_2 receives it from n.

This difference in behaviour can be detected by testing network

$$\mathcal{T} = \Gamma_T \rhd o_1[\![c?(x)\,.c!\langle 0\rangle \,.\, \mathbf{0}]\!] \mid o_2[\![c?(x)\,.c?(y)\,.\mathsf{if}\ y = 0\ \mathsf{then}\ \mathbf{0}\ \mathsf{else}\ \omega]\!]$$

assuming v is different than 0; here we assume Γ_T is the simple network which connects o_1 with o_2. Both $\mathcal{M} \Vdash \mathcal{T}$ and $\mathcal{N} \Vdash \mathcal{T}$ are well-formed and note that they are both non-probabilistic.

Because $\mathcal{N}$ simultaneously broadcasts to o_1 and o_2 the second value received by o_2 is always 0 and therefore the test never succeeds; $\mathcal{V}(\mathcal{N} \Vdash \mathcal{T}) = \{0\}$. On the other-hand there is a possibility for the test succeeding when applied to $\mathcal{M}$, $1 \in \mathcal{V}(\mathcal{M} \Vdash \mathcal{T})$. This is because in $\mathcal{M}$ node m might first transmit v to o_1 after which n transmits 0 to o_2; now node n might transmit the value v to o_2 and assuming it is different than 0 we reach a success state. It follows that $\mathcal{M} \not\sqsubseteq_{may} \mathcal{N}$.

One might also think it possible to use the difference between broadcast and multicast to design a test which $\mathcal{N}$ passes and $\mathcal{M}$ does not. However this is not possible, and in Example 3 we show that $\mathcal{N} \sqsubseteq_{may} \mathcal{M}$; that is multicast can be implemented by broadcast. □

Theorem 1 (Compositionality). *Suppose* $\mathsf{Int}(\mathcal{M}_1) = \mathsf{Int}(\mathcal{M}_2)$. *Then* $\mathcal{M}_1 \sqsubseteq_{may} \mathcal{M}_2$ *implies* $\mathcal{M}_1 \Vdash \mathcal{N} \sqsubseteq_{may} \mathcal{M}_1 \Vdash \mathcal{N}$, *whenever the composite networks are well-defined. The same result holds for* $\sqsubseteq_{must}$. □

The testing preorders over networks can be defined using any (partial) binary constructor $\|$ over networks, although we would only want to use constructors which are in some sense *reasonable*, which we define as follows.

Let us say that the partial constructor is a *merge* operator if whenever it is defined the result coincides with the definition in (1) above. We say it is *invariant under renaming*, if whenever $\mathcal{M} \parallel \mathcal{N}$ is well-defined then so is $\mathcal{M}\sigma \parallel \mathcal{N}$, where σ is an aribitrary renaming of nodes which leaves both $\mathsf{Int}(\mathcal{M})$ and nodes($\mathcal{N}$) unchanged, and whose range does not intersect nodes($\mathcal{N}$); intuitively this means that the well-definedness of the composite $\mathcal{M} \parallel \mathcal{N}$ is not affected by a re-organisation of the internal nodes of $\mathcal{M}$. Then we say $\|$ is *reasonable* if it is a *merge* operator, it *preserves interfaces*, and is *invariant under renaming*; note that $\Vdash$ is reasonable. Let us denote the resulting testing pre-orders by $\sqsubseteq^{alt}_{may}$, $\sqsubseteq^{alt}_{must}$ respectively.

Theorem 2. *If the constructor* $\|$ *is reasonable and symmetric then the resulting testing preorders are degenerate; that is* $\mathcal{M}_1 \sqsubseteq^{alt}_{may} \mathcal{M}_2$ *and* $\mathcal{M}_1 \sqsubseteq^{alt}_{must} \mathcal{M}_2$, *for all networks* $\mathcal{M}_1$, $\mathcal{M}_2$ *such that* $\mathsf{Int}(\mathcal{M}_1) = \mathsf{Int}(\mathcal{M}_2)$. □

5 Proof techniques for the testing preorders

Motivated by [3], our intention is to define simulations over a pLTS to provide reasonable proof techniques for inferring $\mathcal{M} \sqsubseteq_{may} \mathcal{N}$ and $\mathcal{M} \sqsubseteq_{must} \mathcal{N}$. The pLTS induced by the intensional semantics in Figure 3 is much too coarse for this purpose. Instead we need to define *extensional* actions, which capture more closely the manner in which the behaviour of wireless systems can be detected at their interfaces. The following remarks are relevant.

(i) A node m which receives a value v has no information about the name of the node, internal or external, which is responsible for the broadcast; it can only check the content of the value.

$$\text{(B-Tau)}\quad \frac{\Gamma \rhd M \xrightarrow{m.\tau} \Delta}{\Gamma \rhd M \xrightarrow{\tau} \Gamma \rhd \Delta}\ \begin{array}{l}[\![p]\!] = \Delta \\ \omega(\Gamma \rhd M) = \mathit{false}\end{array} \qquad \text{(B-In)}\quad \frac{\Gamma \rhd M \xrightarrow{c.m?v} \Delta}{\Gamma \rhd M \xrightarrow{c.m?v} \Gamma \rhd \Delta}\ \begin{array}{l}m \in \mathsf{Int}(\Gamma \rhd M) \\ \omega(\Gamma \rhd M) = \mathit{false}\end{array}$$

$$\text{(B-Shh)}\quad \frac{\Gamma \rhd M \xrightarrow{c.m!v} \Delta}{\Gamma \rhd M \xrightarrow{\tau} \Gamma \rhd \Delta}\ \begin{array}{l}\{\, m \in \mathsf{Int}(\Gamma_M \rhd M) \mid \Gamma \vdash m \leftrightarrow n \,\} = \emptyset \\ \omega(\Gamma \rhd M) = \mathit{false}\end{array}$$

$$\text{(B-Out)}\quad \frac{\Gamma \rhd M \xrightarrow{c.m!v} \Delta}{\Gamma \rhd n[\![s]\!] \xrightarrow{c!v \rhd \eta} \Gamma \rhd \Delta}\ \begin{array}{l}\eta := \{\, m \in \mathsf{Int}(\Gamma_M \rhd M) \mid \Gamma \vdash m \leftrightarrow n \,\} \\ \eta \neq \emptyset \\ \omega(\Gamma \rhd M) = \mathit{false}\end{array}$$

Fig. 6. Extensional semantics of networks

(ii) On the other hand, the set of nodes in the interface of a network $\mathcal{M}$ which are affected by a broadcast performed by a node $m \in \mathrm{nodes}(\mathcal{M})$ is relevant; these are the only nodes at the external environment which can detect the broadcast.

(iii) As a consequence, if a broadcast originated by a node in $\mathcal{M}$ does not affect any node in its interface, then this activity cannot be observed by the external environment of $\mathcal{M}$.

(iv) The effect on a network $\mathcal{M}$ by external activity can be captured adequately by broadcasts fired from nodes in the interface of $\mathcal{M}$.

(v) Since we are not interested in the behaviour of a network after it has reached a successful configuration, we require that extensional transitions can be performed only by non-successful network.

These observations motivate the definition of external actions in Figure 6. Note that these actions endow a network with the structure of a pLTS; we say that a network is *finitary* if so is the pLTS it generates via the transitions defined in Figure 6. Henceforth in this Section we always assume that a network is finitary. The extension to weak actions is also non-standard:

Definition 6 (Weak extensional action). *Let Δ, Θ be network sub-distributions over* Nets*. We say that*

(i) $\Delta \stackrel{\tau}{\Longrightarrow} \Theta$ *if* $\Delta \Longrightarrow \Theta$ *in the pLTS induced by the extensional transitions of Figure 6.*

(ii) $\Delta \stackrel{c.m?v}{\Longrightarrow} \Theta$ *if* $\mathcal{M} \stackrel{\tau}{\Longrightarrow} \stackrel{c.m?v}{\longrightarrow} \stackrel{\tau}{\Longrightarrow} \Theta$

(iii) $\Delta \stackrel{c!v \rhd \eta}{\Longrightarrow} \Theta$ *if either* $\Delta \stackrel{\tau}{\Longrightarrow} \stackrel{c!v \rhd \eta}{\longrightarrow} \stackrel{\tau}{\Longrightarrow} \Theta$ *or* $\Delta \stackrel{c!v \rhd \eta_1}{\Longrightarrow} \stackrel{c!v \rhd \eta_2}{\Longrightarrow} \Theta$*, where η_1, η_2 are two non-empty sets of nodes which constitute a partition of η.* □

The non-standard *(iii)* is motivated in Example 3.

These weak actions endow the set of networks Nets with the structure of another pLTS, called the *extensional pLTS* and denoted by $\mathsf{pLTS}_{\mathsf{Nets}}$. It is in this pLTS that we give our definitions of simulations.

The first one is based on the simulation preorder from [3]; for reasons best explained there it is defined as a relation from states to distributions, rather than the more standard states to states.

Definition 7 (Simulation preorder). In $\mathsf{pLTS}_{\mathsf{Nets}}$ we let $\lhd_{sim}$ denote the largest relation in $\mathsf{Nets} \times \mathcal{D}(\mathsf{Nets})$ such that if $s \lhd_{sim} \Theta$ then:

- if $\omega(s) =$ true, then $\Theta \stackrel{\tau}{\Longrightarrow} \Theta'$ such that for every $t \in \lceil \Theta' \rceil, \omega(t) =$ true
- otherwise, whenever $\overline{s} \stackrel{\mu}{\Longrightarrow} \Delta'$, for $\mu \in \mathsf{Act}_\tau$, then there is a $\Theta' \in \mathcal{D}(S)$ with $\Theta \stackrel{\mu}{\Longrightarrow} \Theta'$ and $\Delta' \; \overline{\lhd_{sim}} \; \Theta'$. □

Our second proof technique is a variation on the *failure simulation preorder* of [3]. Unlike in that more general framework we have no need of acceptance sets. Instead it is sufficient to consider the ability of systems to *deadlock*. See [4] for details. We say that a network $\mathcal{M}$ is *deadlocked*, denoted $\mathcal{M} \not\rightarrow$ whenever $\omega(\mathcal{M}) =$ false and $\mathcal{M} \stackrel{\tau}{\not\rightarrow}, \mathcal{M} \stackrel{c!v \rhd \eta}{\not\rightarrow}$ for any c, v, η. A sub-distribution Δ over $\mathcal{D}_{sub}(\mathsf{Nets})$ is *deadlocked* if any network in its support is deadlocked.

For reasons explained in [3] it is more straightforward to express this form of simulation as a relation from sub-distributions to sub-distributions.

Definition 8 (Deadlock Simulations). *In* $\mathsf{pLTS}_{\mathsf{Nets}}$ *we let* $\sqsupseteq_{DS}$ *denote the largest relation in* $\mathcal{D}_{sub}(\mathsf{Nets}) \times \mathcal{D}_{sub}(\mathsf{Nets})$ *such that if* $\Delta \sqsupseteq_{DS} \Theta$ *then:*

- *whenever* $\Delta \stackrel{\mu}{\Longrightarrow} \sum_{i \in I}(p_i \cdot \Delta'_i)$*, where* I *is an index set such that* $\sum_{i \in I} p_i \leq 1$*, then there are* $\Theta'_i \in \mathcal{D}_{sub}(\mathsf{Nets})$ *such that* $\Theta \stackrel{\mu}{\longrightarrow} \sum_{i \in I}(p_i \cdot \Theta'_i)$ *and, for any* $i \in I$*,* $\Delta'_i \sqsupseteq_{DS} \Theta'_i$
- *whenever* $\Delta \Longrightarrow \not\rightarrow$ *then* $\Theta \Longrightarrow \not\rightarrow$. □

Theorem 3 (Proof methods for the testing preorders). *Let* $\mathcal{M}, \mathcal{N}$ *be two networks such that* $\mathsf{Int}(\mathcal{M}) = \mathsf{Int}(\mathcal{N})$*. Then*

- *if* $\mathcal{M} \lhd_{sim} \overline{\mathcal{N}}$ *then* $\mathcal{M} \sqsubseteq_{may} \mathcal{N}$
- *if* $\overline{\mathcal{M}} \sqsupseteq_{DS} \overline{\mathcal{N}}$ *then* $\mathcal{M} \sqsubseteq_{must} \mathcal{N}$. □

Thus in order to relate two wireless systems it is sufficient to exhibit an appropriate simulation relation.

Example 3. Consider again the networks $\mathcal{M}$ and $\mathcal{N}$ in Figure 5. It is easy to show that both of them can perform the weak extensional action $\stackrel{c!v \rhd \{o_1, o_2\}}{\Longrightarrow}$. However, the inference of this action is different for the individual networks; while in network $\mathcal{N}$ it is implied by the execution of a single broadcast action, detected by both nodes o_1 and o_2 simultaneously, in $\mathcal{M}$ this is implied by a sequence of weak extensional actions $\mathcal{M} \stackrel{c!v \rhd \{o_1\}}{\Longrightarrow} \stackrel{c!v \rhd \{o_2\}}{\Longrightarrow}$.

It is therefore possible to exhibit a simulation between $\mathcal{N}$ and $\mathcal{M}$, thus showing that $\mathcal{N} \lhd_{sim} \overline{\mathcal{M}}$; By Theorem 3 it follows that $\mathcal{N} \sqsubseteq_{may} \mathcal{M}$. Similarly, it is possible to prove that $\overline{\mathcal{M}} \sqsupseteq_{DS} \overline{\mathcal{N}}$, and therefore $\mathcal{M} \sqsubseteq_{must} \mathcal{N}$.

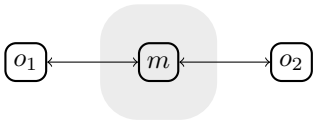
Now suppose that we employed a standard definition of weak extensional actions, and that the simulation preorder had been defined according to this

Fig. 7. The network $\mathcal{N}_c$

notion. In this case it would not be possible to exhibit a simulation between $\mathcal{N}$ and $\mathcal{M}$, and thus it would not be possible to prove that $\mathcal{N} \sqsubseteq_{may} \mathcal{M}$. The same applies for the $\sqsubseteq_{must}$ testing preorder and deadlock simulations. □

Although simulations provide a sound proof technique for $\sqsubseteq_{may}$ and $\sqsubseteq_{must}$, in general they are not complete. For example one can show, referring to Example 1, that $\mathcal{M} \sqsubseteq_{may} \mathcal{N}$ but $\mathcal{M} \not\lhd_{sim} \overline{\mathcal{N}}$. It remains to be seen if our notions of simulation can be further adapted so as to provide complete proof methodologies.

6 An application: probabilistic routing



Sequential routing means that a network can only route one message per time; if a message is received by the network while another one is being routed, then it is ignored. A simple specification (or model) network $\mathcal{M}$ that can be used for such routing in networks is depicted on the left, where $P \Leftarrow c?(x).c!\langle x\rangle.P$. It is trivial to see that, whenever node m in $\mathcal{M}$ receives a message from o_1, it will be forwarded to nodes o_1 and o_2; that is, the message has been routed from the external node o_1 to the external node o_2. The model also routes messages from o_2 to o_1 in a symmetric fashion. We provide a possible implementation of this specification as a probabilistic (and nondeterministic) network $\mathcal{N}$ such that $\mathcal{M} \sqsubseteq_{may} \mathcal{N}$; this means that $\mathcal{N}$ will include all the possible behaviour of $\mathcal{M}$, such as the sequential routing between the interface nodes o_1, o_2, but may also have additional behaviour.

In fact we design an entire class of networks $\mathcal{N}$ with this property. Each will have the structure $\mathcal{N} = \mathcal{N}_c \Vdash \mathcal{C}$, where $\mathcal{N}_c$ is the network depicted in Figure 7. This acts as a connector between the interface nodes o_1, o_2 and the internal router which it accesses via the nodes n_1, n_2; here $P_m \Leftarrow c?(x).c!\langle x\rangle.P_m + c?(x).P_m$.

The network $\mathcal{C} = \Gamma_C \rhd C$, on the other hand, is defined parametrically. It can be any network that satisfies the following requirements:

1. $n_1, n_2 \in \text{nodes}(\Gamma_C \rhd C)$. For the sake of simplicity, we also assume that $\text{nodes}(\Gamma_C \rhd C) = (\Gamma_C)_V = \{n_1, \cdots, n_k\}$ for some $k > 2$.
2. The connectivity graph Γ_C contains a single connected component.

3. Every node n_i, $i = 1, \cdots, k$ is associated with a channel c_i and a probability distribution $\Lambda_i : \{1, \cdots, k\} \to [0, 1]$. The latter are defined so that $\lceil \Lambda_i \rceil = \{j \mid \Gamma_C \vdash n_i \leftrightarrow n_j\}$, for any $i = 1, \cdots, k$. That is, if node n_i is connected to node n_j, then $\Lambda_i(j) > 0$.
4. $C = \prod_{i \in I} n_i[\![P_i]\!]$, where

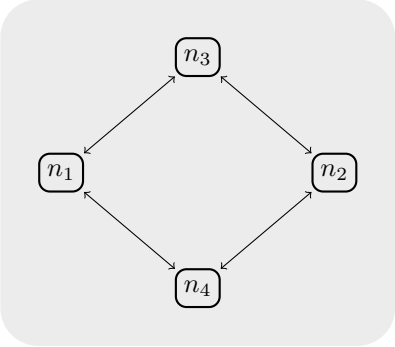
$$P_i = c?(x) \,.\, \left[\bigoplus_{j=1}^{k} \Lambda_i(j) \cdot c_j!\langle x \rangle \,.P_i \right] + c?(x) \,.P_i +$$

$$+ \, c_i?(x) \,.\, \left[\bigoplus_{j=1}^{k} \Lambda_i(j) \cdot c_j!\langle x \rangle \,.P_i \right] + c_i?(x) \,.c!\langle x \rangle \,.P_i, \quad i = 1, 2$$

$$P_i = c_i?(x) \,.(\bigoplus_{j=1}^{k} \Lambda_i(j) \cdot c_j!\langle x \rangle \,.P_i), \qquad i > 2$$

Here the derived construct $\bigoplus_{i \in I} p_i \cdot P_i$ is interpreted in the obvious manner as a probability distribution.

Let us explain, informally, one of the possible behaviours of a typical network $\mathcal{N}$. The connectivity of one possible $\mathcal{C}$, with only two extra nodes n_3, n_4, is given on the left. Upon receiving a message along channel c from the external node o_1, node m_1 will forward it to the node n_1; here note that the external node o_1 also detects the broadcast. Node n_1 forwards the message again to one of its neighbours n_j with a strictly positive probability. Here n_j is selected as the next hop in the routing path by forwarding the message along channel c_j. In fact, node c_j is the only one which can detect messages broadcast along such a channel.



This procedure is iterated until the message v is forwarded to node n_2, at least with some probability, which in turn will forward it to node m_2; a final broadcast from the latter node will cause the message to be detected by the external node o_2. Since the connectivity graph of $\mathcal{C}$ has a single connected component, it is possible to show that, upon being received by node n_1, a message v eventually is delivered to node n_2 almost surely, i.e. with probability 1.

This informal line of reasoning can be used to provide a simulation between the networks $\mathcal{M}$ and $\mathcal{N}$; we can construct a simulation in the extensional $\mathsf{pLTS}_{\mathsf{Nets}}$ containing the pair $(\mathcal{M}, \overline{\mathcal{N}})$, for any $\mathcal{N}$ whose internal component $\mathcal{C}$ satisfies the four constraints given above; thus for any such $\mathcal{N}_c$ we have $\mathcal{M} \sqsubseteq_{may} \mathcal{N}$. The details may be found in [1]

We finish with three remarks about this example. First note that it is necessary to employ our non-standard definition of weak output actions, for a broad-

cast of network $\mathcal{M}$ to the nodes o_1, o_2 can only be simulated by $\mathcal{N}$ via a sequence of two broadcasts. The first, fired by node m_1, can be detected only by o_1; the second one, fired by node m_2, can be detected only by node o_2, and this only in a probabilistic limit, for which we require the infinitary version of probabilistic weak actions. Secondly note that the network $\mathcal{N}$ implements sequential routing because the nodes m_1, m_2 in $\mathcal{N}$ can non-deterministically decide to ignore a broadcast performed by o_1, o_2 respectively. Finally note that this example emphasises the fact that our formalism can in principle be employed to examine practical routing algorithms. Routing protocols in which the next-hop in a routing path is determined via a probability distribution, as in our example, are of practical significance; see for example, [2].

7 Conclusions

To the best of our knowledge, we believe that our work is the first to apply testing theories to wireless systems, and in particular probabilistic wireless systems. One major strand of research into calculi based on broadcast communications starts with CBS from [15]; here behaviour is defined in terms of various forms of bisimulations. Later developments include local broadcast communication in the Extended CBS of [13] and the use of connectivity graphs in the CBS# of [14].

In [11] a different attempt to formalize wireless networks is made. The authors develop a calculus CWS, where the concepts of node names and location are differentiated; thus, a process is associated both with a node name and a location. Also, every process has a positive real value associated to it, denoting the radius of transmission. The network topology is determined by a metric on locations and a transmision radius. It is worth mentioning that in this calculus the communication between nodes consists of two phases, one to start it and one to end it. The authors also model the possibility of a message whose transmission has started to be corrupted by another transmission, thus modeling collisions.

In [10], a descrete timed calculus for wireless systems (TCWS) is presented; in this case, the authors address the problem of representing collisions in wireless networks, suggesting that formal tools for dealing with interferences in wireless networks can aid in the development of MAC level protocols. The topology of the wireless networks here is described by associating every node a semantic tag representing its set of neighbours. The authors propose a compositional theory for wireless networks based on the notion of reduction barbed congruence; further, they develop a sound proof methodology based on bisimulations over an extensional lts. It is of considerable interest that, despite their targeting at low-level collision prone behaviour, the set of extensional actions they propose (and the activities that can be detected by the external environment) is very similar to those we have suggested.

References

1. Andrea Cerone and Matthew Hennessy. A simple probabilistic broadcast language. *Technical Report, Trinity College Dublin*, (CS-TR-2012-02), 2012. Available at `http://www.scss.tcd.ie/~ceronea/works/simpleProbabilisticNetworks.pdf`.
2. Eoin Curran and Jim Dowling. Sample: Statistical network link modelling in an on-demand probabilistic routing protocol for ad hoc networks. In *WONS*, pages 200–205. IEEE Computer Society, 2005.
3. Yuxin Deng, Rob van Glabbeek, Matthew Hennessy, and Carroll Morgan. Testing finitary probabilistic processes. In *Proceedings of the 20th International Conference on Concurrency Theory*, volume 5710 of *Lecture Notes in Computer Science*, pages 274–288. Springer, 2009. Full-version available from http://www.scss.tcd.ie/Matthew.Hennessy/onlinepubs.html.
4. Cristian Ene and Traian Muntean. Testing theories for broadcasting processes. *Sci. Ann. Cuza Univ*, 11:214–230, 2002.
5. Fatemeh Ghassemi, Wan Fokkink, and Ali Movaghar. Verification of mobile ad hoc networks: An algebraic approach. *Theor. Comput. Sci*, 412(28):3262–3282, 2011.
6. Jens Chr. Godskesen. Observables for mobile and wireless broadcasting systems. In Dave Clarke and Gul A. Agha, editors, *COORDINATION*, volume 6116 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 2010.
7. Raja Jurdak, Cristina Videira Lopes, and Pierre Baldi. A survey, classification and comparative analysis of medium access control protocols for ad hoc networks. *IEEE Communications Surveys and Tutorials*, 6(1-4):2–16, 2004.
8. Ivan Lanese and Davide Sangiorgi. An operational semantics for a calculus for wireless systems. *Theor. Comput. Sci*, 411(19):1928–1948, 2010.
9. Ruggero Lanotte and Massimo Merro. Semantic analysis of gossip protocols for wireless sensor networks. In Joost-Pieter Katoen and Barbara König, editors, *CONCUR*, volume 6901 of *Lecture Notes in Computer Science*, pages 156–170. Springer, 2011.
10. Massimo Merro and Eleonora Sibilio. A timed calculus for wireless systems. In Farhad Arbab and Marjan Sirjani, editors, *FSEN*, volume 5961 of *Lecture Notes in Computer Science*, pages 228–243. Springer, 2009.
11. Nicola Mezzetti and Davide Sangiorgi. Towards a calculus for wireless systems. *Electr. Notes Theor. Comput. Sci*, 158:331–353, 2006.
12. R. Milner. A calculus of communicating systems. *LNCS*, 92, 1980.
13. Sebastian Nanz and Chris Hankin. Static analysis of routing protocols for ad-hoc networks, March 25 2004.
14. Sebastian Nanz and Chris Hankin. A framework for security analysis of mobile wireless networks. *TCS: Theoretical Computer Science*, 367, 2006.
15. K. V. S. Prasad. A calculus of broadcasting systems. *Science of Computer Programming*, 25(2–3):285–327, December 1995.
16. Roberto Segala and Nancy Lynch. Probabilistic simulations for probabilistic processes. *Nordic J. of Computing*, 2:250–273, June 1995.
17. Andrew S. Tanenbaum. *Computer networks*. PTR Prentice-Hall.

Appendix: Some definitions

Definition 9 (Hyper-derivations). *In a pLTS a hyper-derivation consists of a collection of sub-distributions $\Delta, \Delta_k^{\rightarrow}, \Delta_k^{\times}$, for $k \geq 0$, with the following properties:*

$$\begin{array}{lcl} \Delta & = & \Delta_0^{\rightarrow} + \Delta_0^{\times} \\ \Delta_0^{\rightarrow} & \xrightarrow{\tau} & \Delta_1^{\rightarrow} + \Delta_1^{\times} \\ & \vdots & \\ \Delta_k^{\rightarrow} & \xrightarrow{\tau} & \Delta_{k+1}^{\rightarrow} + \Delta_{k+1}^{\times} \\ & \vdots & \end{array}$$

If $\omega(s) =$ false for each $s \in \lceil \Delta_k^{\rightarrow} \rceil$ and $k \geq 0$ we call $\Delta' = \sum_{k=0}^{\infty} \Delta_k^{\times}$ a hyper-derivative *of Δ, and write $\Delta \Longrightarrow \Delta'$.* □

In maximal computations, we require the computation to proceed as long as some internal activity can be performed. To this end, we say that $\Delta \Longrightarrow\!\!\!\!\succ \Delta'$ if

- $\Delta \Longrightarrow \Delta'$,
- for every $s \in \lceil \Delta_k^{\times} \rceil$, $s \xrightarrow{\tau}$ implies $\omega(s) = \text{true}$. □

This is a mild generalisation of the notion of *extreme derivative* from [3]. Note that the last constraint models exactly the requirement of performing some internal activity whenever it is possible; In other words extreme derivatives correspond to a probabilistic version of maximal computations.