Modelling MAC-layer communications in wireless systems (Extended abstract)

Andrea Cerone[1], Matthew Hennessy[1]*, and Massimo Merro[2]**

[1] Department of Computer Science and Statistics, Trinity College Dublin, Ireland
{acerone, Matthew.Hennessy}@scss.tcd.ie
[2] Dipartimento di Informatica,
Università degli Studi di Verona, Italy
massimo.merro@univr.it

Abstract We present a timed broadcast process calculus for wireless networks at the MAC-sublayer where time-dependent communications are exposed to collisions. We define a reduction semantics for our calculus which leads to a contextual equivalence for comparing the external behaviour of wireless networks. Further, we construct an extensional LTS (labelled transition system) which models the activities of stations that can be directly observed by the external environment. Standard bisimulations in this novel LTS provide a sound proof method for proving that two systems are contextually equivalent. In addition, the main contribution of the paper is that our proof technique is also complete for a large class of systems.

1 Introduction

Wireless networks are becoming increasingly pervasive with applications across many domains, [19,1]. They are also becoming increasingly complex, with their behaviour depending on ever more sophisticated protocols. There are different levels of abstraction at which these can be defined and implemented, from the very basic level in which the communication primitives consist of sending and receiving electromagnetic signals, to the higher level where the basic primitives allow the set up of connections and exchange of data between two nodes in a wireless system [23].

Assuring the correctness of the behaviour of a wireless network has always been difficult. Several approaches have been proposed to address this issue for networks described at a high level [16,13,6,5,22,11,2,3]; these typically allow the formal description of protocols at the *network layer* of the *TCP/IP* reference model [23]. However there are few frameworks in the literature which consider networks described at the MAC-Sublayer of the *TCP/IP* reference model [12,14]. This is the topic of the current paper. We propose a process calculus for describing and verifying wireless networks at the *MAC-Sublayer* of the *TCP/IP* reference model.

This calculus, called the Calculus of Collision-prone Communicating Processes (CCCP), has been largely inspired by TCWS [14]; in particular CCCP inherits its communication features but simplifies considerably the syntax, the reduction semantics, the

* Supported by SFI project SFI 06 IN.1 1898.

** Author partially supported by the PRIN 2010-2011 national project ”Security Horizons”

notion of observation, and as we will see the behavioural theory. In CCCP a wireless system is considered to be a collection of wireless stations which transmit and receive messages. The transmission of messages is broadcast, and it is time-consuming; the transmission of a message v can require several time slots (or instants). In addition, wireless stations in our calculus are sensitive to collisions; if two different stations are transmitting a value over a channel c at the same time slot a collision occurs, and the content of the messages originally being transmitted is lost.

More specifically, in CCCP a state of a wireless network (or simply network, or system) will be described by a *configuration* of the form $\Gamma \triangleright W$ where W describes the code running at individual wireless stations and Γ represents the communication state of channels. At any given point of time there will be *exposed* communication channels, that is channels containing messages (or values) in transmission; this information will be recorded in Γ.

Such systems evolve by the broadcast of messages between stations, the passage of time, or some other internal activity, such as the occurrence of collisions and their consequences. One of the topics of the paper is to capture formally these complex evolutions, by defining a *reduction semantics*, whose judgments take the form $\Gamma_1 \triangleright W_1 \rightarrowtriangle \Gamma_2 \triangleright W_2$. The reduction semantics satisfies some desirable properties such as *time determinism*, *patience* and *maximal progress* [17,9,25].

However the main aim of the paper is to develop a behavioural theory of wireless networks. To this end we need a formal notion of when two such systems are indistinguishable from the point of view of users. Having a reduction semantics it is now straightforward to adapt a standard notion of *contextual equivalence*: $\Gamma_1 \triangleright W_1 \simeq \Gamma_2 \triangleright W_2$. Intuitively this means that either system, $\Gamma_1 \triangleright W_1$ or $\Gamma_2 \triangleright W_2$, can be replaced by the other in a larger system without changing the observable behaviour of the overall system. Formally we use the approach of [10], often called *reduction barbed congruence*; the only parameter in the definition is the choice of primitive observation or *barb*. Our choice is natural for wireless systems: the ability to transmit on an idle channel, that is a channel with no active transmissions.

As explained in papers such as [20,7], contextual equivalences are determined by so-called *extensional actions*, that is the set of minimal observable interactions which a system can have with its external environment. For CCCP determining these actions is non-trivial. Although values can be transmitted and received on channels, the presence of collisions means that these are not necessarily observable. In fact the important point is not the transmission of a value, but its successful delivery. Also, although the basic notion of observation on systems does not involve the recording of the passage of time, this has to be taken into account extensionally in order to gain a proper extensional account of systems.

The extensional semantics determines an LTS (labelled transition system) over configurations, which in turn gives rise to the standard notion of (weak) bisimulation equivalence between configurations. This gives a powerful co-inductive proof technique: to show that two systems are behaviourally equivalent it is sufficient to exhibit a witness bisimulation which contains them.

One result of this paper is that weak bisimulation in the extensional LTS is sound with respect to the touchstone contextual equivalence: if two systems are related by some bisimulation in the extensional LTS then they are contextually equivalent. However, the main contribution is that completeness holds for a large class of networks, called *well-formed*. If two such networks are contextually equivalent then there is some bisimulation, based on our novel extensional actions, which contains them. In [14], a sound but not complete bisimulation based proof method is developed for (a different form of) reduction barbed congruence. Here, by simplifying the calculus and isolating novel extensional actions we obtain both soundness and completeness.

The rest of the paper is organised as follows: in Section 2 we define the syntax which we will use for modelling wireless networks. The reduction semantics is given in Section 3 from which we develop in the same section our notion of reduction barbed congruence. In Section 4 we define the extensional semantics of networks, and the (weak) bisimulation equivalence it induces. In Section 5 we state the main results of the paper, namely that bisimulation is sound with respect to barbed congruence and, for a large class of systems, it is also complete. Detailed proofs of the results can be found in the associated technical report [4]. The paper comes with an appendix showing an initial case study of our proof technique. Two particular instances of networks are compared; the first forwards two messages to the external environment using a TDMA modulation technique, the second performs the same task by routing the messages along different stations.

2 The calculus

Formally we assume a set of channels **Ch**, ranged over by $c, d, \cdots$, and a set of values **Val**, which contains a set of data-variables, ranged over by $x, y, \cdots$ and a special value err; this value will be used to denote faulty transmissions. The set of *closed values*, that is those not containing occurrences of variables, are ranged over by $v, w, \cdots$. We also assume that every closed value $v \in \mathbf{Val}$ has an associated strictly positive integer δ_v, which denotes the number of time slots needed by a wireless station to transmit v.

A channel environment is a mapping $\Gamma : \mathbf{Ch} \to \mathbb{N} \times \mathbf{Val}$. In a configuration $\Gamma \triangleright W$ where $\Gamma(c) = (n, v)$ for some channel c, a wireless station is currently transmitting the value v for the next n time slots. We will use some suggestive notation for channel environments: $\Gamma \vdash_t c : n$ in place of $\Gamma(c) = (n, w)$ for some w, $\Gamma \vdash_v c : w$ in place of $\Gamma(c) = (n, w)$ for some n. If $\Gamma \vdash_t c : 0$ we say that channel c is idle in Γ, and we denote it with $\Gamma \vdash c : \mathbf{idle}$. Otherwise we say that c is exposed in Γ, denoted by $\Gamma \vdash c : \mathbf{exp}$. The channel environment Γ such that $\Gamma \vdash c : \mathbf{idle}$ for every channel c is said to be *stable*.

The syntax for system terms W is given in Table 1, where P ranges over code for programming individual stations, which is also explained in Table 1. A system term W is a collection of individual threads running in parallel, with possibly some channels restricted. Each thread may be either an inactive piece of code P or an active code of the form $c[x].P$. This latter term represents a wireless station which is receiving a value from the channel c; when the value is eventually received the variable x will be replaced with the received value in the code P. The restriction operator $\nu c : (n, v).W$ is non-standard, for a restricted channel has a positive integer and a closed value associated with it; roughly speaking, the term $\nu c : (n, v).W$ corresponds to the term W where

Table 1 CCCP: Syntax

W	$::=$	P	station code
	$\mid$	$c[x].P$	active receiver
	$\mid$	$W_1 \mid W_2$	parallel composition
	$\mid$	$\nu c{:}(n, v).W$	channel restriction
P, Q	$::=$	$c\,!\langle u\rangle.P$	broadcast
	$\mid$	$\lfloor c?(x).P\rfloor Q$	receiver with timeout
	$\mid$	$\sigma.P$	delay
	$\mid$	$\tau.P$	internal activity
	$\mid$	$P + Q$	choice
	$\mid$	$[b]P, Q$	matching
	$\mid$	X	process variable
	$\mid$	nil	termination
	$\mid$	$\mathsf{fix}\,X.P$	recursion

Channel Environment: $\Gamma : \mathbf{Ch} \to \mathbb{N} \times \mathbf{Val}$

channel c is local to W, and the transmission of value v over channel c will take place for the next n slots of time.

The syntax for station code is based on standard process calculus constructs. The main constructs are time-dependent reception from a channel $\lfloor c?(x).P\rfloor Q$, explicit time delay $\sigma.P$, and broadcast along a channel $c\,!\langle u\rangle.P$. Here u denotes either a data-variable or closed value $v \in \mathbf{Val}$. Of the remaining standard constructs the most notable is matching, $[b]P, Q$ which branches to P or Q, depending on the value of the Boolean expression b. We leave the language of Boolean expressions unspecified, other than saying that it should contain equality tests for values, $u_1 = u_2$. More importantly, it should also contain the expression $\exp(c)$ for checking if in the current configuration the channel c is exposed, that is it is being used for transmission.

In the construct $\mathsf{fix}\,X.P$ occurrences of the recursion variable X in P are bound; similarly in the terms $\lfloor c?(x).P\rfloor Q$ and $c[x].P$ the data-variable x is bound in P. This gives rise to the standard notions of free and bound variables, α-conversion and capture-avoiding substitution; we assume that all occurrences of variables in system terms are bound and we identify systems up to α-conversion. Moreover we assume that all occurrences of recursion variables are *guarded*; they must occur within either a broadcast, input or time delay prefix, or within an execution branch of a matching construct. We will also omit trailing occurrences of nil, and write $\lfloor c?(x).P\rfloor$ in place of $\lfloor c?(x).P\rfloor\mathsf{nil}$.

Our notion of wireless networks is captured by pairs of the form $\Gamma \triangleright W$, which represent the system term W running in the channel environment Γ. Such pairs are called configurations, and are ranged over by the metavariable $\mathcal{C}$.

Table 2 Intensional semantics: transmission

$$\text{(Snd)}\ \frac{}{\Gamma \triangleright c\,!\langle v\rangle.P \xrightarrow{c!v} \sigma^{\delta_v}.P} \qquad \text{(Rcv)}\ \frac{\Gamma \vdash c : \textbf{idle}}{\Gamma \triangleright \lfloor c?(x).P \rfloor Q \xrightarrow{c?v} c[x].P}$$

$$\text{(RcvIgn)}\ \frac{\neg \mathsf{rcv}(W, c)}{\Gamma \triangleright W \xrightarrow{c?v} W} \qquad \text{(Sync)}\ \frac{\Gamma \triangleright W_1 \xrightarrow{c!v} W_1' \quad \Gamma \triangleright W_2 \xrightarrow{c?v} W_2'}{\Gamma \triangleright W_1 \mid W_2 \xrightarrow{c!v} W_1' \mid W_2'}$$

$$\text{(RcvPar)}\ \frac{\Gamma \triangleright W_1 \xrightarrow{c?v} W_1' \quad \Gamma \triangleright W_2 \xrightarrow{c?v} W_2'}{\Gamma \triangleright W_1 \mid W_2 \xrightarrow{c?v} W_1' \mid W_2'}$$

3 Reduction semantics and contextual equivalence

The reduction semantics is defined incrementally. We first define the evolution of system terms with respect to a channel environment Γ via a set of SOS rules whose judgments take the form $\Gamma \triangleright W_1 \xrightarrow{\lambda} W_2$. Here λ can take the form $c!v$ denoting a broadcast of value v along channel c, $c?v$ denoting an input of value v being broadcast along channel c, τ denoting an internal activity, or σ, denoting the passage of time. However these actions will also have an effect on the channel environment, which we first describe, using a functional $\text{upd}_\lambda(\cdot) : \textbf{Env} \rightarrow \textbf{Env}$, where $\textbf{Env}$ is the set of channel environments.

The channel environment $\text{upd}_\lambda(\Gamma)$ describes the update of the channel environment Γ when the action λ is performed, is defined as follows: for $\lambda = \sigma$ we let

$$\text{upd}_\sigma(\Gamma) \vdash_t c : (n-1) \text{ whenever } \Gamma \vdash_t c : n, \qquad \text{upd}_\sigma(\Gamma) \vdash_v c : w \text{ whenever } \Gamma \vdash_v c : w.$$

For $\lambda = c!v$ we let $\text{upd}_{c!v}(\Gamma)$ be the channel environment such that

$$\text{upd}_{c!v}(\Gamma) \vdash_t c : \begin{cases} \delta_v & \text{if } \Gamma \vdash c : \textbf{idle} \\ \max(\delta_v, k) & \text{if } \Gamma \vdash c : \textbf{exp} \end{cases} \quad \text{upd}_{c!v}(\Gamma) \vdash_v c : \begin{cases} v & \text{if } \Gamma \vdash c : \textbf{idle} \\ \mathsf{err} & \text{if } \Gamma \vdash c : \textbf{exp} \end{cases}$$

where $\Gamma \vdash_t c : k$. Finally, we let $\text{upd}_{c?v}(\Gamma) = \text{upd}_{c!v}(\Gamma)$ and $\text{upd}_\tau(\Gamma) = \Gamma$.

Let us describe the intuitive meaning of this definition. When time passes, the time of exposure of each channel decreases by one time unit[3]. The predicates $\text{upd}_{c!v}(\Gamma)$ and $\text{upd}_{c?v}(\Gamma)$ model how collisions are handled in our calculus. When a station begins broadcasting a value v over a channel c this channel becomes exposed for the amount of time required to transmit v, that is δ_v. If the channel is not free a collision happens. As a consequence, the value that will be received by a receiving station, when all transmissions over channel c terminate, is the error value err, and the exposure time is adjusted accordingly.

For the sake of clarity, the inference rules for the evolution of system terms, $\Gamma \triangleright W_1 \xrightarrow{\lambda} W_2$, are split in four tables, each one focusing on a particular form of activity.

[3] For convenience we assume $0 - 1$ to be 0.

Table 2 contains the rules governing transmission. Rule (Snd) models a non-blocking broadcast of message v along channel c. A transmission can fire at any time, independently on the state of the network; the notation σ^{δ_v} represents the time delay operator σ iterated δ_v times. So when the process $c\,!\langle v\rangle.P$ broadcasts it has to wait δ_v time units before the residual P can continue. On the other hand, reception of a message by a time-guarded listener $\lfloor c?(x).P\rfloor Q$ depends on the state of the channel environment. If the channel c is free then rule (Rcv) indicates that reception can start and the listener evolves into the active receiver $c[x].P$.

The rule (RcvIgn) says that if a system can not receive on the channel c then any transmission along it is ignored. Intuitively, the predicate $\mathsf{rcv}(W, c)$ means that W contains among its parallel components at least one non-guarded receiver of the form $\lfloor c?(x).P\rfloor Q$ which is actively awaiting a message. Formally, the predicate $\mathsf{rcv}(W, c)$ is the least predicate such that $\mathsf{rcv}(\lfloor c?(x).P\rfloor Q, c) =$ true and which satisfies the equations $\mathsf{rcv}(P + Q, c) = \mathsf{rcv}(P, c) \vee \mathsf{rcv}(Q, c)$, $\mathsf{rcv}(W_1 \mid W_2, c) = \mathsf{rcv}(W_1, c) \vee \mathsf{rcv}(W_2, c)$ and $\mathsf{rcv}(\nu d.W, c) = \mathsf{rcv}(W, c)$ if $d \neq c$. The remaining two rules in Table 2 (Sync) and (RcvPar) serve to synchronise parallel stations on the same transmission [8,17,18].

Example 1 (Transmission). Let $C_0 = \Gamma_0 \triangleright W_0$, where $\Gamma_0 \vdash c,\ d$: **idle** and $W_0 = c!\langle v_0\rangle \mid \lfloor d?(x).\mathsf{nil}\rfloor(\lfloor c?(x).Q\rfloor) \mid \lfloor c?(x).P\rfloor$ where $\delta_{v_0} = 2$.

Using rule (Snd) we can infer $\Gamma_0 \triangleright c!\langle v_0\rangle \xrightarrow{c!v_0} \sigma^2$; this station starts transmitting the value v_0 along channel c. Rule (RcvIgn) can be used to derive the transition $\Gamma_0 \triangleright \lfloor d?(x).\mathsf{nil}\rfloor(\lfloor c?(x).Q\rfloor) \xrightarrow{c?v_0} \lfloor d?(x).\mathsf{nil}\rfloor(\lfloor c?(x).Q\rfloor)$, in which the broadcast of value v_0 along channel c is ignored. On the other hand, Rule (RcvIgn) cannot be applied to the configuration $\Gamma_0 \triangleright \lfloor c?(x).P\rfloor$, since this station is waiting to receive a value on channel c; however we can derive the transition $\Gamma_0 \triangleright \lfloor c?(x).P\rfloor \xrightarrow{c?v_0} c[x].P$ using Rule (Rcv).

We can put the three transitions derived above together using rule (Sync), leading to the transition $C_0 \xrightarrow{c!v} W_1$, where $W_1 = \sigma^2 \mid \lfloor d?(x).\mathsf{nil}\rfloor(\lfloor c?(x).Q\rfloor) \mid c[x].P$. □

The transitions for modelling the passage of time, $\Gamma \triangleright W \xrightarrow{\sigma} W'$, are given in Table 3. In the rules (ActRcv) and (EndRcv) we see that the active receiver $c[x].P$ continues to wait for the transmitted value to make its way through the network; when the allocated transmission time elapses the value is then delivered and the receiver evolves to $\{^w/_x\}P$. The rule (SumTime) is necessary to ensure that the passage of time does not resolve non-deterministic choices. Finally (Timeout) implements the idea that $\lfloor c?(x).P\rfloor Q$ is a time-guarded receptor; when time passes it evolves into the alternative Q. However this only happens if the channel c is not exposed. What happens if it is exposed is explained later in Table 4. Finally, Rule (TimePar) models how σ-actions are derived for collections of threads.

Example 2 (Passage of Time). Let $C_1 = \Gamma_1 \triangleright W_1$, where $\Gamma_1(c) = (2, v_0), \Gamma_1 \vdash d$: **idle** and W_1 is the system term derived in Example 1.

We show how a σ-action can be derived for this configuration. First note that $\Gamma_1 \triangleright \sigma^2 \xrightarrow{\sigma} \sigma$; this transition can be derived using Rule (Sleep). Since d is idle in Γ_1, we can apply Rule (TimeOut) to infer the transition $\Gamma_1 \triangleright \lfloor d?(x).\mathsf{nil}\rfloor(\lfloor c?(x).Q\rfloor) \xrightarrow{\sigma} \lfloor c?(x).Q\rfloor$; time passed before a value could be broadcast along channel d, causing a timeout in the

Table 3 Intensional semantics: timed transitions

$$\text{(TimeNil)}\ \frac{}{\Gamma \triangleright \mathsf{nil} \xrightarrow{\sigma} \mathsf{nil}} \qquad \text{(Sleep)}\ \frac{}{\Gamma \triangleright \sigma.P \xrightarrow{\sigma} P}$$

$$\text{(ActRcv)}\ \frac{\Gamma \vdash_t c : n,\ n > 1}{\Gamma \triangleright c[x].P \xrightarrow{\sigma} c[x].P} \qquad \text{(EndRcv)}\ \frac{\Gamma \vdash_t c : 1,\ \Gamma \vdash_v c : w}{\Gamma \triangleright c[x].P \xrightarrow{\sigma} \{^{w}/_{x}\}P}$$

$$\text{(SumTime)}\ \frac{\Gamma \triangleright P \xrightarrow{\sigma} P' \quad \Gamma \triangleright Q \xrightarrow{\sigma} Q'}{\Gamma \triangleright P + Q \xrightarrow{\sigma} \Gamma' \triangleright P' + Q'} \qquad \text{(Timeout)}\ \frac{\Gamma \vdash c : \mathbf{idle}}{\Gamma \triangleright \lfloor c?(x).P \rfloor Q \xrightarrow{\sigma} Q}$$

$$\text{(TimePar)}\ \frac{\Gamma \triangleright W_1 \xrightarrow{\sigma} W_1' \quad \Gamma \triangleright W_2 \xrightarrow{\sigma} W_2'}{\Gamma \triangleright W_1 \mid W_2 \xrightarrow{\sigma} W_1' \mid W_2'}$$

Table 4 Intensional semantics: internal activity

$$\text{(RcvLate)}\ \frac{\Gamma \vdash c : \mathbf{exp}}{\Gamma \triangleright \lfloor c?(x).P \rfloor Q \xrightarrow{\tau} c[x].\{^{\mathsf{err}}/_{x}\}P} \qquad \text{(Tau)}\ \frac{}{\Gamma \triangleright \tau.P \xrightarrow{\tau} P}$$

$$\text{(Then)}\ \frac{[\![b]\!]_{\Gamma} = \text{true}}{\Gamma \triangleright [b]P, Q \xrightarrow{\tau} \sigma.P} \qquad \text{(Else)}\ \frac{[\![b]\!]_{\Gamma} = \text{false}}{\Gamma \triangleright [b]P, Q \xrightarrow{\tau} \sigma.Q}$$

station waiting to receive a value along d. Finally, since $\Gamma_1 \vdash_n c : 2$, we can use Rule (ActRcv) to derive $\Gamma_1 \triangleright c[x].P \xrightarrow{\sigma} c[x].P$.

At this point we can use Rule (TimePar) twice to infer a σ-action performed by C_1. This leads to the transition $C_1 \xrightarrow{\sigma} W_2$, where $W_2 = \sigma \mid \lfloor c?(x).Q \rfloor \mid c[x].P$. □

Table 4 is devoted to internal transitions $\Gamma \triangleright W \xrightarrow{\tau} W'$. Let us first explain rule (RcvLate). Intuitively the process $\lfloor c?(x).P \rfloor Q$ is ready to start receiving a value on an exposed channel c. This means that a transmission is already taking place. Since the process has therefore missed the start of the transmission it will receive an error value. Thus Rule (RcvLate) reflects the fact that in wireless systems a broadcast value cannot be correctly received by a station in the case of a misalignment between the sender and the receiver.

The remaining rules are straightforward except that we we use a channel environment dependent evaluation function for Boolean expressions $[\![b]\!]_{\Gamma}$, because of the presence of the exposure predicate $\exp(c)$ in the Boolean language. However in wireless systems it is not possible to both listen and transmit within the same time unit, as communication is half-duplex, [19]. So in our intensional semantics, in the rules (Then) and (Else), the execution of both branches is delayed of one time unit; this is a slight

Table 5 Intensional semantics: - structural rules

$$\text{(TauPar)}\ \frac{\Gamma \triangleright W_1 \xrightarrow{\tau} W_1'}{\Gamma \triangleright W_1 \mid W_2 \xrightarrow{\tau} W_1' \mid W_2} \qquad \text{(Rec)}\ \frac{\{\mathsf{fix}\, X.P/X\}P \xrightarrow{\lambda} W}{\Gamma \triangleright \mathsf{fix}\, X.P \xrightarrow{\lambda} W}$$

$$\text{(Sum)}\ \frac{\Gamma \triangleright P \xrightarrow{\lambda} W \quad \lambda \in \{\tau, c!v\}}{\Gamma \triangleright P + Q \xrightarrow{\lambda} W} \qquad \text{(SumRcv)}\ \frac{\Gamma \triangleright P \xrightarrow{c?v} W \quad \mathsf{rcv}(P, c) \quad \Gamma \vdash c : \mathbf{idle}}{\Gamma \triangleright P + Q \xrightarrow{c?v} W}$$

$$\text{(ResI)}\ \frac{\Gamma[c \mapsto (n, v)] \triangleright W \xrightarrow{c!v} W'}{\Gamma \triangleright \nu c{:}(n, v).W \xrightarrow{\tau} \nu c{:}\mathrm{upd}_{c!v}(\Gamma)(c).W'} \quad \text{(ResV)}\ \frac{\Gamma[c \mapsto (n, v)] \triangleright W \xrightarrow{\lambda} W',\ c \notin \lambda}{\Gamma \triangleright \nu c{:}(n, v).W \xrightarrow{\lambda} \nu c{:}(n, v).W'}$$

simplification, as evaluation is delayed even if the Boolean expression does not contain an exposure predicate.

Example 3. Let Γ_2 be a channel environment such that $\Gamma_2(c) = (1, v)$, and consider the configuration $C_2 = \Gamma_2 \triangleright W_2$, where W_2 has been defined in Example 2.

Note that this configuration contains an active receiver along the exposed channel c. We can think of such a receiver as a process which missed the synchronisation with a broadcast which has been previously performed along channel c; as a consequence this process is doomed to receive an error value.

This situation is modelled by Rule (RcvLate), which allows us to infer the transition $\Gamma_2 \triangleright \lfloor c?(x).Q \rfloor \xrightarrow{\tau} c[x].\{\mathsf{err}/x\}Q$. As we will see, Rule (TauPar), introduced in Table 5, ensures that τ-actions are propagated to the external environment. This means that the transition derived above allows us to infer the transition $C_2 \xrightarrow{\tau} W_3$, where $W_3 = \sigma \mid c[x].\{\mathsf{err}/x\}Q \mid c[x].P$. □

The final set of rules, in Table 5, are structural. Here we assume that Rules (Sum), (SumRcv) and (SumTime) have a symmetric counterpart. Rules (ResI) and (ResV) show how restricted channels are handled. Intuitively moves from the configuration $\Gamma \triangleright \nu c{:}(n, v).W$ are inherited from the configuration $\Gamma[c \mapsto (n, v)] \triangleright W$; here the channel environment $\Gamma[c \mapsto (n, v)]$ is the same as Γ except that c has associated with it (temporarily) the information (n, v). However if this move mentions the restricted channel c then the inherited move is rendered as an internal action τ, (ResI). Moreover the information associated with the restricted channel in the residual is updated, using the function $\mathrm{upd}_{c!v}(\cdot)$ previously defined.

We are now ready to define the reduction semantics; formally, we let $\Gamma_1 \triangleright W_1 \mathrel{-\!\!\triangleright} \Gamma_2 \triangleright W_2$ whenever $\Gamma_1 \triangleright W_1 \xrightarrow{\lambda} W_2$ and $\Gamma_2 = \mathrm{upd}_\lambda(\Gamma_1)$ for some $\lambda = \tau, \sigma, c!v$.
Note that input actions cannot be used to infer reductions for computations; following the approach of [15,21] reductions are defined to model only the internal of a system. In order to distinguish between timed and untimed reductions in $\Gamma_1 \triangleright W_1 \mathrel{-\!\!\triangleright} \Gamma_2 \triangleright W_2$ we use $\Gamma_1 \triangleright W_1 \mathrel{-\!\!\triangleright}_\sigma \Gamma_2 \triangleright W_2$ if $\Gamma_2 = \mathrm{upd}_\sigma(W_1)$ and $\Gamma_1 \triangleright W_1 \mathrel{-\!\!\triangleright}_i \Gamma_2 \triangleright W_2$ if $\Gamma_2 = \mathrm{upd}_\lambda(\Gamma_1)$ for some $\lambda = \tau, c!v$.

Proposition 1 (Maximal Progress and Time Determinism). *Suppose $C \mathrel{-\!\!\triangleright}_\sigma C_1$; then $C \mathrel{-\!\!\triangleright}_\sigma C_2$ implies $C_1 = C_2$, and $C \not\mathrel{-\!\!\triangleright}_i C_3$ for any C_3.*

Example 4. We now show how the transitions we have inferred in the Examples 1-3 can be combined to derive a computation fragment for the configuration C_0 considered in Example 1.

Let $C_i = \Gamma_i \triangleright W_i, i = 0, \cdots, 2$ be as defined in these examples. Note that $\Gamma_1 = \mathrm{upd}_{c!v_0}(\Gamma_0)$ and $\Gamma_2 = \mathrm{upd}_\sigma(\Gamma_1)$. We have already shown that $C_0 \xrightarrow{c!v_0} W_1$; this transition, together with the equality $\Gamma_1 = \mathrm{upd}_{c!v_0}(\Gamma_0)$, can be used to infer the reduction $C_0 \mathrel{-\!\!\triangleright}_i C_1$. A similar argument shows that $C_1 \mathrel{-\!\!\triangleright}_\sigma C_2$. Also if we let C_3 denote $\Gamma_2 \triangleright W_3$ we also have $C_2 \mathrel{-\!\!\triangleright}_i C_3$ since $\Gamma_2 = \mathrm{upd}_\tau(\Gamma_2)$. □

Example 5 (Collisions). Consider the configuration $C = \Gamma \triangleright W$, where $\Gamma \vdash c : \mathbf{idle}$ and $W = c!\langle w_0 \rangle \mid c!\langle w_1 \rangle \mid \lfloor c?(x).P \rfloor$; here we assume $\delta_{w_0} = \delta_{w_1} = 1$. Using rules (Snd), (RcvIgn), (Rcv) and (Sync) we can infer the transition $\Gamma \triangleright W \xrightarrow{c!w_0} W_1$, where $W_1 = \sigma \mid c!\langle w_1 \rangle \mid c[x].P$. Let $\Gamma_1 := \mathrm{upd}_{c!w_0}(\Gamma)$, that is $\Gamma_1(c) = (1, w_0)$. This equality and the transition above lead to the instantaneous reduction $C \mathrel{-\!\!\triangleright}_i C_1 = \Gamma_1 \triangleright W_1$.

For C_1 we can use the rules (RcvIgn), (Snd) and (Sync) to derive the transition $C_1 \xrightarrow{c!w_1} W_2$, where $W_2 = \sigma \mid \sigma \mid c[x].P$. This transition gives rise to the reduction $C_1 \mathrel{-\!\!\triangleright}_i C_2 = \Gamma_2 \triangleright W_2$, where $\Gamma_2 = \mathrm{upd}_{c!w_1}(\Gamma_1)$. Note that, since $\Gamma_1 \vdash c : \mathbf{exp}$ we obtain that $\Gamma_2(c) = (1, \mathsf{err})$. The broadcast along a busy channel caused a collision to occur.

Finally, rules (Sleep), (EndRcv) and (TimePar) can be used to infer the transition $C_2 \xrightarrow{\sigma} W_3 = \mathsf{nil} \mid \mathsf{nil} \mid \{\mathsf{err}/x\}P$. Let $\Gamma_3 := \mathrm{upd}_\sigma(\Gamma'')$; then the transition above induces the timed reduction $C_2 \mathrel{-\!\!\triangleright}_\sigma C_3 = \Gamma_3 \triangleright W_3$, in which an error is received instead of either of the transmitted values w_0, w_1. □

We now define a contextual equivalence between configurations, following the approach of [10]. This relies on two crucial concepts: a notion of reduction, already been defined, and a notion of minimal observable activity, called a *barb*.

While in other process algebras the basic observable activity is chosen to be an output on a given channel [21,7], for our calculus it is more appropriate to rely on the exposure state of a channel: because of possible collisions transmitted values may never be received. Formally, we say that a configuration $\Gamma \triangleright W$ has a barb on channel c, written $\Gamma \triangleright W \downarrow_c$, whenever $\Gamma \vdash c : \mathbf{exp}$. A configuration $\Gamma \triangleright W$ has a weak barb on c, denoted by $\Gamma \triangleright W \Downarrow_c$, if $\Gamma \triangleright W \mathrel{-\!\!\triangleright}^* \Gamma' \triangleright W'$ for some $\Gamma' \triangleright W'$ such that $\Gamma' \triangleright W' \downarrow_c$. As we will see, it turns out that using this notion of barb we can observe the contents of a message being broadcast only at the end of its transmission. This is in line with the standard theory of wireless networks, in which it is stated that collisions can be observed only at reception time [23,19].

Definition 1. *Let $\mathcal{R}$ be a relation over configurations.*

(1) $\mathcal{R}$ is said to be barb preserving *if $\Gamma_1 \triangleright W_1 \Downarrow_c$ implies $\Gamma_2 \triangleright W_2 \Downarrow_c$, whenever $(\Gamma_1 \triangleright W_1)\ \mathcal{R}\ (\Gamma_2 \triangleright W_2)$.*

(2) It is reduction-closed *if $(\Gamma_1 \triangleright W_1)\ \mathcal{R}\ (\Gamma_2 \triangleright W_2)$ and $\Gamma_1 \triangleright W_1 \mathrel{-\!\!\triangleright} \Gamma_1' \triangleright W_1'$ imply there is some $\Gamma_2' \triangleright W_2'$ such that $\Gamma_2 \triangleright W_2 \mathrel{-\!\!\triangleright}^* \Gamma_2' \triangleright W_2'$ and $(\Gamma_1' \triangleright W_1')\ \mathcal{R}\ (\Gamma_2' \triangleright W_2')$.*

Table 6 Extensional actions

$$\text{(Input)}\ \frac{\mathit{\Gamma} \triangleright W \xrightarrow{c?v} W'}{\mathit{\Gamma} \triangleright W \xmapsto{c?v} \mathrm{upd}_{c?v}(\mathit{\Gamma}) \triangleright W'} \qquad \text{(Time)}\ \frac{\mathit{\Gamma} \triangleright W \xrightarrow{\sigma} W'}{\mathit{\Gamma} \triangleright W \xmapsto{\sigma} \mathrm{upd}_{\sigma}(\mathit{\Gamma}) \triangleright W'}$$

$$\text{(Shh)}\ \frac{\mathit{\Gamma} \triangleright W \xrightarrow{c!v} W'}{\mathit{\Gamma} \triangleright W \xmapsto{\tau} \mathrm{upd}_{c!v}(\mathit{\Gamma}) \triangleright W'} \qquad \text{(TauExt)}\ \frac{\mathit{\Gamma} \triangleright W \xrightarrow{\tau} W'}{\mathit{\Gamma} \triangleright W \xmapsto{\tau} \mathit{\Gamma} \triangleright W'}$$

$$\text{(Deliver)}\ \frac{\mathit{\Gamma}(c) = (1, v) \quad \mathit{\Gamma} \triangleright W \xrightarrow{\sigma} W'}{\mathit{\Gamma} \triangleright W \xmapsto{\gamma(c,v)} \mathrm{upd}_{\sigma}(\mathit{\Gamma}) \triangleright W'} \qquad \text{(Idle)}\ \frac{\mathit{\Gamma} \vdash c : \textbf{idle}}{\mathit{\Gamma} \triangleright W \xmapsto{\iota(c)} \mathit{\Gamma} \triangleright W}$$

(3) It is contextual *if* $\mathit{\Gamma}_1 \triangleright W_1 \ \mathcal{R}\ \mathit{\Gamma}_2 \triangleright W_2$*, implies* $\mathit{\Gamma}_1 \triangleright (W_1 \mid W) \ \mathcal{R}\ \mathit{\Gamma}_2 \triangleright (W_2 \mid W)$ *for all processes* W*.* □

Reduction barbed congruence, written $\simeq$*, is the largest symmetric relation over configurations which is barb preserving, reduction-closed and contextual.*

Example 6. We first give some examples of configurations which are not barbed congruent; here we assume that $\mathit{\Gamma}$ is the stable environment.

- $\mathit{\Gamma} \triangleright c!\langle v_0 \rangle \not\simeq \mathit{\Gamma} \triangleright c!\langle v_1 \rangle$; let $T = \lfloor c?(x).[x = v_0]d!\langle ok \rangle \mathsf{nil}, \rfloor$, where $d \neq c$ and ok is an arbitrary value. It is easy to see that $\mathit{\Gamma} \triangleright c!\langle v_0 \rangle \mid T \Downarrow_d$, whereas $\mathit{\Gamma} \triangleright c!\langle v_1 \rangle \mid T \not\Downarrow_d$.
- $\mathit{\Gamma} \triangleright c!\langle v \rangle \not\simeq \mathit{\Gamma} \triangleright \sigma.c!\langle v \rangle$; let $T = [\exp(c)]d!\langle ok \rangle, \mathsf{nil}$. In this case we have that $\mathit{\Gamma} \triangleright c!\langle v \rangle \mid T \Downarrow_d$, while $\mathit{\Gamma} \triangleright \sigma.c!\langle v \rangle \mid T \not\Downarrow_d$.

On the other hand, consider the configurations $\mathit{\Gamma} \triangleright c!\langle v_0 \rangle \mid c!\langle v_1 \rangle$ and $\mathit{\Gamma} \triangleright c!\langle \mathsf{err} \rangle$, where $\delta_{v_0} = \delta_{v_1}$ and for the sake of convenience we assume that $\delta_{\mathsf{err}} = \delta_{v_0}$. In both cases a communication along channel c starts, and in both cases the value that will be eventually delivered to some receiving station is err, independently of the behaviour of the external environment. This gives us the intuition that these two configurations are barbed congruent. Later in the paper we will develop the tools that will allow us to prove this statement formally. □

4 Extensional Semantics

In this section we give a co-inductive characterisation of the contextual equivalence $\simeq$ between configurations, using a standard bisimulation equivalence over an extensional LTS, with configurations as nodes, but with a special collection of *extensional actions*; these are defined in Table 6.

Rule (Input) simply states that input actions are observable, as is the passage of time, by Rule (Time). Rule (TauExt) propagates τ-intensional actions to the extensional semantics. Rule (Shh) states that broadcasts are always treated as internal activities in the extensional semantics. This choice reflects the intuition that the content of a message being broadcast cannot be detected immediately; in fact, it cannot be detected until the end of the transmission.

Rule (Idle) introduces a new label $\iota(c)$, parameterized in the channel c, which is not inherited from the intensional semantics. Intuitively this rules states that it is possible to observe whether a channel is exposed. Finally, Rule (Deliver) states that the delivery of a value v along channel c is observable, and it corresponds to a new action whose label is $\gamma(c, v)$. In the following we range over extensional actions by α.

Example 7. Consider the configuration $\Gamma \triangleright c!\langle v\rangle$, where Γ is the stable channel environment. By an application of Rule (Shh) we have the transition $\Gamma \triangleright c!\langle v\rangle \xmapsto{\tau} \Gamma' \triangleright \sigma^{\delta_v}$, with $\Gamma' \vdash c : \mathbf{exp}$. Furthermore, $\Gamma \triangleright c!\langle v\rangle \xmapsto{\iota(c)}$ since channel c is idle in Γ. Notice that $\Gamma' \triangleright \sigma^{\delta_v}$ cannot perform a $\iota(c)$ action, and that the extensional semantics gives no information about the value v which has been broadcast.

The extensional semantics endows configurations with the structure of an LTS. Weak extensional actions in this LTS are defined as usual, and the formulation of bisimulations is facilitated by the notation $C \overset{\hat{\alpha}}{\Longmapsto} C'$, which is again standard: for $\alpha = \tau$ this denotes $C \longmapsto^* C'$ while for $\alpha \neq \tau$ it is $C \xmapsto{\tau}^* \xmapsto{\alpha} \xmapsto{\tau}^* C'$.

Definition 2 (Bisimulations). *Let $\mathcal{R}$ be a symmetric binary relation over configurations. We say that $\mathcal{R}$ is a (weak) bisimulation if for every extensional action α, whenever $C_1 \,\mathcal{R}\, C_2$, then $C_1 \overset{\alpha}{\Longmapsto} C_1'$ implies $C_2 \overset{\hat{\alpha}}{\Longmapsto} C_2'$ for some C_2' satisfying $C_1' \,\mathcal{R}\, C_2'$ We let $\approx$ be the the largest bisimulation.* □

Example 8. Let us consider again the configurations $\Gamma \triangleright W_0 = c!\langle v_0\rangle \mid c!\langle v_1\rangle$, $\Gamma \triangleright W_1 = c!\langle \mathsf{err}\rangle$ of Example 6. Recall that in this example we assumed that Γ is the stable channel environment; further, $\delta_{v_0} = \delta_{v_1} = \delta_{\mathsf{err}} = k$ for some $k > 0$.

We show that $\Gamma \triangleright W_0 \approx \Gamma \triangleright W_1$ by exhibiting a witness bisimulation $\mathcal{S}$ such that $\Gamma \triangleright W_0 \;\mathcal{S}\; \Gamma \triangleright W_1$. To this end, let us consider the relation

$$\begin{aligned} \mathcal{S} = \{ \; & (\Delta \triangleright W_0, \Delta \triangleright W_1) \quad , (\Delta' \triangleright \sigma^k \mid c!\langle v_1\rangle, \Delta'' \triangleright \sigma^k) \; , (\Delta' \triangleright c!\langle v_0\rangle, \Delta'' \triangleright \sigma^k) \\ , \quad & (\Delta \triangleright \sigma^j \mid \sigma^j, \Delta \triangleright \sigma^j) \mid \Delta' \vdash_t c : n, \Delta''(c) = (n, \mathsf{err}) \text{ for some } n > 0, j \leq k\} \end{aligned}$$

Note that this relation contains an infinite number of pairs of configurations, which differ by the state of channel environments.This is because input actions can affect the channel environment of configurations. It is easy to show that the relation $\mathcal{S}$ is a bisimulation which contains the pair $(\Gamma_0 \triangleright W_0, \Gamma_1 \triangleright W_1)$, therefore $\Gamma \triangleright W_0 \approx \Gamma \triangleright W_1$. □

One essential property of weak bisimulation is that it does not relate configurations which differ by the exposure state of some channel:

Proposition 2. *Suppose $\Gamma_1 \triangleright W_1 \approx \Gamma_2 \triangleright W_2$. Then for any channel c, $\Gamma_1 \vdash c$: **idle** iff $\Gamma_2 \vdash c$: **idle**.* □

5 Full abstraction

The aim of this section is to prove that weak bisimilarity in the extensional semantics is a proof technique which is both sound and complete for reduction barbed congruence.

Theorem 1 (Soundness). *$C_1 \approx C_2$ implies $C_1 \simeq C_2$.*

Proof. It suffices to prove that bisimilarity is reduction-closed, barb preserving and contextual. Reduction closure follows from the definition of bisimulation equivalence. The preservation of barbs follows directly from Proposition 2. The proof of contextuality on the other hand is quite technical, and is addressed in detail in the associated technical report [4]. One subtlety lies in the definition of τ-extensional actions, which include broadcasts. While broadcasts along exposed do not affect the external environment, and hence cannot affect the external environment, this is not true for broadcasts performed along idle channels. However, we can take advantage of Proposition 2 to show that these extensional τ-actions preserve the contextuality of bisimilar configurations. □

To prove completeness, the converse of Theorem 1, we restrict our attention to the subclass of *well-formed* configurations. Informally $\Gamma \triangleright W$ is well-formed if the system term W does not contain active receivers along idle channels; a wireless station cannot be receiving a value along a channel if there is no value being transmitted along it.

Definition 3 (Well-formedness). *The set of well-formed configurations WNets is the least set such that $\Gamma \triangleright P \in$ Wnets for all processes P, if $\Gamma \vdash c :$ **exp** then $\Gamma \triangleright c[x].P \in$ WNets, is closed under parallel composition and such that if $\Gamma[c \mapsto (n, v)] \triangleright W \in$ WNets then $\Gamma \triangleright \nu c : (n, v).W \in$ WNets.* □

By focusing on well-formed configurations we can prove a counterpart of Proposition 2 for our contextual equivalence:

Proposition 3. *Let $\Gamma_1 \triangleright W_1, \Gamma_2 \triangleright W_2$ be two well formed configurations such that $\Gamma_1 \triangleright W_1 \simeq \Gamma_2 \triangleright W_2$. Then for any channel c, $\Gamma_1 \vdash c :$ **idle** implies $\Gamma_2 \vdash c :$ **idle**.* □

Proposition 3 does not hold for ill-formed configurations. For example, let $\Gamma_1 \vdash c :$ **exp**, $\Gamma_1 \vdash d :$ **idle** and $\Gamma_2 \vdash c, d :$ **idle** and consider the two configurations $C_1 = \Gamma_1 \triangleright \mathsf{nil} \mid d[x].P$ and $C_2 = \Gamma_2 \triangleright c!\langle v\rangle \mid d[x].P$, neither of which are well-formed; nor do they let time pass, $C_i \not\rightarrowtriangle_\sigma$. As a consequence $C_1 \simeq C_2$. However Proposition 2 implies that they are not bisimilar, since they differ on the exposure state of c.

Another essential property of well-formed systems is patience: time can always pass in networks with no instantaneous activities.

Proposition 4 (Patience). *If C is well-formed and $C \not\rightarrowtriangle_i$, then $C \rightarrowtriangle_\sigma C'$ for some C'.*□

This means that, if we restrict our attention to well-formed configurations, we can never reach a configuration which is deadlocked; at the very least time can always proceed.

Theorem 2 (Completeness). *On well-formed configurations, reduction barbed congruence implies bisimilarity.*

The proof relies on showing that for each extensional action α it is possible to exhibit a test T_α which determines whether or not a configuration $\Gamma \triangleright W$ can perform the action α. The main idea is to equip the test with some fresh channels; the test T_α is designed so that a configuration $\Gamma \triangleright W \mid T_\alpha$ can reach another one $C' = \Gamma' \triangleright W' \mid T'$, where T' is determined uniquely by the barbs of the introduced fresh channel; these are enabled in $\Gamma' \triangleright T'$, if and only if C can weakly perform the action α.

The tests T_α are defined by performing a case analysis on the extensional action α:

$$\begin{array}{ll}
T_\tau & = \mathit{eureka}!\langle ok\rangle \\
T_\sigma & = \sigma.(\tau.\mathit{eureka}!\langle ok\rangle + \mathit{fail}!\langle no\rangle) \\
T_{\gamma(c,v)} & = \nu d{:}(0,\cdot).((c[x].([x{=}v]d!\langle ok\rangle, \mathsf{nil}) + \mathit{fail}!\langle no\rangle) \mid \\
 & \mid\ \sigma^2.[\exp(d)]\mathit{eureka}!\langle ok\rangle, \mathsf{nil} \mid \sigma.\mathit{halt}!\langle ok\rangle) \\
T_{c?v} & = (c\,!\langle v\rangle.\mathit{eureka}!\langle ok\rangle + \mathit{fail}!\langle no\rangle) \mid \mathit{halt}!\langle ok\rangle \\
T_{\iota(c)} & = ([\exp(c)]\mathsf{nil}, \mathit{eureka}!\langle ok\rangle) + \mathit{fail}!\langle no\rangle \mid \mathit{halt}!\langle ok\rangle
\end{array}$$

where $\mathit{eureka}, \mathit{fail}, \mathit{halt}$ are arbitrary distinct channels and ok, no are two values such that $\delta_{ok} = \delta_{no} = 1$.

For the sake of simplicity, for any action α we define also the tests T'_α as follows:

$$\begin{array}{ll}
T'_\tau = T'_\sigma & = \mathit{eureka}!\langle ok\rangle \\
T'_{\gamma(c,v)} & = \nu d{:}(0,\cdot).(\sigma.d!\langle ok\rangle\mathsf{nil} \mid \sigma.[\exp(d)]\mathit{eureka}!\langle ok\rangle, \mathsf{nil} \mid \mathit{halt}!\langle ok\rangle) \\
T'_{c?v} & = \sigma^{\delta_v}.\mathit{eureka}!\langle ok\rangle \mid \mathit{halt}!\langle ok\rangle \\
T'_{\iota(c)} & = \sigma.\mathit{eureka}!\langle ok\rangle \mid \mathit{halt}!\langle ok\rangle
\end{array}$$

Proposition 5 (Distinguishing contexts). *Let $\Gamma \triangleright W$ be a well-formed configuration, and suppose that the channels* eureka, halt, fail *do not appear free in W, nor they are exposed in Γ. Then for any extensional action α, $\Gamma \triangleright W \stackrel{\alpha}{\Longmapsto} \Gamma' \triangleright W'$ iff $\Gamma \triangleright W \mid T_\alpha \rightarrowtriangle^* \Gamma' \triangleright W' \mid T'_\alpha$.* □

A pleasing property of the tests T'_α is that they can be identified by the (both strong and weak) barbs that they enable in a computation rooted in the configuration $\Gamma \triangleright W \mid T_\alpha$.

Proposition 6 (Uniqueness of successful testing components). *Let $\Gamma \triangleright W$ be a configuration such that* eureka, halt, fail *do not appear free in W, nor they are exposed in Γ. Suppose that $\Gamma \triangleright W \mid T_\alpha \rightarrowtriangle^* C'$ for some configuration C' such that*

- *if $\alpha = \tau, \sigma$, then $C' \not\downarrow_{\mathsf{eureka}}$, $C' \Downarrow_{\mathsf{eureka}}$, $C' \not\Downarrow_{\mathit{fail}}$,*
- *otherwise, $C' \not\downarrow_{\mathsf{eureka}}$, $C' \not\downarrow_{\mathit{halt}}$, $C' \Downarrow_{\mathsf{eureka}}$, $C' \Downarrow_{\mathit{halt}}$, $C' \not\Downarrow_{\mathit{fail}}$.*

Then $C' = \Gamma' \triangleright W' \mid T'_\alpha$ for some configuration $\Gamma' \triangleright W'$. □

Note the use of the fresh channel *halt* when testing some of these actions. This is because of a time mismatch between a process performing the action, and the test used to detect it. For example the weak action $\stackrel{\iota(c)}{\Longmapsto}$ does not involve the passage of time but the corresponding test uses a branching construct which needs at least one time step to execute. Requiring a weak barb on *halt* in effect prevents the passage of time.

Outline proof of Theorem 2: It is sufficient to show that reduction barbed congruence, $\simeq$, is a bisimulation. As an example suppose $\Gamma_1 \triangleright W_1 \simeq \Gamma_2 \triangleright W_2$ and $\Gamma_1 \triangleright W_1 \stackrel{\gamma(c,v)}{\longmapsto} \Gamma'_1 \triangleright W'_1$. We show how to find a matching move from $\Gamma_2 \triangleright W_2$.

Suppose that $\Gamma_1 \triangleright W_1 \stackrel{\gamma(c,v)}{\longmapsto} \Gamma'_1 \triangleright W'_1$, we need to show that $\Gamma_2 \triangleright W_2 \stackrel{\gamma(c,v)}{\Longmapsto} \Gamma'_2 \triangleright W'_2$ for some $\Gamma'_2 \triangleright W'_2$ such that $\Gamma'_1 \triangleright W'_1 \simeq \Gamma'_2 \triangleright W'_2$. By Proposition 5 we know that $\Gamma_1 \triangleright W_1 \mid T_{\gamma(c,v)} \rightarrowtriangle^* \Gamma'_1 \triangleright W'_1 \mid T'_\alpha$.By the hypothesis it follows that $\Gamma_1 \triangleright W_1 \mid T_{\gamma(c,v)} \simeq \Gamma_2 \triangleright W_2 \mid T_{\gamma(c,v)}$, therefore $\Gamma_2 \triangleright W_2 \mid T_{\gamma(c,v)} \rightarrowtriangle^* C_2$ for some $C_2 \simeq \Gamma'_1 \triangleright W'_1 \mid T'_{\gamma(c,v)}$.

Let $C_1 = \varGamma'_1 \triangleright W'_1 \mid T'_{\gamma(c,v)}$. It is easy to check that $C_1 \not\downarrow_{\mathit{eureka}}, C_1 \not\downarrow_{\mathit{halt}}, C_1 \not\Downarrow_{\mathit{fail}}$ and $C_1 \Downarrow_{\mathit{eureka}}, C_1 \Downarrow_{\mathit{halt}}$. By definition of reduction barbed congruence and Proposition 3 we obtain that $C_2 \not\downarrow_{\mathit{eureka}}, C_2 \not\downarrow_{\mathit{halt}}$, $C_2 \Downarrow_{\mathit{eureka}}, C_2 \Downarrow_{\mathit{halt}}$ and $C_2 \not\Downarrow_{\mathit{fail}}$. Proposition 6 ensures then that $C_2 = \varGamma'_2 \triangleright W'_2 \mid T'_{\gamma(c,v)}$ for some $\varGamma'_2, W'_2$. An application of Proposition 5 leads to $\varGamma_2 \triangleright W_2 \overset{\gamma(c,v)}{\Longmapsto} \varGamma'_2 \triangleright W'_2$. Now standard process calculi techniques enable us to infer from this that $\varGamma'_1 \triangleright W'_1 \simeq \varGamma'_2 \triangleright W'_2$. □

6 Conclusions and Related work

In this paper we have given a behavioural theory of wireless systems at the MAC level. We believe that our reduction semantics, given in Section 2, captures much of the subtlety of intensional MAC-level behaviour of wireless systems. We also believe that our behavioural theory is the only one for wireless networks at the MAC-Layer which is both sound and complete. The only other calculus which considers such networks is TCWS from [14] which contains a sound theory; as we have already stated we view CCCP as a simplification of this TCWS, and by using a more refined notion of extensional action we also obtain completeness.

We are aware of only two other papers modelling networks at the MAC-Sublayer level of abstraction, these are [12,24]. They present a calculus CWS which views a network as a collection of nodes distributed over a metric space. [12] contains a reduction and an intensional semantics and the main result is their consistency. In [24], time and node mobility is added.

On the other hand there are numerous papers which consider the problem of modelling networks at a higher level. Here we briefly consider a selection; for a more thorough review see [4].

Nanz and Hankin [16] have introduced an untimed calculus for Mobile Wireless Networks (CBS$^\sharp$), relying on a graph representation of node localities. The main goal of that paper is to present a framework for specification and security analysis of communication protocols for mobile wireless networks. Merro [13] has proposed an untimed process calculus for mobile ad-hoc networks with a labelled characterisation of reduction barbed congruence, while [6] contains a calculus called CMAN, also with mobile ad-hoc networks in mind. Singh, Ramakrishnan and Smolka [22] have proposed the ω-calculus, a conservative extension of the π-calculus. A key feature of the ω-calculus is the separation of a node's communication and computational behaviour from the description of its physical transmission range. Another extension of the π-calculus, which has been used for modelling the LUNAR ad-hoc routing protocol, may be found in [2].

In [3] a calculus is proposed for describing the probabilistic behaviour of wireless networks. There is an explicit representation of the underlying network, in terms of a connectivity graph. However this connectivity graph is static. In contrast Ghassemi et al. [5] have proposed a process algebra called RBPT where topological changes to the connectivity graph are implicitly modelled in the operational semantics rather than in the syntax. Kouzapas and Philippou [11] have developed a theory of confluence for a calculus of dynamic networks and they use their machinery to verify a leader-election algorithm for mobile ad hoc networks.

References

1. I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38(4):393–422, March 2002.
2. J. Borgström, S. Huang, M. Johansson, P. Raabjerg, B. Victor, J. Å. P., and Joachim Parrow. Broadcast psi-calculi with an application to wireless protocols. In *SEFM*, volume 7041 of *LNCS*, pages 74–89. Springer, 2011.
3. A. Cerone and M. Hennessy. Modelling probabilistic wireless networks (extended abstract). In Holger Giese and Grigore Rosu, editors, *FMOODS/FORTE*, volume 7273 of *LNCS*, pages 135–151. Springer, 2012.
4. A. Cerone, M. Hennessy, and M. Merro. Modelling mac-layer communications in wireless systems. Technical Report, Trinity College Dublin, 2012. `https://www.scss.tcd.ie/~acerone/works/CCCP.pdf`.
5. F. Ghassemi, W. Fokkink, and A. Movaghar. Equational reasoning on mobile ad hoc networks. *Fundamenta Informaticae*, 105(4):375–415, 2010.
6. J.C. Godskesen. A Calculus for Mobile Ad Hoc Networks. In *COORDINATION*, volume 4467 of *LNCS*, pages 132–150. Springer Verlag, 2007.
7. M. Hennessy. *A distributed Pi-calculus*. Cambridge University Press, 2007.
8. M. Hennessy and J. Rathke. Bisimulations for a calculus of broadcasting systems. *TCS*, 200(1–2):225–260, 1998.
9. M. Hennessy and T. Regan. A process algebra for timed systems. *IaC*, 117(2):221–239, March 1995.
10. K. Honda and N. Yoshida. On reduction-based process semantics. *TCS*, 152(2):437–486, 1995.
11. D. Kouzapas and A. Philippou. A process calculus for dynamic networks. In *FMOODS/FORTE*, volume 6722 of *LNCS*, pages 213–227. Springer, 2011.
12. I. Lanese and D. Sangiorgi. An operational semantics for a calculus for wireless systems. *TCS*, 411(19):1928–1948, 2010.
13. M. Merro. An Observational Theory for Mobile Ad Hoc Networks (full paper). *IaC*, 207(2):194–208, 2009.
14. M. Merro, F. Ballardin, and E. Sibilio. A timed calculus for wireless systems. *TCS*, 412(47):6585–6611, 2011.
15. R. Milner. *Communicating and Mobile Systems: The π-calculus*. Cambridge University Press, 1999.
16. S. Nanz and C. Hankin. Static analysis of routing protocols for ad-hoc networks. In *ACM SIGPLAN and IFIP WG*, volume 1, pages 141–152. Citeseer, 2004.
17. X. Nicollin and J. Sifakis. The algebra of timed processes, atp: Theory and application. *IaC*, 114(1):131–178, 1994.
18. K. V. S. Prasad. A calculus of broadcasting systems. *SCP*, 25(2–3):285–327, December 1995. ESOP '94 (Edinburgh, 1994).
19. T.S. Rappaport. *Wireless communications - principles and practice*. Prentice Hall, 1996.
20. J. Rathke and P. Sobocinski. Deconstructing behavioural theories of mobility. In *Fifth IFIP ICTCScience*, volume 273 of *IFIP*, pages 507–520. Springer, 2008.
21. D. Sangiorgi and D. Walker. *The Pi-Calculus — A Theory of Mobile Processes*. Cambridge University Press, 2001.
22. A. Singh, C.R. Ramakrishnan, and S.A. Smolka. A process calculus for mobile ad hoc networks. *SCP*, 75(6):440 – 469, 2010.
23. A.S. Tanenbaum. *Computer Networks, 4th ed.* Prentice-Hall International, Inc., 2003.
24. M. Wang and Y. Lu. A timed calculus for mobile ad hoc networks. *arXiv preprint arXiv:1301.0045*, 2013.
25. W. Yi. *A Calculus of Real Time Systems*. Ph.D Thesis, Chalmers University, 1991.

A Case Study: TDMA versus Routing

As an example of the use of CCCP we consider two different mechanisms for broadcasting values between wireless stations; we prove that at least in a simple instance they lead to behaviourally equivalent systems. The two different mechanisms are described as configurations, $\mathcal{C}_0$, $\mathcal{C}_1$ below, and we establish $\mathcal{C}_0 \simeq \mathcal{C}_1$ indirectly by proving $\mathcal{C}_0 \approx \mathcal{C}_1$. This in turn is proved by finding a specification of the overall behaviour of the systems $\mathcal{S}$, and proving separately $\mathcal{C}_0 \approx \mathcal{S}$ and $\mathcal{C}_1 \approx \mathcal{S}$; this is achieved by exhibiting bisimulations in the extensional LTS which contain the relevant pairs.

Let us first consider how the TDMA modulation technique [23] can be described in CCCP. *Time Division Multiple Access* (TDMA) is a type of Time Division Multiplexing, where instead of having one transmitter connected to one receiver, there are multiple transmitters. TDMA is used in the digital 2G cellular systems such as *Global System for Mobile Communications* (GSM). TDMA allows several users to share the same frequency channel by dividing the signal into different time slots. The users transmit in rapid succession, one after the other, each using their own time slot. This allows multiple stations to share the same transmission medium (e.g. radio frequency channel) while using only a part of its channel capacity. But to apply TDMA the internal clocks of wireless stations need to be synchronised; otherwise the broadcasts can overlap, thus causing a collision.

Here we show how to model a simple wireless network where two stations try to transmit two different values, v_0 and v_1 respectively, along the same channel using the TDMA modulation technique. We assume that the synchronisation between these wireless stations can fail, leading to a situation in which they broadcast a value in the same slot of time.

For this example we assume $\delta_{v_0} = \delta_{v_1} = 2$ and use an error value err such that $\delta_{\mathsf{err}} = 2$. The main idea here is to split each of the values v_i into two packets of length one, transmit the packets individually, which will then be concatenated together before being forwarded to the external environment. So let us assume values $v_0^0, v_0^1, v_1^0, v_1^1$, each of which requires one instant of time to be transmitted, and a binary operator $\circ$ for composing values such that

$$v_0^0 \circ v_0^1 = v_0, \qquad v_1^0 \circ v_1^1 = v_1, \qquad \text{and } v \circ \mathsf{err} = \mathsf{err} \circ v = \mathsf{err} \text{ for any value } v$$

Next we model four different wireless stations, s_0, s_1, r_0, r_1, running the code $\hat{S}_0$, $\hat{S}_1$, $\hat{R}_0$, $\hat{R}_1$ respectively, with the overall network being described by

$$\mathcal{C}_0 = \Gamma \triangleright \nu d : (0, \cdot).(\hat{S}^0 \mid \hat{S}^1 \mid \hat{R}_0 \mid \hat{R}_1)$$

where Γ is the stable channel environment. The individual station codes are given by:

$$\begin{aligned}
\hat{S}_0 &= d\,!\langle v_0^0\rangle.\sigma.d!\langle v_0^1\rangle + \tau.\sigma.d\,!\langle v_0^0\rangle.\sigma.d!\langle v_0^1\rangle \\
\hat{S}_1 &= d\,!\langle v_1^0\rangle.\sigma.d!\langle v_1^1\rangle + \tau.\sigma.d\,!\langle v_1^0\rangle.\sigma.d!\langle v_1^1\rangle \\
\hat{R}_0 &= \lfloor d?(x).\sigma.\lfloor d?(y).\sigma.c!\langle x \circ y\rangle\rfloor\rfloor \\
\hat{R}_1 &= \sigma.\lfloor d?(x).\sigma.\lfloor d?(y).\sigma^2.c!\langle x \circ y\rangle\rfloor\rfloor
\end{aligned}$$

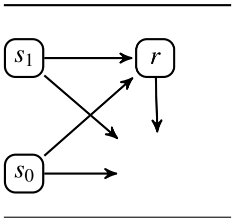
The station s_i, running the thread $\hat{S}_i$, wishes to broadcast value v_i, on the same (restricted) channel d. Both stations split their value into its packets and non-deterministically chooses the time frame to be used for sending the individual packets. For example, the two packets in which the value v_0 has been split can be sent either on the first and third time slot, or on the second and the fourth. The same applies to the station s_1 and value v_1.

Since channel d is restricted in our network, the broadcasts of the individual packets in which the values v_0, v_1 have been split cannot be observed externally. But the two other stations r_o, r_1 wait to collect the packets broadcast along d. The first, r_o is interested only in packets sent in the first time frame (that is in the first and third time slots), while r_1 detects only values sent in the second time frame (second and fourth time slots). At the end of their associated time frame, the stations r_0 and r_1 have received two packets; these are concatenated together and then broadcast to the external environment along channel c. Note that station r_1 is a little slower than r_0, for we have added a delay of two time units before broadcasting the concatenated values.

Let Spec be the specification given by

$$\text{Spec} = \tau.\sigma^4.c\,!\langle v_0\rangle.c!\langle v_1\rangle \;+\; \tau.\sigma^4.c\,!\langle v_1\rangle.c!\langle v_0\rangle \;+\; \tau.\sigma^4.c!\langle \mathsf{err}\rangle \;+\; \tau.\sigma^6.c!\langle \mathsf{err}\rangle \qquad (1)$$

and let $\mathcal{S}$ denote the configuration $\Gamma \triangleright \text{Spec}$. In the full version of the paper we have described the behaviour of the networks C_0 and $\mathcal{S}$ in detail, allowing the reader to infer a bisimulation which contains the pair $(C_0, \mathcal{S})$.



As an alternative to TDMA, the two values v_0, v_1 can also be delivered to the external environment by means of a simple routing. Note that in CCCP we can use channel restriction to refine the topological structure of networks. For example in the network to the left we have three stations, s_0, s_1 and r. Let us suppose that stations s_0 and s_1 can only broadcast messages, either to the external environment, or to the station r. However broadcasts directed to r can not be observed externally. On the other hand, station r can only broadcast values to the external environment; in some sense station r acts as an intermediary internal station.

A network with this topology can be modelled by a configuration of the form

$$\Gamma \triangleright \nu d{:}(0,\cdot).(S_0 \mid S_1 \mid R),$$

where S_0, S_1, R are the threads run by the stations s_0, s_1, r respectively, and the restricted channel d implements in some sense the above network topology. Specifically

- S_0 and S_1 do not contain any receiver operator; this constraint models the fact that the stations s_0, s_1 cannot detect any message broadcast by any of the stations s_0, s_1, r or the external environment,
- S_0 and S_1 can broadcast along any channel,
- R only contains receivers along the restricted channel d; this is because the station r can detect messages broadcast by s_0, s_1, but it cannot detect any message broadcast by the external environment.

Let us consider a particular example, where

$$S_0 = \tau.\sigma^4.c!\langle v_0\rangle + \tau.\sigma^4.d!\langle v_0\rangle, \qquad S_1 = \tau.\sigma^4.c!\langle v_1\rangle + \tau.\sigma^4.d!\langle v_1\rangle, \quad R = d?(x).c!\langle x\rangle$$

where $d?(x).P$ is a shortcut for denoting the process $\mathsf{fix}\, X.\lfloor d?(x).P \rfloor X$; intuitively this is a persistent listener at a channel d. In this configuration the wireless station s_0 chooses whether to broadcast the value v_0 to the external environment, or to forward it to the station r. In either case the broadcast will happen only after 4 instants of time. The behaviour of the station s_1 is similar, using value v_1. Finally, station r behaves as a forwarder; as soon as it receives a value it broadcasts it to the external environment along channel c.

There are four different possible evolution for this network. In the first case, both the stations s_0 and s_1 broadcast the values v_0, v_1 directly to the external environment along channel c, thus causing a collision. As a result, the external environment will start receiving the error value err after 4 instants of time. In the second scenario, the stations forward the values to r, in which case the station r will actually receive the error value since again the transmission of the values v_i has caused a collision. Since r will forward the received value to the external environment, in this case the broadcast of an error value will be observed after 6 instants of time.

The last two scenarios are symmetric; here we describe only one. The station s_0 broadcasts the value v_0 directly to the external environment, while s_1 decides to broadcast the value v_1 to station r. As a result, value v_0 is correctly received by the external environment, its transmission beginning after 4 instants of time. The value v_1 is also received correctly, with its transmission beginning after 6 instants of time.

Again in the full version of the paper we have described the formal behaviour of $\mathcal{C}_1$ and $\mathcal{S}$ in detail. We also exhibited a bisimulation between two systems $\mathcal{C}_1', \mathcal{S}'$, which can be seen as simplifications of the networks $\mathcal{C}_1$ and $\mathcal{S}$ where collisions are not present; this bisimulation has then been used to outline how to build a larger bisimulation which contains the pair $(\mathcal{C}_1, \mathcal{S})$.

Since $\mathcal{C}_0 \approx \mathcal{S} \approx \mathcal{C}_1$, Theorem 1 ensures that $\mathcal{C}_0 \simeq \mathcal{C}_1$; that is, we have related behaviourally two networks which try to forward two different values to the external environment, one using TDMA and the other one using routing. Note that, since we explicitly inserted a time delay in the code of the stations in $\mathcal{C}_1$, in this particular scenario TDMA is slower than routing.