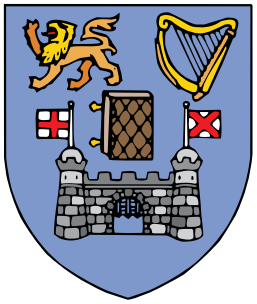


TRINITY COLLEGE DUBLIN

COLÁISTE NA TRÍONÓIDE, BAILE ÁTHA CLIATH

Modelling MAC-layer communications in wireless systems

Andrea Cerone, Matthew Hennessy

Trinity College Dublin, Ireland

Massimo Merro

Università degli Studi di Verona, Italy

Computer Science Department Technical Report **TCD-CS-2012-17**

Foundations and Methods Research Group **1st October 2012**

Modelling MAC-layer communications in wireless systems

Andrea Cerone and Matthew Hennessy*

Trinity College Dublin, Ireland

Massimo Merro

Università degli Studi di Verona, Italy

1st October 2012

Abstract

We present a timed process calculus for modelling wireless networks in which individual stations broadcast and receive messages; moreover the broadcasts are subject to collisions. Based on a reduction semantics for the calculus we define a contextual equivalence to compare the external behaviour of such 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 LTS provide a sound proof method for proving systems contextually equivalence. We illustrate the usefulness of the proof methodology by a series of examples. Finally we show that this proof method is also complete, for a large class of systems.

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

1 Introduction

Wireless networks are becoming increasingly pervasive with applications across many domains, [38, 1]. They are also becoming increasingly complex, with their behaviour depending on ever more sophisticated protocols. Assuring the correctness of their behaviour has always been difficult, and with the increase in their complexity this problem will get even more urgent. This paper addresses this issue by proposing:

(1) a process calculus for describing wireless networks

(2) a semantic theory for comparing their extensional behaviour, based on their performance when embedded in larger systems

(3) a sound and complete co-inductive proof methodology for guaranteeing their extensional behaviour.

There are many different levels of abstraction at which wireless networks might be described, [29, 24, 34, 8]; our process calculus, called the Calculus of Collision-prone Communicating Processes (CCCP), is designed for modelling the behaviour of protocols designed at the *MAC sub-layer* of the *ISO/OSI Protocol Suite* [45, 22]. Specifically, it is designed around the following concepts:

- Values are broadcast along channels between all wireless stations; for the sake of simplicity, we assume a flat network topology. Broadcasts can not be delayed and happen whether or not there are any stations ready to consume communications.
- Communication between stations take time; each value v has a designated amount of time δ_v which is needed for it to be sent along a channel.
- Communications are subject to possible collisions; if more than one value ends up being transmitted simultaneously on a channel a collision occurs and receivers are notified of an error.
- Stations listening on a channel c automatically initiate reception as soon as a transmission is detected. However, the successful transmission of a value between stations depends on the transmitting station and the receiving stations being correctly synchronised.

The calculus is described in detail in Section 2. Formally (a state of) a wireless system will be given by a *configuration* of the form $\Gamma \triangleright W$ where W describes the code running at individual wireless stations and Γ represents the state of the associated communication network. At any given point of time there will be *exposed* communication channels, that is channels containing values in transmission; this information will be recorded by Γ. The main object of Section 2 is to describe how a given system evolves over time. This is defined in terms of a *reduction semantics*, whose judgements take the form

$$\Gamma_1 \triangleright W_1 \twoheadrightarrow \Gamma_2 \triangleright W_2$$

This represents one step in the evolution of the system, and may model the passage of a discrete time step, the broadcast of a message between stations, or some other internal activity. We will illustrate this semantics using a series of simple examples, and also show that it satisfies some expected properties such as *time determinism* and *maximal progress*, [35, 18, 47].

However the main aim of the paper is to develop a behavioural theory of wireless systems. 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 [19, 41], often called *reduction barbed congruence*, rather than that of [30];[1] 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 unexposed channel. Section 2 ends with some examples of expected equivalences between systems.

As explained in papers such as [39, 17], 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. It turns out that 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, to gain a proper extensional account of systems we also need to take this into account.

This is the topic of Section 3.1 and in the standard manner [31] these actions determine an LTS (labelled transition system) over systems, which in turn gives rise to the standard notion of (weak) bisimulation equivalence between systems. 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. This is illustrated via a number of small examples in Section 3.2; more substantial wireless systems are considered in Section 5.

In Section 4 we prove that our proof technique 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. We also show completeness for a so-called *well-formed* systems: if two such systems are contextually equivalent then there is some bisimulation which contains them. The paper ends with a brief comparison with other work.

2 The calculus

As already explained a wireless system will be represented by a *configuration* of the form $\Gamma \triangleright W$ where W describes the code running at individual wireless stations and Γ is a channel environment

[1] See page 106 of [43] for a brief discussion of the difference.

Table 1 Wireless systems

W ::=	P	station code
	$c[x].P$	active receiver
|	$W_1 \mid W_2$	parallel composition
|	$\nu c{:}(n, v).W$	channel restriction

Network: $\Gamma : C \to \mathbb{N}^\infty \times \mathcal{V}$

containing the transmission information for channels. A possible evolution of a system will then be given by a sequence of computation steps:

$$\Gamma_1 \triangleright W_1 \twoheadrightarrow \Gamma_2 \triangleright W_2 \twoheadrightarrow \ldots\ldots \twoheadrightarrow \Gamma_k \triangleright W_k \ldots \twoheadrightarrow \ldots \tag{1}$$

where intuitively each step corresponds to either the passage of time, a broadcast from a station, or some unspecified internal computation; the code running at stations evolves as a computation proceeds, but so also does the state of the underlying channel environment. In the following we will use the meta-variable C to range over configurations.

2.1 Syntax

The syntax for configurations is given in Table 1, where P ranges over code for programming individual stations, explained in Table 2. Here c ranges over some set of channel names and v is an element from some unspecified set of values $\mathcal{V}$; all we require of this set is that it contains some error constant err, which denotes a faulty transmission, and that each value v (included err) is associated with a strictly positive integer δ_v, which denotes the amount of time instants required by a wireless station to transmit the value.

A system term W is a collection of individual threads running in parallel, with possibly some channels restricted. Note that the definition of the restriction operator $\nu c{:}(n, v).W$ is non-standard, for a restricted channel has a positive integer and a closed value associated; roughly speaking, the term $\nu c{:}(n, v).W$ corresponds to the term W where channel c is restricted, and the transmission of value v over channel c will take place for the next n instants of time.

Each thread may be inactive or active, of the form $c[x].P$, where x ranges over some unspecified set of data-variables; this process represents a wireless station in the process of 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 syntax for station code is based on standard process calculus constructs. This assumes some set of recursion variables ranged over by X, for use in the recursive definitions $\mathsf{fix}\, X.P$. The main constructs are time-dependent reception from a channel $\lfloor c?(x).P \rfloor Q$, an explicit time delay

Table 2 Wireless systems: station code

P, Q ::=	$c\,!\langle u\rangle.P$	broadcast
	$\lfloor c?(x).P\rfloor Q$	receiver with timeout
	$\sigma.P$	delay
	$\tau.P$	internal activity
	$P + Q$	choice
	$[b]P, Q$	matching
	X	process variable
	nil	termination
	$\mathsf{fix}\, X.P$	recursion

$\sigma.P$, and broadcast along a channel $c\,!\langle u\rangle.P$. Although in principle we could consider a sub-language for data manipulation, for the examples consider in this paper this is unnecessary; so in the broadcast construct u can be taken to be either a variable x or an actual data value. 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 currently used for transmitting.

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, $\{Q/X\}P$ and $\{^v/_x\}P$ respectively; we ignore the details. We simply assume that all occurrences of variables in system terms are bound and we will identify systems up to α-conversion. Moreover we assume 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 use a number of notational conventions. $\prod_{i\in I} W_i$ means the parallel composition of all systems W_i, for $i \in I$. We identify $\prod_{i\in I} W_i$ with nil if $I = \emptyset$. We will omit trailing occurrences of nil, render $\nu c : (n, v).W$ as $\nu c.W$ when the values (n, v) are not relevant to the discussion, and use $\nu\tilde{c}.W$ as an abbreviation for a sequence $\tilde{c}$ of such restrictions. We write $\lfloor c?(x).P\rfloor$ for $\lfloor c?(x).P\rfloor\mathsf{nil}$. Finally, we abbreviate the recursive process $\mathsf{fix}\, X.\lfloor c?(x).P\rfloor X$ with $c?(x).P$; as we will see this is a persistent listener at channel c waiting for an incoming message.

A channel environment is adequately represented as a function from channel names to pairs (n, v) where $n \in \mathbb{N}^\infty$ and v is a value. Intuitively $\Gamma(c) = (n, v)$ means that the network is in the process of transmitting the value v along the channel c, and the transmission will be complete in n more time units. We will use some suggestive notation for looking up the current state of a channel environment in a network:

- $\Gamma \vdash_t c : n$ 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.

Intuitively, a channel is *exposed* when it is currently used for transmitting (at least) a value, that is $\Gamma \vdash_t c : n$ for some $n > 0$. Channel exposure plays a major role in our semantics, and to emphasise this we use the suggestive notation $\Gamma \vdash c : \mathbf{exp}$; the converse, when c is ready for a transmission, will be denoted by $\Gamma \vdash c : \mathbf{idle}$. A channel environment Γ is said to be *stable* if $\Gamma \vdash c : \mathbf{idle}$ for every channel c. We also write $\Gamma \leq \Gamma'$ if $\Gamma \vdash_t c : n$ implies $\Gamma' \vdash_t c : n'$ and $n \leq n'$, for every channel c.

2.2 Intensional semantics

Our intention is to formally define one computation step of the form $\Gamma_1 \triangleright W_1 \multimap \Gamma_2 \triangleright W_2$. In this section, we provide an intensional semantics for system terms, where judgements take the form $\Gamma_1 \triangleright W_1 \xrightarrow{\lambda} W_2$. Here λ is an action corresponding to either the broadcast or reception of a value, time passing or internal activity. Then we show how a computation step of the configuration $\Gamma_1 \triangleright W_1$ affects the channel environment, leading to Γ_2. Evolutions of channel environments and system terms in a configuration will be combined in the next section to give computation steps for configurations.

We define our intensional semantics on station code using the following four judgements:

(1) $\Gamma \triangleright W \xrightarrow{c!v} W'$: the system W broadcasts the value v along channel c, resulting in residual W'

(2) $\Gamma \triangleright W \xrightarrow{\sigma} W'$: the passage of time transforms the system W into W'

(3) $\Gamma \triangleright W \xrightarrow{\tau} W'$: an internal action transforms W into W'

(4) $\Gamma \triangleright W \xrightarrow{c?v} W'$: the effect of the transmission of v along the channel c (by some unknown entity) transforms W into W'; this relation is used primarily to make the definition of the judgements in (1) more understandable.

In the sequel we use λ as a meta-variable ranging over the intensional action labels τ, $c!v$, $c?v$ and σ.

Table 3 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 is activated. 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 $c[x].P$.

If the channel is already exposed then by rule (RcvFail) the transmission is ignored and the reception is doomed to fail. This rule reflects the fact that, in general, collisions can be detected only at the end of a transmission. Although there are some protocols which allow a station to discover prematurely if a transmission is colliding with another one, for engineering reasons this is rarely done in wireless networks; see [45], page 186 for a discussion.

Table 3 Intensional semantics: transmission

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

$$\text{(RcvFail)}\ \frac{\Gamma \vdash c : \mathbf{exp}}{\Gamma \triangleright W \xrightarrow{c?v} \Gamma \triangleright W} \qquad \text{(RcvIgn)}\ \frac{\neg \mathsf{rcv}(W, c)}{\Gamma \triangleright W \xrightarrow{c?v} W}$$

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

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

$$\begin{aligned}
\mathsf{rcv}(\lfloor c?(x).P \rfloor Q, c) &= \text{true} \\
\mathsf{rcv}(P, c) = \text{true} \quad &\text{implies} \quad \mathsf{rcv}(P + Q, c) = \text{true} \\
\mathsf{rcv}(Q, c) = \text{true} \quad &\text{implies} \quad \mathsf{rcv}(P + Q, c) = \text{true} \\
\mathsf{rcv}(P, c) = \text{true} \quad &\text{implies} \quad \mathsf{rcv}(\mathsf{fix}\, X.P, c) = \text{true} \\
\\
\mathsf{rcv}(W_1, c) = \text{true} \quad &\text{implies} \quad \mathsf{rcv}(W_1 \mid W_2, c) = \text{true} \\
\mathsf{rcv}(W_2, c) = \text{true} \quad &\text{implies} \quad \mathsf{rcv}(W_1 \mid W_2, c) = \text{true} \\
\mathsf{rcv}(W, c) = \text{true} \quad &\text{implies} \quad \mathsf{rcv}(\nu d.W, c) = \text{true provided } d \neq c
\end{aligned}$$

The remaining two rules in Table 3 (Sync) and (RcvPar) serve to synchronise parallel stations on the same transmission [16, 35, 36].

The transitions for modelling the passage of time, $\Gamma \triangleright W \xrightarrow{\sigma} W'$, are given in Table 4. In the rules (ActRec) 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 *time determinism*. 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 in Table 5.

This table 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 channel c. However if c is exposed 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 the rule (RcvLate) reflects

Table 4 Intensional semantics: timed transitions

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

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

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

Table 5 Intensional semantics: - internal activity

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

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

the fact that in wireless systems a collision takes place if there is a misalignment between the transmission and reception of a message. The remaining rules are straightforward. Note that in the matching construct we use a channel environment dependent evaluation function for Boolean expressions $[\![b]\!]_\Gamma$, because of the presence of the exposure predicate $\mathrm{exp}(c)$ in the Boolean language. However checking for the exposure of a channel amounts to listening on the channel for a value. But in wireless systems it is not possible to both listen and transmit within the same time unit, as communication is half-duplex, [38]. As a consequence in our intensional semantics, in the rules (Then) and (Else), the execution of both branches is delayed of one time unit.

The final set of rules, in Table 6, are structural. In particular (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 a function $\mathrm{upd}_{c!v}(\cdot)$ which is defined later in this section, in Definition 2.4.

Let us provide some results which illustrate the intensional semantics. The first says that transmissions are non-blocking actions as stations can always synchronise on a transmission at channel c by performing the action $c?v$; in the terminology of [26], all systems are *input-enabled*.

Table 6 Intensional semantics: - structural rules

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

$$\text{(Rec)}\ \frac{\{\mathsf{fix}\, X.P/X\}P \xrightarrow{\lambda} W}{\Gamma \triangleright \mathsf{fix}\, X.P \xrightarrow{\lambda} W} \qquad \text{(Sum)}\ \frac{\Gamma \triangleright P \xrightarrow{\lambda} W \quad \lambda \in \{\tau, c!v\}}{\Gamma \triangleright P + Q \xrightarrow{\lambda} W}$$

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

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

Lemma 2.1 [Receive enabled] Let $\Gamma \triangleright W$ be a configuration. Then for any channel c and value v it holds $\Gamma \triangleright W \xrightarrow{c?v} W'$ for some W'; further

1. $\neg\mathsf{rcv}(W, c)$ implies $W' = W$
2. $\Gamma \vdash c : \mathbf{exp}$ implies $W' = W$.
3. $\mathsf{rcv}(W, c)$ and $\Gamma \vdash c : \mathbf{idle}$ implies $W' \neq W$, and for every value w $\Gamma \triangleright W \xrightarrow{c?w} W'$.

Proof By transition induction and by inspection of the rules (RcvIgn), (Rcv), (RcvFail) and (RcvPar). □

We can also show that our model of time conforms to a well-established approach in the literature; see for example [35, 47]:

Proposition 2.2

Time determinism: Suppose $C \xrightarrow{\sigma} W_1$ and $C \xrightarrow{\sigma} W_2$. Then $W_1 = W_2$.

Maximal Progress: Suppose $C \xrightarrow{\sigma} W_1$. Then $C \xrightarrow{\lambda} \not\rightarrow$ for $\lambda = \tau$, $c!v$ for any c, v, W_2.

Proof Both are proved by induction on the derivation of $C \xrightarrow{\sigma} W_1$. □

We end our discussion on the intensional semantics with a technical result on the interaction between stations in systems; this will be useful in later developments.

Proposition 2.3 [Parallel components] Let $\Gamma \triangleright W_1 \mid W_2$ be a configuration.

1. $\Gamma \triangleright W_1 \mid W_2 \xrightarrow{\tau} W$ if and only if

- either there is W_1' such that $\Gamma \triangleright W_1 \xrightarrow{\tau} W_1'$ with $W = W_1' \mid W_2$
- or there is W_2' such that $\Gamma \triangleright W_2 \xrightarrow{\tau} W_2'$ with $W = W_1 \mid W_2'$.

2. $\Gamma \triangleright W_1 \mid W_2 \xrightarrow{c?v} W$ if and only if there are W_1' and W_2' such that $\Gamma \triangleright W_1 \xrightarrow{c?v} W_1'$, $\Gamma \triangleright W_2 \xrightarrow{c?v} W_2'$ and $W = W_1' \mid W_2'$.

3. $\Gamma \triangleright W_1 \mid W_2 \xrightarrow{c!v} W$ if and only if there are W_1' and W_2' such that

   - $\Gamma \triangleright W_1 \xrightarrow{c!v} W_1'$, $\Gamma \triangleright W_2 \xrightarrow{c?v} W_2'$ and $W = W_1' \mid W_2'$
   - or $\Gamma \triangleright W_1 \xrightarrow{c?v} W_1'$, $\Gamma \triangleright W_2 \xrightarrow{c!v} W_2'$ and $W = W_1' \mid W_2'$.

4. $\Gamma \triangleright W_1 \mid W_2 \xrightarrow{\sigma} W$ if and only if there are W_1' and W_2' such that $\Gamma \triangleright W_1 \xrightarrow{\sigma} W_1'$, $\Gamma \triangleright W_2 \xrightarrow{\sigma} W_2'$ and $W = W_1' \mid W_2'$. □

Let us now turn our attention to the evolution of channel environments in a configuration. Here we define a predicate $\mathrm{upd}_\lambda(\Gamma)$ which describes how the channel environment Γ changes when performing the action λ.

Definition 2.4 [Channel Environment update] Let Γ be an arbitrary channel environment and λ an intensional action. We let $\mathrm{upd}_\lambda(\Gamma)$ be the unique channel environment determined by the following four definitions:[2]

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

We let $\mathrm{upd}_{c!v}(\Gamma)$ be the channel environment such that

$$\mathrm{upd}_{c!v}(\Gamma) \vdash_t c = \begin{cases} \delta_v & \text{if } \Gamma \vdash c : \mathbf{idle} \\ j & \text{if } \Gamma \vdash c : \mathbf{exp} \end{cases} \qquad \mathrm{upd}_{c!v}(\Gamma) \vdash_v c = \begin{cases} v & \text{if } \Gamma \vdash c : \mathbf{idle} \\ \mathsf{err} & \text{if } \Gamma \vdash c : \mathbf{exp} \end{cases}$$

where $\Gamma \vdash_t c : k$ and j is given by $\max(\delta_v, k)$. Finally, we let $\mathrm{upd}_{c?v}(\Gamma) = \mathrm{upd}_{c!v}(\Gamma)$ and $\mathrm{upd}_\tau(\Gamma) = \Gamma$.

The definition of $\mathrm{upd}_\sigma(\Gamma)$ is straightforward; when time passes, the time of exposure of each channel decreases by one time unit. The predicates $\mathrm{upd}_{c!v}(\Gamma)$ and $\mathrm{upd}_{c?v}(\Gamma)$ 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. Finally the definition of $\mathrm{upd}_\tau(\Gamma)$ reflects the intuition that internal activities do not affect the exposure state of channels.

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

2.3 Reduction semantics

We are now in a position to formally define the individual computation steps for wireless systems, alluded to informally in (1) above.

Definition 2.5 [Reduction] We write $\Gamma \triangleright W \twoheadrightarrow \Gamma' \triangleright W'$ if

(i) (Transmission) $\Gamma \triangleright W \xrightarrow{c!v} W'$ for some channel c and value v, where $\Gamma' = \mathrm{upd}_{c!v}(\Gamma)$

(ii) (Time) $\Gamma \triangleright W \xrightarrow{\sigma} W'$ and $\Gamma' = \mathrm{upd}_{\sigma}(\Gamma)$

(iii) or (Internal) $\Gamma \triangleright W \xrightarrow{\tau} W'$ and $\Gamma' = \mathrm{upd}_{\tau}(\Gamma)$. □

The intuition here should be obvious; computation proceeds either by the transmission of values between stations, the passage of time, or internal activity; further, the exposure state of channels is updated according to the performed transition.

Sometimes it will be useful to distinguish between instantaneous reductions and timed reductions; instantaneous reductions, $\Gamma_1 \triangleright W_1 \twoheadrightarrow_i \Gamma_2 \triangleright W_2$, are those derived via clauses (i) or (iii) above; timed reductions are denoted with the symbol $\twoheadrightarrow_\sigma$ and coincide with reductions derived using clause (ii).

Example 2.6 [Time-consuming Transmission] Consider a wireless system with two stations, that is a configuration C_1 of the form $\Gamma_1 \triangleright P_1 \mid Q_1$. Let us suppose

$$P_1 \text{ is } c\,!\langle w\rangle.R, \qquad Q_1 \text{ is } \lfloor c?(x).S \rfloor T_1$$

and Γ_1 is a stable channel environment, in that $\Gamma_1 \vdash_t d : 0$ for all channels d. Then, assuming δ_w is 2,

$$C_1 \twoheadrightarrow C_2 \tag{2}$$

where C_2 has the form $\Gamma_2 \triangleright P_2 \mid Q_2$ and

$$P_2 \text{ is } \sigma^2.R \qquad Q_2 \text{ is } c[x].S \qquad \Gamma_2 \vdash_t c : 2 \qquad \Gamma_2 \vdash_v c : w$$

The move from P_1 to P_2 is via an application of the rule (Snd), from Q_1 to Q_2 relies on (Rcv) and they are combined together using (Sync) to obtain $\Gamma_1 \triangleright P_1 \mid Q_1 \xrightarrow{c!w} P_2 \mid Q_2$ and then the step (2) results from (Transmission) in Definition 2.5.

The next step $C_2 \twoheadrightarrow C_3$ is via (Time) in Definition 2.5; C_3 is of the form $\Gamma_3 \triangleright \sigma.R \mid Q_2$ where the only change to the channel environment is that now $\Gamma_3 \vdash_t c : 1$. The inference of the transition

$$\Gamma_2 \triangleright P_2 \mid Q_2 \xrightarrow{\sigma} \sigma.R \mid Q_2$$

uses the rules (Sleep), (ActRec) and (TimePar).

The final move we consider, $C_3 \twoheadrightarrow C_4 = \Gamma \triangleright R \mid \{^w/_x\}S$, is another instance of (Time). However here the delay action is inferred using (Sleep), (EndRcv) and (TimePar). Thus in three reduction

steps the value w has been transmitted from the first station to the second one along the channel c, in two units of time.

Now suppose we change P_1 to $P_1' = \sigma.P_1$, obtaining thus the configuration $C_1' = \Gamma_1 \triangleright P_1' \mid Q_1$. Then the first step, $C_1' \twoheadrightarrow C_2'$ is a (Time) step, with $C_2' = \Gamma_1 \triangleright P_1 \mid T_1$. Here an instance of the rule (Timeout) is used in the transition from Q_1 to T_1. In C_2' the station P_1 is now ready to transmit on channel c, but the second station has stopped listening. The next step depends on the exact form of T_1; if for example $\mathsf{rcv}(T_1, c)$ is false then by an application of rule (RcvIgn) we can derive $C_2' \twoheadrightarrow C_3' = \Gamma_2 \triangleright P_2 \mid T_1$. Here the transmission of w along c started but nobody was listening.

Finally, suppose T_1 is a delayed listener on channel c, say $\sigma.T_2$ where T_2 is $\lfloor c?(y).S_2 \rfloor U_2$. Then we have the (Time) step $C_3' \twoheadrightarrow C_4' = \Gamma_3 \triangleright \sigma.R \mid T_2$ and now the second station, T_2, is ready to listen. However, as $\Gamma_3 \vdash c : \mathbf{exp}$, station T_2 is joining the transmission too late. To reflect this we can derive we can derive the (Internal) step

$$C_4' \twoheadrightarrow C_5' = \Gamma_3 \triangleright \sigma.R \mid c[y].\{^{\mathsf{err}}/_y\}S_2$$

using the rules (RcvLate) and (TauPar), amongst others. At the end of the transmission, in one more time step, the second station will therefore end up with an error in reception.

In the revised system $C_1' = \Gamma_1 \triangleright \sigma.P_1' \mid Q_1$ the second station missed the delayed transmission from P_1'. However we can change the code at the second station to accommodate this delay, by replacing Q_1 with the persistent listener $Q_1' = c?(x).S$. We leave the reader to check that starting from the configuration $\Gamma_1 \triangleright \sigma.P_1' \mid Q_1'$ the value w will be successfully transmitted between the stations in four reduction steps. □

Example 2.7 [Collisions] Let us now consider a system with three stations,

$$C_1 = \Gamma \triangleright S_1 \mid S_2 \mid R_1$$

where

$$\begin{aligned} S_1 &= \sigma.c!\langle v_0 \rangle \\ S_2 &= c!\langle v_1 \rangle \\ R_1 &= \lfloor c?(x).P \rfloor T \end{aligned}$$

and Γ is a stable environment; suppose further that $\delta_{v_0} = 1$, $\delta_{v_1} = 3$. In this configuration the station S_2 can perform a broadcast, leading to the reduction $C_1 \twoheadrightarrow C_2 = \Gamma_1 \triangleright S_1 \mid \sigma^3 \mid c[x].P$, the derivation of which requires an instance of the rule (RcvIgn), $\Gamma \triangleright S_1 \xrightarrow{c?v_1} S_1$; here the channel environment Γ_1 is defined as $\mathrm{upd}_{c!v_1}(\Gamma)$, leading to $\Gamma_1(c) = (3, v_1)$. We can now derive the reduction $C_2 \twoheadrightarrow C_3 = \Gamma_2 \triangleright c!\langle v_0 \rangle \mid \sigma^2 \mid c[x].P$, where $\Gamma_2 = \mathrm{upd}_{\sigma}(\Gamma_1)$ meaning that $\Gamma_2 \vdash_t c : 2$.

In this configuration the first station is ready to broadcast a value v_0 along channel c. Since there is already a value being transmitted along this channel, we expect this second broadcast to cause a collision; further, since the amount of time required for transmitting value v_0 is shorter than that needed to end the transmission of value v_1, we expect that the broadcast performed by the first station does not affect the amount of time for which the channel C is exposed.

Formally this is reflected in the reduction $C_3 \twoheadrightarrow C_3' = \Gamma_2' \triangleright \sigma \mid \sigma^2 \mid c[x].P$. Here the reduction of the system term uses the sub-inference $\Gamma_2 \triangleright \sigma^2 \mid c[x].P \xrightarrow{c?v_1} \sigma^2 \mid c[x].P$, which can be derived using either rules (RcvFail), (RcvIgn). Consequently $\Gamma_2' = \mathrm{upd}_{c!v_0}(\Gamma_2)$, and since $\Gamma_2 \vdash c : \mathbf{exp}$ we obtain $\Gamma_2'(c) = (2, \mathsf{err})$; this represents the fact that a collision has occurred, and thus the special value err will eventually be delivered on c.

At this point we can derive the reductions $C_3' \twoheadrightarrow_\sigma \twoheadrightarrow_\sigma C_4 = \Gamma \triangleright \mathsf{nil} \mid \mathsf{nil} \mid \{\mathsf{err}/x\}P$, meaning that the transmission along channel c terminates in 2 time instants, leading the receiving station to detect a collision each of the two transitions above can be inferred using the rules in Table 4.

Now, suppose we change the amount of time required to transmit value v_0 from 1 to 3, and consider again the configuration C_3 above. In this case the transmission of value v_0 will also cause a collision; however, in this case the transmission of value v_0 is long enough to continue after that of value v_1 has finished; as a consequence, we expect that the time required for channel c to be released rises when the broadcast of v_0 happens.

In fact, in this case we have the reduction $C_3 \twoheadrightarrow C_3'' = \Gamma_2'' \triangleright \sigma^3 \mid \sigma^2 \mid c[x].P$, where $\Gamma_2'' = \mathrm{upd}_{c!v_0}(\Gamma_2)$ and specifically $\Gamma_2''(c) = (3, \mathsf{err})$. Now, three time instants are needed for the transmission along channel c to end, leading to the sequence of (timed) reductions $C_3'' \twoheadrightarrow_\sigma \twoheadrightarrow_\sigma \twoheadrightarrow_\sigma C_4$. □

2.4 Behavioural semantics

In this section we propose a notion of timed behavioural equivalence for our wireless networks. Our touchstone system equality is *reduction barbed congruence* [19, 42, 30, 21], a standard contextually defined process equivalence. Intuitively, two terms are reduction barbed congruent if they have the same *basic observables*, in all parallel contexts, under all possible *computations*. The formal definition relies on two crucial concepts, a reduction semantics to describe how systems evolve, which we have already defined, and a notion of basic observable which says what the environment can observe directly of a system. There is some choice as to what to take as a basic observation, or *barb*, of a wireless system. In standard process calculi this is usually taken to be the ability of the environment to receive a value along a channel. But the series of examples we have just seen demonstrates that this problematic, in the presence of possible collisions and the passage of time. Instead we choose a more appropriate notion for wireless systems, one which is already present in our language for station code.

Definition 2.8 [Barbs] We say the configuration $\Gamma \triangleright W$ has a *strong barb on* c, written $\Gamma \triangleright W \downarrow_c$, if $\Gamma \vdash c : \mathbf{exp}$. We write $\Gamma \triangleright W \Downarrow_c$, a *weak barb*, if there exists a configuration C' such that $\Gamma \triangleright W \twoheadrightarrow^* C'$ and $C' \downarrow_c$. Note that we allow the passage of time in this definition of weak barbs. □

Definition 2.9 Let $\mathcal{R}$ be a relation over well-formed 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 $\Gamma_1 \triangleright W_1 \twoheadrightarrow \Gamma_1' \triangleright W_1'$ imply there is some $\Gamma_2' \triangleright W_2'$ such that $\Gamma_2 \triangleright W_2 \twoheadrightarrow^* \Gamma_2' \triangleright W_2'$ and $(\Gamma_1' \triangleright W_1')\ \mathcal{R}\ (\Gamma_2' \triangleright W_2')$.

(3) It is *contextual* if $\Gamma_1 \triangleright W_1 \ \mathcal{R} \ \Gamma_2 \triangleright W_2$, implies $\Gamma_1 \triangleright (W_1 \mid W) \ \mathcal{R} \ \Gamma_2 \triangleright (W_2 \mid W)$ for all processes W. □

With these concepts we now have everything in place for a standard definition of contextual equivalence between systems:

Definition 2.10 [Reduction barbed congruence], written $\simeq$, is the largest symmetric relation over configurations which is barb preserving, reduction-closed and contextual. □

In the remainder of this section we explore via examples the implications of Definition 2.10. The notion of a fresh channel will be important; we say that c is *fresh* for the configuration $\Gamma \triangleright W$ if it does not occur free in W and $\Gamma \vdash c : \mathbf{idle}$. Note that we can always pick a fresh channel for an arbitrary configuration.

Example 2.11 Let us assume that $\Gamma \vdash c : \mathbf{idle}$. Then it is easy to see that

$$\Gamma \triangleright c\,!\langle v_o\rangle.P \ \not\simeq \ \Gamma \triangleright c\,!\langle v_1\rangle.P \tag{3}$$

under the assumption that v_o and v_1 are different values. For let T be the testing context

$$\lfloor c?(x).[x = v_o]\mathit{eureka}!\langle \mathit{ok}\rangle, \mathsf{nil}\rfloor$$

where *eureka* is fresh, and *ok* is some arbitrary value. Then $\Gamma \triangleright c\,!\langle v_o\rangle.P \mid T$ has a weak barb on *eureka* which is not the case for $\Gamma \triangleright c\,!\langle v_1\rangle.P \mid T$. Since $\simeq$ is contextual and barb preserving, (3) above follows.

However such tests will not distinguish between

$$Q_1 = c!\langle v_o\rangle \mid c\,!\langle v_1\rangle.P \qquad \text{and} \qquad Q_2 = c!\langle v_1\rangle \mid c\,!\langle v_0\rangle.P$$

under the assumption that $\delta_{v_0} = \delta_{v_1}$. In both configurations $\Gamma \triangleright Q_1$ and $\Gamma \triangleright Q_2$ a collision will occur at channel c and a third station, such as T, will receive the error value err at the end of the transmission. So there is reason to hope that $\Gamma \triangleright Q_1 \ \simeq \ \Gamma \triangleright Q_2$. However we must wait for the next section for proof techniques for establishing such equivalences; see Example 3.4. □

The above example suggests that transmitted values can be observed only at the end of a transmission; so if a collision happens, there is no possibility of determining the value that was originally broadcast. This concept is stressed even more in the following example.

Example 2.12 [Equating values] Let Γ be a stable channel environment, $W_0 = c!\langle v_0\rangle$, $W_1 = c!\langle v_1\rangle$ and consider the configurations $\Gamma \triangleright W_0, \Gamma \triangleright W_1$; here we assume that v_0 and v_1 are two different values with possibly different transmission times.

We already argued in Example 2.11 that these two configurations can be distinguished by the context

$$\lfloor c?(x).[x = v_o]\mathit{eureka}!\langle \mathit{ok}\rangle, \mathsf{nil}\rfloor$$

However, the two configurations above can be made indistinguishable if we add to each of them a parallel component that causes a collision on channel c. To this end, let

$$Eq(v_0, v_1) = \sigma^h.c!\langle ok\rangle$$

for some positive integer h and value ok such that $h < \min(\delta_{v_0}, \delta_{v_1})$ and $\delta_{ok} \geq \max(\delta_{v_0}, \delta_{v_1}) - h$. Now, consider the configurations $C_0 = \Gamma \triangleright W_0 \mid Eq(v_0, v_1)$, $C_1 = \Gamma \triangleright W_1 \mid Eq(v_0, v_1)$.

One could hope that there exists a context which is able to distinguish these two configurations. However, before the transmission of v_0 ends in C_0, a second broadcast along channel c will fire, causing a collision; the same happens before the end of transmission of value v_1 in C_1. Further, the total amount of time for which channel c will be exposed is the same for both configurations, so that one can argue that it is impossible to provide a context which is able to distinguish C_0 from C_1. In order to prove this to be formally true, we have to wait until the next section. □

Collisions can also be used to merge two different transmissions on the same channel in a single corrupted transmission.

Example 2.13 [Merging Transmissions] Let Γ be a stable channel environment, $W_0 = c!\langle v_0\rangle.c!\langle v_1\rangle$, $W_1 = c!\langle v_1\rangle.c!\langle v_0\rangle$. In $\Gamma \triangleright W_0$ a broadcast of value v_0 along channel c can fire; when the transmission of v_0 is finished, a second broadcast of value v_1 along the same channel can fire. The behaviour of $\Gamma \triangleright W_1$ is similar, though the order of the two values to be broadcast is swapped. Note that it is possible to distinguish the two configurations $\Gamma \triangleright W_0$ and $\Gamma \triangleright W_1$ using the test

$$\lfloor c?(x).[x = v_o]eureka!\langle ok\rangle, \mathsf{nil}\rfloor$$

we have already seen in the previous example.

However suppose now that we add a parallel component to both configurations which broadcasts another value along channel c before the transmission of value v_0 (v_1) has finished, and which terminates after the broadcast of value v_1 (v_0) has begun. More formally, let

$$Mrg(v_0, v_1) = \sigma^h.c!\langle ok\rangle$$

where $h = \min(\delta_{v_0}, \delta_{v_1}) - 1$ and $\delta_{ok} = |\delta_{v_0} - \delta_{v_1}| + 2$.

Consider the configurations $\Gamma \triangleright W_0 \mid Mrg(v_0, v_1)$, $\Gamma \triangleright W_1 \mid Mrg(v_0, v_1)$. In both configurations a collision occurs; further, once the transmission of value v_0 has begun in the former configuration, channel c will remain exposed until the transmission of value v_1 has finished. A similar behaviour can be observed on the second configuration. This leads to the intuition that $\Gamma \triangleright W_0 \mid Mrg(v_0, v_1) \simeq \Gamma' \triangleright W_1 \mid Mrg(v_0, v_1)$; we prove this in Example 3.6, for a particular instance of transmission values for v_0, v_1. □

A priori reductions ignore the passage of time, and therefore one might suspect that reduction barbed congruence is impervious to the precise timing of activities. But the next example demonstrates that this is not the case.

Example 2.14 [Observing the passage of time] Consider the two processes $Q_1 = c!\langle v_o\rangle$ and $Q_2 = \sigma.Q_1$, and again let us assume that $\Gamma \vdash c : \mathbf{idle}$. There is very little difference between the behaviours of $\Gamma \triangleright Q_1$ and $\Gamma \triangleright Q_2$; both will transmit (successfully) the value v_o, although the latter is a little slower. However this slight difference can be observed. Consider the test T defined by

$$[\exp(c)]\mathit{eureka}!\langle \mathit{ok}\rangle, \mathsf{nil}$$

In fact, $\Gamma \triangleright (Q_1 \mid T)$ can start a transmission along channel c, after which the predicate $\exp(c)$ will be evaluated in the system term T. The resulting configuration is given by $\Gamma' \triangleright \sigma^{\delta_{v_0}} \mid \sigma.\mathit{eureka}!\langle \mathit{ok}\rangle$; at this point, it is not difficult to note that the configuration has a weak barb on *eureka*.

On the other hand, the unique reduction from $C_2 = \Gamma \triangleright (Q_2 \mid T)$ leads to the evaluation of the exposure predicate $\exp(c)$; since $\Gamma \vdash c : \mathbf{idle}$ the only possibility for the resulting configuration is given by $C_2' = \Gamma \triangleright Q_2 \mid \sigma$. Since *eureka* is a fresh channel, it is now immediate to note that $C_2' \not\Downarrow_{\mathit{eureka}}$. For the test to work correctly it is essential that $\Gamma \vdash c : \mathbf{idle}$; using the proof methodology developed in Section 3.2 we are able to show that if $\Gamma' \vdash_t c : n$ and $n > \delta_{v_0}$ then $\Gamma' \triangleright Q_1 \simeq \Gamma' \triangleright Q_2$. □

Behind this example is the general principle that reduction barbed congruence is actually sensitive to the passage of time; this is proved formally in Proposition 4.17 of Section 4.3.

Example 2.15 As a final example we illustrate the use of channel restriction. Assume that v_1 and v_2 are some kind of comparable values. Consider the configuration
$\Gamma \triangleright \nu c : (0, \cdot).(c!\langle v_1\rangle \mid P_e \mid R)$ where the station code is given by

$$\begin{aligned} P_e &= \sigma.\mathsf{fix}\, X.([\exp(c)]X, c!\langle v_2\rangle) \\ R &= c?(x).R_1 \\ R_1 &= c?(y).[x > y]d!\langle x\rangle, d!\langle y\rangle \end{aligned}$$

Intuitively the receiver R waits indefinitely for two values along the restricted channel c and broadcasts the largest on channel d. Intuitively the use of channel restriction here shelters c from external interference. Assuming $\Gamma \vdash d : \mathbf{idle}$ we will be able to show that

$$\Gamma \triangleright \nu c : (0, \cdot).(c\,!\langle v_1\rangle.\mathsf{nil} \mid P_e \mid R) \quad \simeq \quad \Gamma \triangleright \sigma^{\delta_{v_1}+\delta_{v_2}+2}.d\,!\langle w\rangle.\mathsf{nil}$$

provided $w = \max(v_1, v_2)$. □

We end this section with a small technical result, which will be extremely useful in the development of our behavioural theory. Informally it says that internal reductions do not affect the remaining delivery time of values.

Lemma 2.16 Whenever $\Gamma \triangleright W \rightarrow\!\triangleright_i^* \Gamma' \triangleright W'$ it holds that $\Gamma \leq \Gamma'$.

Proof Follows from the definition of $\mathrm{upd}_{c!v}(\Gamma)$ and $\mathrm{upd}_{\tau}(\Gamma)$ □

Table 7 Extensional actions

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

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

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

3 Extensional Semantics

The intention here is to give a co-inductive characterisation of the contextual equivalence $\simeq$ between configurations, in terms of a standard bisimulation equivalence over some extensional LTS. First, we present the extensional semantics, then we recall the standard definition of (weak) bisimulation over configurations; finally its usefulness is illustrated by means of a number of examples.

3.1 Extensional actions

The question here is what activity of a wireless system is observable externally? Example 2.14 indicates that the passage of time is observable. From Lemma 2.1, we know that all systems are always ready to receive transmissions, that is are input-enabled, but we will have to take into account the effect of these inputs. In contrast the discussion in Example 2.11 indicates that, due to the possibility of collisions, the treatment of transmissions is more subtle. It turns out that the transmission itself is not important; instead we must take into consideration the successful delivery of the transmitted value.

In Table 7 we give the rules defining the extensional actions, $C \overset{\alpha}{\longmapsto} C'$, which can take one of the forms:

- Input: $C \overset{c?v}{\longmapsto} C'$. This is inherited directly from the intensional semantics.
- Time: $C \overset{\sigma}{\longmapsto} C'$, also inherited from the intensional semantics.
- Internal: $C \overset{\tau}{\longmapsto} C'$. This corresponds to the combination of the Internal and Transmission rules from the reduction semantics, in Definition 2.5.
- Delivery: $C \overset{\gamma(c,v)}{\longmapsto} C'$. This corresponds to the successful delivery of the value v which was in transmission along the channel c.

- Free: $C \overset{\iota(c)}{\longmapsto} C$, a predicate indicating that channel c is not exposed, and therefore ready to start a potentially successful transmission.

3.2 Bisimulation equivalence

The extensional actions of the previous section endows systems in CCCP with the structure of an LTS. Weak extensional actions in this LTS are defined as usual, with $C \overset{\alpha}{\Longmapsto} C'$ denoting $C \overset{\tau}{\longmapsto}{}^* \overset{\alpha}{\longmapsto} \overset{\tau}{\longmapsto}{}^* C'$. We will use $C \Longmapsto C'$ to denote $C \overset{\tau}{\longmapsto}{}^* C'$, 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 $C \overset{\alpha}{\Longmapsto} C'$. We now have the standard definition of weak bisimulation equivalence in the resulting LTS which for convenience we recall.

Definition 3.1 Let $\mathcal{R}$ be a binary relation over configurations. We say that $\mathcal{R}$ is a bisimulation if for every extensional action α, whenever $C_1 \mathcal{R} C_2$

(i) $C_1 \overset{\alpha}{\longmapsto} C_1'$ implies $C_2 \overset{\hat{\alpha}}{\Longmapsto} C_2'$, for some C_2', satisfying $C_1' \mathcal{R} C_2'$

(ii) conversely, $C_2 \overset{\alpha}{\longmapsto} C_2'$ implies $C_1 \overset{\hat{\alpha}}{\Longmapsto} C_1'$, for some C_1', such that $C_1' \mathcal{R} C_2'$.

We write $C_1 \approx C_2$, if there is a bisimulation $\mathcal{R}$ such that $C_1 \mathcal{R} C_2$. □

Our goal is to demonstrate that this form of bisimulation provides a sound and useful proof method for showing behavioural equivalence between wireless systems described in CCCP; moreover for a large class of systems it will also turn out to be complete. But our first examples show that the introduction of the actions $\iota(c)$ and $\gamma(c, v)$ are necessary for soundness.

Example 3.2 [On the rule (Idle)] Suppose we were to drop the rule (Idle) in the extensional semantics; then consider the configurations

$$\begin{aligned} \Gamma_1 \triangleright W_1 &= \tau.\mathsf{nil} \\ \Gamma_2 \triangleright W_2 &= c!\langle v \rangle \end{aligned}$$

where $\Gamma_1(c) = (1, v)$, $\Gamma_2(c) = (0, \cdot)$ and $\delta_v = 1$.

If we were to drop the actions $\iota(c)$ from the extensional semantics then the extensional LTS generated by these two configurations would be isomorphic; recall that an output action in the intensional semantics always corresponds to a τ action in its extensional counterpart. Thus they would be related by the amended version of bisimulation equivalence.

However, we also have that $\Gamma_1 \triangleright W_1 \not\approx \Gamma_2 \triangleright W_2$. This can be proved by exhibiting a distinguishing context. To this end, consider the system $T = [\exp(c)]\mathsf{nil}, \mathrm{eureka}!\langle ok \rangle$. Then $\Gamma_2 \triangleright W_2 \mid T$ has a weak barb on the channel eureka, which obviously $\Gamma_2 \triangleright W_1 \mid T$ can not match. □

Example 3.3 [On the rule (Deliver)] Consider the configuration

$$\Gamma_2 \triangleright W_3 = c!\langle w\rangle$$

where $\delta_w = 1$ and Γ_2 is as in the previous example, and the testing context $T' = c(x).[x = v]\text{eureka}!\langle ok\rangle, \mathsf{nil}$. Then, assuming w is different from v, $\Gamma_2 \triangleright W_3 \mid T'$ can not produce a barb on eureka.

However if W_2 is the code $c!\langle v\rangle$, as in the previous example, then obviously $\Gamma_2 \triangleright W_3 \mid T'$ can produce such a barb. It follows that $\Gamma_2 \triangleright W_2 \not\approx \Gamma_2 \triangleright W_3$.

Now if we were to drop the rule (Deliver) in the extensional semantics, thereby eliminating the actions $\gamma(c, v)$, then it would be straightforward to exhibit a bisimulation containing this pair of configurations. Thus again the amended version of bisimulation equivalence would be unsound.

□

The two examples above show that both rules (Idle) and (Deliver) are necessary to achieve the soundness of our bisimulation proof method for reduction barbed congruence. In the remainder of this section we give a further series of examples, showing that bisimulations in our extensional LTS offers a viable proof technique for demonstrating behavioural equivalence for at least simple wireless systems.

Example 3.4 [Transmission] Here we revisit Example 2.11. Let Γ be a stable channel environment, and consider the configurations $C_0 = \Gamma \triangleright W$, $C_1 = \Gamma \triangleright V$, where $W = c\,!\langle v_0\rangle.P \mid c!\langle v_1\rangle$, $V = c\,!\langle v_1\rangle.P \mid c!\langle v_0\rangle$; note these two configurations are taken from the second part of in Example 2.11.

Our aim is to show that $C_0 \approx C_1$, and for convenience let us we assume that $\delta_{v_0} = \delta_{v_1} = 1$. The idea here is to describe the required bisimulation by matching up system terms. To this end we define the following system terms:

$$\begin{array}{rclcrcl} W_0 & = & \sigma.P \mid c!\langle v_1\rangle & \qquad & V_1 & = & \sigma.P \mid c!\langle v_0\rangle \\ W_1 & = & c\,!\langle v_0\rangle.P \mid \sigma & & V_0 & = & c\,!\langle v_1\rangle.P \mid \sigma \\ E & = & \sigma.P \mid \sigma & & E' & = & P \mid \mathsf{nil} \end{array}$$

Then for any channel environment Δ we have the following transitions in the extensional semantics:

$$\begin{array}{lll} \Delta \triangleright W \overset{\tau}{\longmapsto} \mathrm{upd}_{c!v_0}(\Delta) \triangleright W_0 & \qquad & \Delta \triangleright V \overset{\tau}{\longmapsto} \mathrm{upd}_{c!v_0}(\Delta) \triangleright V_0 \\ \Delta \triangleright W \overset{\tau}{\longmapsto} \mathrm{upd}_{c!v_1}(\Delta) \triangleright W_1 & & \Delta \triangleright V \overset{\tau}{\longmapsto} \mathrm{upd}_{c!v_1}(\Delta) \triangleright V_1 \\ \Delta \triangleright W \overset{d?w}{\longmapsto} \mathrm{upd}_{d?w}(\Delta) \triangleright W & & \Delta \triangleright V \overset{d?w}{\longmapsto} \mathrm{upd}_{d?w}(\Delta) \triangleright V \\ \Delta \triangleright W \overset{\iota(d)}{\longmapsto} \Delta \triangleright W \text{ if } \Delta \vdash d : \mathbf{idle} & & \Delta \triangleright V \overset{\iota(d)}{\longmapsto} \Delta \triangleright V \text{ if } \Delta \vdash d : \mathbf{idle} \end{array}$$

$$\begin{array}{lll} \Delta \triangleright W_0 \overset{\tau}{\longmapsto} \mathrm{upd}_{c!v_1}(\Delta) \triangleright E & \qquad & \Delta \triangleright V_0 \overset{\tau}{\longmapsto} \mathrm{upd}_{c!v_1}(\Delta) \triangleright E \\ \Delta \triangleright W_0 \overset{d?w}{\longmapsto} \mathrm{upd}_{d?w}(\Delta) \triangleright W_0 & & \Delta \triangleright V_0 \overset{d?w}{\longmapsto} \mathrm{upd}_{d?w}(\Delta) \triangleright V_0 \\ \Delta \triangleright W_0 \overset{\iota(d)}{\longmapsto} \Delta \triangleright W_0 \text{ if } \Delta \vdash d : \mathbf{idle} & & \Delta \triangleright V_0 \overset{\iota(d)}{\longmapsto} \Delta \triangleright V_0 \text{ if } \Delta \vdash d : \mathbf{idle} \end{array}$$

Table 8 A relation $\mathcal{S}$ for comparing the configurations C_0, C_1 of Example 3.5

$$
\begin{array}{rcl}
\Delta \triangleright W & \mathcal{S} & \Delta \triangleright V \\
\Delta \triangleright W_0 & \mathcal{S} & \Delta \triangleright V_0 \\
(\Delta[c \mapsto (1, v_0)]) \triangleright W_0 & \mathcal{S} & (\Delta[c \mapsto (2, v_1)]) \triangleright V_1 \\
(\Delta[c \mapsto (1, \mathsf{err})]) \triangleright W_0 & \mathcal{S} & (\Delta[c \mapsto (2, \mathsf{err})]) \triangleright V_1 \\
\Lambda \triangleright W_{ok} & \mathcal{S} & \Lambda \triangleright V_{ok} \\
\Delta \triangleright W_{\mathsf{err}} & \mathcal{S} & \Delta \triangleright V_{\mathsf{err}} \\
\Delta \triangleright W' & \mathcal{S} & \Delta \triangleright V'
\end{array}
$$

Δ arbitrary channel environment,
Λ arbitrary channel environment such that $\Lambda(c) = (k, w)$ for some $k \geq 2$

$$
\begin{array}{lll}
\Delta \triangleright W_1 \overset{\tau}{\longmapsto} \mathrm{upd}_{c!v_0}(\Delta) \triangleright E & & \Delta \triangleright V_1 \overset{\tau}{\longmapsto} \mathrm{upd}_{c!v_0}(\Delta) \triangleright E \\
\Delta \triangleright W_1 \overset{d?w}{\longmapsto} \mathrm{upd}_{d?w}(\Delta) \triangleright W_1 & & \Delta \triangleright V_1 \overset{d?w}{\longmapsto} \mathrm{upd}_{d?w}(\Delta) \triangleright V_1 \\
\Delta \triangleright W_1 \overset{\iota(d)}{\longmapsto} \Delta \triangleright W_1 \text{ if } \Delta \vdash d : \mathbf{idle} & & \Delta \triangleright V_1 \overset{\iota(d)}{\longmapsto} \Delta \triangleright V_1 \text{ if } \Delta \vdash d : \mathbf{idle}
\end{array}
$$

Here d ranges over arbitrary channel names, including c.

Then consider the following relation:

$$\mathcal{S} = \{(\Delta \triangleright W, \Delta \triangleright V),\ (\Delta \triangleright W_0,\ \Delta \triangleright V_0), (\Delta \triangleright W_1, \Delta \triangleright V_1)\ \mid\ \Delta \text{ is a channel environment}\}$$

Using the above tabulation of actions one can now show that $\mathcal{S}$ is a *strong* bisimulation; for $C\,\mathcal{S}\,C'$ each possible action of C can be matched by C' by performing exactly the same action, and vice-versa.

Since $(C_0, C_1) \in \mathcal{S}$, it follows that $C_0 \approx C_1$. □

Example 3.5 [Equators] Let us consider again the configurations C_0, C_1 of Example 2.12. Recall that $C_0 = \Gamma \triangleright W$, where $W = c!\langle v_0\rangle \mid \sigma^h.c!\langle ok\rangle$ and $C_1 = \Gamma \triangleright V$, where $V = c!\langle v_1\rangle \mid \sigma^h.c!\langle ok\rangle$; further, recall that Γ is a stable channel environment and h, ok are a positive integer and a value, respectively, such that $h < \min(\delta_{v_0}, \delta_{v_1})$, $\delta_{ok} \geq \max(\delta_{v_0}, \delta_{v_1}) - h$. Without loss of generality, for this example we assume $\delta_{v_0} = 1, \delta_{v_1} = 2$, $h = 0$ and $\delta_{ok} = 2$.

For the sake of convenience we define the following system terms:

$$
\begin{array}{rclrcl}
W_0 & = & \sigma \mid c!\langle ok\rangle & V_1 & = & \sigma^2 \mid c!\langle ok\rangle \\
W_{ok} & = & c!\langle v_0\rangle \mid \sigma^2 & V_{ok} & = & c!\langle v_1\rangle \mid \sigma^2 \\
W_{\mathsf{err}} & = & \sigma \mid \sigma^2 & V_{\mathsf{err}} & = & \sigma^2 \mid \sigma^2 \\
W' & = & \mathsf{nil} \mid \sigma & V' & = & \sigma \mid \sigma \\
E & = & \mathsf{nil} \mid \mathsf{nil} & & &
\end{array}
$$

Let us consider the relation $\mathcal{S}$ depicted in Table 8; note that $(C_0, C_1) \in \mathcal{S}$, so that in order to prove that $C_0 \approx C_1$ it is sufficient to show that $\mathcal{S}$ is a bisimulation. Note that in the relation $\mathcal{S}$ the system terms W_{ok}, V_{ok} are always associated with a channel environment in which the channel c is exposed. In fact, if Λ were a channel environment such that $\Lambda \vdash c : \mathbf{idle}$, it would not be difficult

to prove that $\Lambda \triangleright W_{\mathsf{err}} \not\approx \Lambda \triangleright V_{\mathsf{err}}$; this is because the values broadcast by these two configurations are different.

Let us list the main the extensional actions from configurations using these system terms:

$$
\begin{array}{lcl}
\Delta \triangleright W & \stackrel{\tau}{\longmapsto} & (\Delta[c \mapsto (1, v_0)]) \triangleright W_0 \text{ if } \Delta \vdash c : \mathbf{idle} \\
\Delta \triangleright V & \stackrel{\tau}{\longmapsto} & (\Delta[c \mapsto (2, v_1)]) \triangleright V_1 \text{ if } \Delta \vdash c : \mathbf{idle} \\
\Delta \triangleright W & \stackrel{\tau}{\longmapsto} & (\Delta[c \mapsto (2, ok)]) \triangleright W_{ok} \\
\Delta \triangleright V & \stackrel{\tau}{\longmapsto} & (\Delta[c \mapsto (2, ok)]) \triangleright V_{ok} \\
\Delta \triangleright W & \stackrel{d?w}{\longmapsto} & (\mathrm{upd}_{d?w}(\Delta)) \triangleright W \\
\Delta \triangleright V & \stackrel{d?w}{\longmapsto} & (\mathrm{upd}_{d?w}(\Delta)) \triangleright V \\
 & & \\
(\Delta[c \mapsto (1, v_0)]) \triangleright W_0 & \stackrel{\tau}{\longmapsto} & (\Delta[c \mapsto (2, \mathsf{err})]) \triangleright W_{\mathsf{err}} \\
(\Delta[c \mapsto (2, v_1)]) \triangleright V_1 & \stackrel{\tau}{\longmapsto} & (\Delta[c \mapsto (2, \mathsf{err})]) \triangleright W_{\mathsf{err}} \\
\Delta \triangleright W_0 & \stackrel{c?w}{\longmapsto} & (\Delta[c \mapsto (1, \mathsf{err})]) \triangleright W_0 \text{ if } \Delta \vdash c : \mathbf{exp}, \delta_w = 1 \\
\Delta \triangleright V_1 & \stackrel{c?w}{\longmapsto} & (\Delta[c \mapsto (2, \mathsf{err})]) \triangleright V_1 \text{ if } \Delta \vdash c : \mathbf{exp}, \delta_w = 1 \\
\Delta \triangleright W_0 & \stackrel{c?w}{\longmapsto} & (\Delta[c \mapsto (\delta_w, \mathsf{err})]) \triangleright W_0 \text{ if } \Delta \vdash c : \mathbf{exp}, \delta_w > 1 \\
\Delta \triangleright V_1 & \stackrel{c?w}{\longmapsto} & (\Delta[c \mapsto (\delta_w, \mathsf{err})]) \triangleright V_1 \text{ if } \Delta \vdash c : \mathbf{exp}, \delta_w > 1 \\
 & & \\
\Lambda \triangleright W_{ok} & \stackrel{\tau}{\longmapsto} & (\mathrm{upd}_{c!v_0}(\Lambda)) \triangleright W_{\mathsf{err}} \\
\Lambda \triangleright V_{ok} & \stackrel{\tau}{\longmapsto} & (\mathrm{upd}_{c!v_1}(\Lambda)) \triangleright V_{\mathsf{err}} \\
 & & \\
\Delta \triangleright W_{\mathsf{err}} & \stackrel{\sigma}{\longmapsto} & (\mathrm{upd}_{\sigma}(\Delta)) \triangleright W' \\
\Delta \triangleright V_{\mathsf{err}} & \stackrel{\sigma}{\longmapsto} & (\mathrm{upd}_{\sigma}(\Delta)) \triangleright V' \\
 & & \\
\Delta \triangleright W' & \stackrel{\sigma}{\longmapsto} & (\mathrm{upd}_{\sigma}(\Delta)) \triangleright E \\
\Delta \triangleright V' & \stackrel{\sigma}{\longmapsto} & (\mathrm{upd}_{\sigma}(\Delta)) \triangleright E
\end{array}
$$

Here Δ, Λ are two arbitrary channel environments, but Λ is subject to the constraint that $\Lambda(c) = (k, w)$ for some value w and integer $k \geq 2$. Note that in this case we have that $(\mathrm{upd}_{c!v_0}(\Lambda)) = (\mathrm{upd}_{c!v_1}(\Lambda))$. With the aid of this tabulation one can now show that $\mathcal{S}$ is indeed a bisimulation and therefore that $C_0 \approx C_1$. □

Example 3.6 [Merging] The last example we provide considers the merging of two transmissions in a single transmission. Let Γ be a stable channel environment and v_0, v_1 be two values such that $\delta_{v_0} = 1, \delta_{v_1} = 2$. Also let ok be a value such that $\delta_{ok} = 3$. Consider the configurations

$$C_0 = \Gamma \triangleright W \qquad\qquad C_1 = \Gamma \triangleright V$$

where $W = c\,!\langle v_0 \rangle.c!\langle v_1 \rangle \mid c!\langle ok \rangle$ and $V = c\,!\langle v_1 \rangle.c!\langle v_0 \rangle \mid \sigma.c!\langle ok \rangle$.

Table 9 A relation $\mathcal{S}$ for comparing the configurations C_0, C_1 of Example 3.6

$$
\begin{array}{rcl}
\Delta_0 \triangleright W & \mathcal{S} & \Delta_0 \triangleright V \\
\Delta_1 \triangleright W_0 & \mathcal{S} & \Delta_2 \triangleright V_1 \\
\Delta_3 \triangleright W_{ok} & \mathcal{S} & \Delta_3 \triangleright V_{ok} \\
\Delta_3 \triangleright W_{\mathsf{err}} & \mathcal{S} & \Delta_3 \triangleright V_{\mathsf{err}} \\
\Delta_2 \triangleright W' & \mathcal{S} & \Delta_2 \triangleright V' \\
\Delta_2 \triangleright W_1 & \mathcal{S} & \Delta_2 \triangleright V' \\
\Delta_1 \triangleright E' & \mathcal{S} & \Delta_1 \triangleright V''
\end{array}
$$

$\Delta_n, n \geq 0$ arbitrary channel environment such that $\Delta \vdash_t c : m$ for some integer $m \geq n$.

Then $C_0 \approx C_1$. As in previous examples, this statement can be proved formally by exhibiting a bisimulation that contains the pair (C_0, C_1); to this end, define the following system terms:

$$
\begin{array}{rclcrcl}
W_0 & = & \sigma.c!\langle v_1\rangle \mid c!\langle ok\rangle & \qquad & V_1 & = & \sigma^2.c!\langle v_0\rangle \mid c!\langle ok\rangle \\
W_{ok} & = & c\,!\langle v_0\rangle.c!\langle v_1\rangle \mid \sigma^3 & & V_{ok} & = & c\,!\langle v_1\rangle.c!\langle v_0\rangle \mid \sigma^3 \\
W_{\mathsf{err}} & = & \sigma.c!\langle v_1\rangle \mid \sigma^3 & & W_{\mathsf{err}} & = & \sigma^2.c!\langle v_0\rangle \mid \sigma^3 \\
W' & = & c!\langle v_1\rangle \mid \sigma^2 & & & & \\
W_1 & = & \sigma^2 \mid \sigma^2 & & V' & = & \sigma.c!\langle v_0\rangle \mid \sigma^2 \\
E' & = & \sigma \mid \sigma & & V'' & = & c!\langle v_0\rangle \mid \sigma \\
E & = & \mathsf{nil} \mid \mathsf{nil} & & & &
\end{array}
$$

Consider now the relation $\mathcal{S}$ depicted in Table 9; note that $C_0\ \mathcal{S}\ C_1$. We leave the reader to check that $\mathcal{S}$ is also a weak bisimulation, from which $C_0 \approx C_1$ follows. □

4 Full abstraction

In this section, we show that the co-inductive proof method based on the bisimulation of the previous section is both sound respect to the contextual equivalence of Section 2.4; this is the subject of Section 4.1. Moreover it is complete for a large class of systems. This class is isolated in the following section, and the completeness result is then given in Section 4.3.

4.1 Soundness

In this section we prove that (weak) bisimulation equivalence is contained in reduction barbed congruence. The main difficulty is in proving the contextuality the bisimulation equivalence. But first some auxiliary results.

Lemma 4.1 [Update of Channel Environments] If $\Gamma \triangleright W \Longmapsto \Gamma' \triangleright W'$ then $\Gamma \leq \Gamma'$.

Proof Follows directly from Proposition 2.16 and the fact that τ-extensional actions coincide by definition with the instantaneous reduction $\rightarrowtriangle_i$. □

Corollary 4.2 For any channel c, $\Gamma \triangleright W \stackrel{\iota(c)}{\Longrightarrow}$ implies $\Gamma \triangleright W \stackrel{\iota(c)}{\longmapsto}$. □

Corollary 4.2 is very useful when proving that the exposure state of channels is preserved by bisimilar configurations.

Below we report a result on channel exposure for bisimilarity; a similar result for reduction barbed congruence will also be proved, in Proposition 4.14.

Lemma 4.3 [Channel exposure wrt $\approx$] Whenever $\Gamma_1 \triangleright W_1 \approx \Gamma_2 \triangleright W_2$ then $\Gamma_1 \vdash c$: **idle** if and only if $\Gamma_2 \vdash c$: **idle**.

Proof Suppose $\Gamma_1 \triangleright W_1 \approx \Gamma_2 \triangleright W_2$. If $\Gamma_1 \vdash c$: **idle** then by definition of Rule (Idle) of Table 7 it follows that $\Gamma_1 \triangleright W_1 \stackrel{\iota(c)}{\longmapsto}$. As $\Gamma_1 \triangleright W_1 \approx \Gamma_2 \triangleright W_2$, it follows that $\Gamma_2 \triangleright W_2 \stackrel{\iota(c)}{\Longrightarrow}$. From Corollary 4.2 we have that $\Gamma_2 \triangleright W_2 \stackrel{\iota(c)}{\longmapsto}$, and by the definition of Rule (Idle) that $\Gamma_2 \vdash c$: **idle**. □

In order to prove that weak bisimulation is sound with respect to reduction barbed congruence we need to show that $\approx$ is preserved by parallel composition.

Theorem 4.4 [$\approx$ is contextual] Suppose $\Gamma_1 \triangleright W_1 \approx \Gamma_2 \triangleright W_2$. Then for any system term W, $\Gamma_1 \triangleright (W_1 \mid W) \approx \Gamma_2 \triangleright (W_2 \mid W)$.

Proof Let the relation $\mathcal{S}$ over configurations be defined as follows:

$$\{(\Gamma_1 \triangleright W_1 \mid W\,,\ \Gamma_2 \triangleright W_2 \mid W) : \quad \Gamma_1 \triangleright W_1 \approx \Gamma_2 \triangleright W_2\}$$

It is sufficient to show that $\mathcal{S}$ is a bisimulation in the extensional semantics. To do so, by symmetry, we need to show that an arbitrary extensional action

$$\Gamma_1 \triangleright W_1 \mid W \stackrel{\alpha}{\longmapsto} \widehat{\Gamma_1} \triangleright \widehat{W_1} \tag{4}$$

can be matched by $\Gamma_1 \triangleright W_2 \mid W$ via a corresponding weak extensional action.

The action (4) can be inferred by any of the six rules in Table 7. We consider only one case, the most difficult one (Shh). So here α is τ and $\Gamma_1 \triangleright W_1 \mid W \stackrel{c!v}{\longrightarrow} \widehat{W_1}$, for some c and v. This transition in turn can always be inferred by an application of the rule (Sync) from Table 3. Without loss of generality we can assume

- $\Gamma_1 \triangleright W_1 \stackrel{c!v}{\longrightarrow} W_1'$
- $\Gamma_1 \triangleright W \stackrel{c?v}{\longrightarrow} W'$
- $\widehat{W_1} = W_1' \mid W'$

By an application of rule (Shh) it follows that $\Gamma_1 \triangleright W_1 \stackrel{\tau}{\longmapsto} \Gamma_1' \triangleright W_1'$, with $\Gamma_1' = \mathrm{upd}_{c!v}(\Gamma_1)$. Since $\Gamma_1 \triangleright W_1 \approx \Gamma_2 \triangleright W_2$, there is $\Gamma_2' \triangleright W_2'$ such that $\Gamma_2 \triangleright W_2 \Longrightarrow \Gamma_2' \triangleright W_2'$ and $\Gamma_1' \triangleright W_1' \approx \Gamma_2' \triangleright W_2'$. Now, there are two possibilities, depending on whether or not c is exposed in Γ_1.

1. Let $\Gamma_1 \vdash c : \mathbf{exp}$. By Lemma 2.1(2), in the transition $\Gamma_1 \triangleright W \xrightarrow{c?v} W'$ it must be that $W' = W$. Since $\Gamma_1 \triangleright W_1 \approx \Gamma_2 \triangleright W_2$ and $\Gamma_1' \triangleright W_1' \approx \Gamma_2' \triangleright W_2'$, by two applications of Lemma 4.3 it follows that:

   - for any channel d, $\Gamma_1 \vdash d : \mathbf{idle}$ iff $\Gamma_2 \vdash d : \mathbf{idle}$
   - for any channel d, $\Gamma_1' \vdash d : \mathbf{idle}$ iff $\Gamma_2' \vdash d : \mathbf{idle}$.

   We recall that $\Gamma_1' = \mathrm{upd}_{c!v}(\Gamma_1)$, and hence Γ_1' and Γ_1 may only differ for the entry at channel c. As a consequence, also Γ_2 and Γ_2' may only differ for the same entry.

   Now, let us analyse the transitions which constitute the weak derivation

   $$\Gamma_2 \triangleright W_2 \Longrightarrow \Gamma_2' \triangleright W_2'$$

   In particular, let

   $$\Gamma_2 \triangleright W_2 \Longrightarrow \Gamma_2^n \triangleright W_2^n \overset{\tau}{\longmapsto} \Gamma_2^{n+1} \triangleright W_2^{n+1} \Longrightarrow \Gamma_2' \triangleright W_2' \ .$$

   There are two possibilities.

   (a) $\Gamma_2^n \triangleright W_2^n \overset{\tau}{\longmapsto} \Gamma_2^{n+1} \triangleright W_2^{n+1}$ is derived by an application of rule (TauExt) because $\Gamma_2^n \triangleright W_2^n \xrightarrow{\tau} W_2^{n+1}$. This case is easy.

   (b) $\Gamma_2^n \triangleright W_2^n \overset{\tau}{\longmapsto} \Gamma_2^{n+1} \triangleright W_2^{n+1}$ is derived by an application of rule (Shh) because $\Gamma_2^n \triangleright W_2^n \xrightarrow{d!w} W_2^{n+1}$, for some d and w. Since Γ_2 and Γ_2' may only differ for the entry at channel c, also Γ_2^n and Γ_2^{n+1} may only differ for the same entry. This is because the derivation $\Gamma_2 \triangleright W_2 \Longrightarrow \Gamma_2' \triangleright W_2'$ is untimed, and once a channel becomes exposed it remains so for the whole derivation. By Lemma 4.3, $\Gamma_1 \vdash c : \mathbf{exp}$ implies $\Gamma_2 \vdash c : \mathbf{exp}$. By definition of rule (Shh), $\Gamma_2^{n+1} \vdash d : \mathbf{exp}$. Since only the entry at c may change during the derivation it follows that $\Gamma_2^n \vdash d : \mathbf{exp}$ (also for $d = c$). By Lemma 2.1(2), this implies $\Gamma_2^n \triangleright W \xrightarrow{d?w} W$. By an application of rule (Sync) and one application of rule (Shh) we can derive

   $$\Gamma_2^n \triangleright W_2^n \mid W \overset{\tau}{\longmapsto} \Gamma_2^{n+1} \triangleright W_2^{n+1} \mid W$$

   As a consequence,

   $$\Gamma_2 \triangleright W_2 \mid W \Longrightarrow \Gamma_2' \triangleright W_2' \mid W$$

   with $(\Gamma_1' \triangleright W_1' \mid W \, , \, \Gamma_2' \triangleright W_2' \mid W) \in \mathcal{S}$.

2. Let $\Gamma_1 \vdash c : \mathbf{idle}$. There are two sub-cases.

   (a) Let $\neg \mathsf{rcv}(W, c)$. This case is similar to case 1. In fact, by Lemma 2.1(1) the system W is not affected by the transmission at c. More precisely, the transition $\Gamma_1 \triangleright W_1 \mid W \xrightarrow{c!v} \widehat{W_1}$ can only be derived by an application of rule (Sync) because

   - $\Gamma_1 \triangleright W_1 \xrightarrow{c!v} W_1'$

- $\Gamma_1 \triangleright W \xrightarrow{c?v} W$
- $\widehat{W_1} = W_1' \mid W$.

(b) Let $\mathsf{rcv}(W, c)$. By Lemma 2.1(3) the transition $\Gamma_1 \triangleright W \xrightarrow{c?v} W'$ must have $W' \neq W$. Since $\Gamma_1 \triangleright W_1 \approx \Gamma_2 \triangleright W_2$, by Lemma 4.3 it follows that $\Gamma_2 \vdash c : \mathbf{idle}$. As $\Gamma_1' = \mathrm{upd}_{c!v}(\Gamma_1)$, it follows that $\Gamma_1' \vdash c : \mathbf{exp}$. Since $\Gamma_1' \triangleright W_1' \approx \Gamma_2' \triangleright W_2'$, by Lemma 4.3 it follows that $\Gamma_2' \vdash c : \mathbf{exp}$. As a consequence, the derivation $\Gamma_2 \triangleright W_2 \Longmapsto \Gamma_2' \triangleright W_2'$ must contain a transition which changes the exposure state of channel c. More precisely, the derivation must have the form

$$\Gamma_2 \triangleright W_2 \Longmapsto \Gamma_2^n \triangleright W_2^n \overset{\tau}{\longmapsto} \Gamma_2^{n+1} \triangleright W_2^{n+1} \Longmapsto \Gamma_2' \triangleright W_2'$$

where the transition $\Gamma_2^n \triangleright W_2^n \overset{\tau}{\longmapsto} \Gamma_2^{n+1} \triangleright W_2^{n+1}$ is due to an application of rule (Shh) because $\Gamma_2^n \triangleright W_2^n \xrightarrow{c!w} W_2^{n+1}$, for some w, with $\Gamma_2^n \vdash c : \mathbf{idle}$ and $\Gamma_2^{n+1} \vdash c : \mathbf{exp}$. By Lemma 2.1(3) it follows that $\Gamma_2^n \triangleright W \xrightarrow{c?w} W'$, for any w. By an application of rule (Sync) and one of rule (Shh), it follows that

$$\Gamma_2^n \triangleright W_2^n \mid W \overset{\tau}{\longmapsto} \Gamma_2^{n+1} \triangleright W_2^{n+1} \mid W'$$

For any other transition in the derivation $\Gamma_2 \triangleright W_2 \Longmapsto \Gamma_2' \triangleright W_2'$ we can apply the same reasoning as in case 1. In particular, for those transitions which are derived by an application of rule (Shh) because of a transition labelled $d!w'$, the channel d must be exposed before and (obviously) after the transition. So, by Lemma 2.1(2) the systems W and W' can perform the corresponding action $d?w'$ remaining unchanged. More precisely, we have

$$\Gamma_2 \triangleright W_2 \mid W \Longmapsto \Gamma_2^n \triangleright W_2^n \mid W \overset{\tau}{\longmapsto} \Gamma_2^{n+1} \triangleright W_2^{n+1} \mid W' \overset{\tau}{\Longmapsto} \Gamma_2' \triangleright W_2' \mid W'$$

with $(\Gamma_1' \triangleright W_1' \mid W' \,,\, \Gamma_2' \triangleright W_2' \mid W') \in \mathcal{S}$. □

Theorem 4.5 [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 Lemma 4.3. Contextuality follows from Theorem 4.4. □

4.2 Well-formed systems

Intuitively the extensional actions represent behaviour which can be detected by contexts. But in CCCP the ability of contexts to detect this behaviour depends to some extent on the system being tested behaving appropriately. Consider a simple example. We know that all configurations are *input-enabled*, that is at any time can receive any value v along any channel channel c. But suppose we wanted to check that a particular configuration $\Gamma \triangleright W$ has actually performed such a

reception; this would be necessary, for example, if we wished to investigate the behaviour of its residual after the action $c?v$.

Let $T_{c,v}$ denote the system $c\,!\langle v\rangle.\mathit{eureka}!\langle ok\rangle + \mathit{fail}!\langle no\rangle$, where *eureka* and *fail* are some fresh channels. Then intuitively we might expect that, at least informally, $\Gamma \triangleright W$ will have performed the action $c?v$ when the combined system $\Gamma \triangleright (W \mid T_{c,v})$ no longer has a barb on *fail* but does have a barb on *eureka*.

However the existence of the barb on *eureka* depends on the ability of time to pass in the combined system: the transmission along c can commence, but the delivery of v takes time. If time can not proceed then the potential barb on *eureka* will never materialise.

Example 4.6 Let W denote the system $d[x].P$, for some channel d different from c, and suppose $\Gamma \vdash d : \mathbf{idle}$. Then $\Gamma \triangleright (W \mid T_{c,v})$ can never produce a barb on *eureka*: $\Gamma \triangleright (W \mid T_{c,v}) \rightarrow\!\!\triangleright^* C'$ implies $C' \downarrow_{\mathit{eureka}}$ is not true.

Note in particular that the sub-configuration $\Gamma \triangleright W$ can not perform a σ action; it does not allow time to pass. The only possible rule that might be applied is (EndRcv) from Table 4; but this requires $\Gamma \vdash_t d : n$ for some $n > 1$, whereas $\Gamma \vdash d : \mathbf{idle}$. □

Definition 4.7 [Well-formedness] The set of well-formed configurations WNets is the least set such that

$$\begin{array}{rcll} \Gamma \triangleright P \in \mathrm{WNets} & & & \text{for all processes } P \\ \Gamma \vdash c : \mathbf{exp} & \text{implies} & \Gamma \triangleright c[x].P \in \mathrm{WNets} & \\ \Gamma \triangleright W_1, \Gamma \triangleright W_2 \in \mathrm{WNets} & \text{implies} & \Gamma \triangleright W_1 \mid W_2 \in \mathrm{WNets} & \\ \Gamma[c \mapsto (n, v)] \triangleright W \in \mathrm{WNets} & \text{implies} & \Gamma \triangleright \nu c : (n, v).W \in \mathrm{WNets} & \end{array}$$ □

Intuitively a configuration $\Gamma \triangleright W$ is well-formed if it does not contain any receiving station along an idle channel. Note that the configuration used in Example 4.6 is not well-formed.

Lemma 4.8 Suppose C is well-formed and $C \rightarrow\!\!\triangleright C'$. Then C' is also well-formed.

Proof Suppose $\Gamma \triangleright W \xrightarrow{\lambda} W'$ and $\Gamma \triangleright W$. Then by rule induction on $\Gamma \triangleright W \xrightarrow{\lambda} W'$ one can show that $\mathrm{upd}_\lambda(\Gamma) \triangleright W'$ is also well-formed.

The result now follows by consideration of the three possible cases for deriving $C \rightarrow\!\!\triangleright C'$ in Definition 2.5. □

The main property of well-formed systems is that they allow the passage of time, so long as all internal activity has ceased:

Proposition 4.9 Let $\Gamma \triangleright W$ be a well-formed configuration such that $\Gamma \triangleright W \not\rightarrow\!\!\triangleright_i$; then $\Gamma \triangleright W \rightarrow\!\!\triangleright_\sigma$.

Proof By structural induction on the definition of the set WNets of well-formed networks; this result relies on the fact that we only allow guarded recursion in the language. □

It would seem that restricting attention to well-formed configurations does not preclude the phenomenon exhibited in Example 4.6 from occurring. Since our language for station code includes recursion the reader could argue that it is possible to write systems which can perform an infinite sequence of instantaneous reductions; we first identify this systems formally, then we show that these cannot be obtained in our calculus.

Example 4.10 Let W denote the code $\mathsf{fix}\, X.\tau.X$. Then we have an infinite sequence of internal actions

$$\Gamma \triangleright W \rightarrow_i C_1 \rightarrow_i \ldots C_k \rightarrow_i$$

Indeed one can show that if $\Gamma \triangleright W \rightarrow^* C'$ then $C' \rightarrow_i$. Maximal progress then ensures that $C' \not\rightarrow_\sigma$. □

Definition 4.11 [Well-timed configurations] A configuration C is *well-timed*, [27], if there exists an upper bound $k \in \mathbb{N}$ such that whenever $C \rightarrow_i^h C'$ for some $h \geq 0$, then $h \leq k$. □

While well-formedness is a simple syntactic constraint, *well-timed* means that the designer of the network has to ensure that the code placed at the station nodes can never lead to divergent behaviour. One simple method for ensuring this is to only use recursive definitions $\mathsf{fix}\, X.P$ where X is weakly guarded in P; that is, every occurrence of X is within an input, output or time delay prefix, or it is included within a branch of a matching construct. These are exactly the conditions that we placed for recursion variables when defining our calculus. Thus, we would expect every configuration in our calculus to be well-timed. In order to prove formally this statement, we need the following technical result:

Proposition 4.12 An environment ρ is a partial mapping from process variables to closed terms. Given a term W and an environment ρ, we denote with $W\rho$ the system term obtained by substituting each free occurrence of any process variable X with $\rho(X)$, if defined.

For any channel environment Γ, (possibly open) term W and environment ρ such that $\Gamma \triangleright W\rho$ is well-defined then it is also well-timed.

Proof By structural induction on the term W, using the hypothesis that we only allow guarded recursion. □

Corollary 4.13 Any well-formed configuration $\Gamma \triangleright W$ is also well-timed.

Proof Note that if $\Gamma \triangleright W$ is well-formed then W is closed by definition. Thus, for any environment ρ we have that $W\rho = W$. Now the result follows directly from Proposition 4.12. □

To end this section we prove a very useful result for well-defined configurations; the proof emphasises the roles of well-formedness and well-timedness in the configurations being tested.

Proposition 4.14 Suppose $\Gamma_1 \triangleright W_1 \simeq \Gamma_2 \triangleright W_2$, where both are well-formed. Then $\Gamma_1 \vdash c : \mathbf{idle}$ implies $\Gamma_2 \vdash c : \mathbf{idle}$.

Proof Let $\Gamma_1 \triangleright W_1 \simeq \Gamma_2 \triangleright W_2$ and suppose $\Gamma_1 \vdash c : \mathbf{idle}$ for some channel c. Consider the testing code:

$$T = [\exp(c)]\mathsf{nil}, \mathit{eureka}!\langle \mathit{ok} \rangle$$

From the definition of $\simeq$ we know that $\Gamma_1 \triangleright W_1 \mid T \simeq \Gamma_2 \triangleright W_2 \mid T$. Since $\Gamma_1 \triangleright W_1$ is well-timed, by definition there is a configuration C such that $\Gamma_1 \triangleright W_1 \ \rightarrow\!\triangleright_i^* \ C$ and $C \not\rightarrow\!\triangleright_i$. Because $\Gamma_1 \triangleright W_1$ is well-formed so is this C and so by Proposition 4.9 there is a configuration C' such that $C \rightarrow\!\triangleright_\sigma C'$.

It follows, by the existence of this C', that $\Gamma_1 \triangleright W_1 \mid T \Downarrow_{\mathit{eureka}}$, which in turn means that $\Gamma_2 \triangleright W_2 \mid T \Downarrow_{\mathit{eureka}}$.

By maximal progress, this is only possible if

$$\Gamma_2 \triangleright W_2 \mid T \ \rightarrow\!\triangleright_i^* \ \Gamma_2' \triangleright W_2' \mid \sigma.\mathit{eureka}!\langle \mathit{ok} \rangle \ \rightarrow\!\triangleright_i^* \rightarrow\!\triangleright_\sigma \rightarrow\!\triangleright_i^* \ \Gamma_2'' \triangleright W_2'' \mid \sigma^{\delta_{ok}}$$

where Γ_2' is a channel environment such that $\Gamma_2' \vdash c : \mathbf{idle}$. From Lemma 2.16 we get the required $\Gamma_2 \vdash c : \mathbf{idle}$. □

4.3 Completeness

In this section we prove that our notion of bisimilarity is not only sound with respect to reduction barbed congruence but is also complete for well-formed configurations. The idea here is to prove that $\simeq$ is a bisimulation in the extensional LTS. We address the various requirements individually. First a technical result about fresh barbs.

Lemma 4.15 Suppose $\Gamma_1 \triangleright W_1 \mid T \simeq \Gamma_2 \triangleright W_2 \mid T$ where T does not contain any free occurrences of channel names which occur free in either W_1 or W_2. Then $\Gamma_1 \triangleright W_1 \simeq \Gamma_2 \triangleright W_2$.

Proof [Outline] This is a variation on analogous results already in the literature for a number of different process calculi; see for example Lemma 2.38 of [17].

Let the relation $\mathcal{R}$ over configurations be defined by letting

$$\Gamma_1 \triangleright W_1 \ \mathcal{R} \ \Gamma_2 \triangleright W_2$$

whenever $\Gamma_1 \triangleright W_1 \mid T \simeq \Gamma_2 \triangleright W_2 \mid T$ for some term T which only uses free channels with respect to W_1 and W_2. One can check that $\mathcal{R}$ is reduction-closed, contextual and barb-preserving, from which the result follows. □

Next we show that reduction barbed congruence is preserved by all the actions in the extensional semantics. This can be accomplished by providing, for each of these extensional actions α, a distinguishing context T_α such that which is able to test whether a configuration can perform the (weak) extensional action α. For some particular α the distinguishing contexts will only work for well-formed configurations. First we show the case regarding extensional τ-actions.

Proposition 4.16 [Preserving extensional τs] Suppose $\Gamma_1 \triangleright W_1 \simeq \Gamma_2 \triangleright W_2$ and $\Gamma_1 \triangleright W_1 \overset{\tau}{\longmapsto} \Gamma_1' \triangleright W_1'$. Then $\Gamma_2 \triangleright W_2 \Longmapsto \Gamma_2' \triangleright W_2'$ such that $\Gamma_1' \triangleright W_1' \simeq \Gamma_2' \triangleright W_2'$.

Proof There are two possible reasons why $\Gamma_1 \triangleright W_1 \overset{\tau}{\longmapsto} \Gamma_1' \triangleright W_1'$:

(i) $\Gamma_1 \triangleright W_1 \xrightarrow{\tau} W_1'$ and $\Gamma_1' = \mathrm{upd}_\tau(\Gamma_1) = \Gamma_1$, by an application of rule (TauExt)

(ii) $\Gamma_1 \triangleright W_1 \xrightarrow{c!v} W_1'$ and $\Gamma_1' = \mathrm{upd}_{c!v}(\Gamma_1)$, by an application of rule (Shh).

We consider the first case; the proof for the second case is virtually identical.

Let *eureka* be a fresh channel; that is it must satisfy $\mathit{eureka} \notin \mathrm{fn}(W)$ and $\Gamma_1 \vdash \mathit{eureka} : \mathbf{idle}$. Let *ok* be a message which requires one time unit to be transmitted, i.e. $\delta_{ok} = 1$. By an application of rules (TauPar) and (TauExt) we derive

$$\Gamma_1 \triangleright W_1 \mid \mathit{eureka}!\langle ok\rangle \overset{\tau}{\longmapsto} \Gamma_1' \triangleright W_1' \mid \mathit{eureka}!\langle ok\rangle$$

with $\Gamma_1' \triangleright W_1' \mid \mathit{eureka}!\langle ok\rangle \Downarrow_{\mathit{eureka}}$ and $\Gamma_1' \vdash \mathit{eureka} : \mathbf{idle}$. By Definition 2.5 this transition corresponds in the reduction semantics to

$$\Gamma_1 \triangleright W_1 \mid \mathit{eureka}!\langle ok\rangle \rightarrowtriangle \Gamma_1' \triangleright W_1' \mid \mathit{eureka}!\langle ok\rangle$$

As $\Gamma_1 \triangleright W_1 \simeq \Gamma_2 \triangleright W_2$ and $\simeq$ is contextual, this step must be matched by a sequence of reductions

$$\Gamma_2 \triangleright W_2 \mid \mathit{eureka}!\langle ok\rangle \rightarrowtriangle^* C \tag{5}$$

such that $\Gamma_1' \triangleright W_1' \mid \mathit{eureka}!\langle ok\rangle \simeq C$. Depending on whether the transmission at *eureka* is part of the sequence of reductions or not, the configuration C must be one of the following:

$$\begin{array}{lcll} C_1 & = & \Gamma_2' \triangleright W_2' \mid \mathit{eureka}!\langle ok\rangle & \text{with} \quad \Gamma_2' \vdash \mathit{eureka} : \mathbf{idle} \\ C_2 & = & \Gamma_2' \triangleright W_2' \mid \sigma.\mathsf{nil} & \text{with} \quad \Gamma_2' \vdash \mathit{eureka} : \mathbf{exp} \\ C_3 & = & \Gamma_2' \triangleright W_2' \mid \mathsf{nil} & \text{with} \quad \Gamma_2' \vdash \mathit{eureka} : \mathbf{idle} \end{array}$$

As *eureka* is a fresh channel and $C_3 \not\Downarrow_{\mathit{eureka}}$, it follows that C cannot be C_3. Since $\Gamma_1' \triangleright W_1' \mid \mathit{eureka}!\langle ok\rangle \simeq C$ and $\Gamma_1' \vdash \mathit{eureka} : \mathbf{idle}$, by Proposition 4.14 (which can be applied since we are assuming that C is both well-formed and well-timed) it follows that C cannot be C_2. So, the only possibility is $C = C_1$. By Lemma 4.15 it follows that $\Gamma_1' \triangleright W_1' \simeq \Gamma_2' \triangleright W_2'$. It remains to show that $\Gamma_2 \triangleright W_2 \Longmapsto \Gamma_2' \triangleright W_2'$.

To this end we can extract out from the reduction sequence (5) above a reduction sequence

$$\Gamma_2 \triangleright W_2 \rightarrowtriangle^* \Gamma_2' \triangleright W_2'$$

We show that each step in this sequence, say $\Gamma \triangleright W \rightarrow\!\triangleright\ \Gamma' \triangleright W'$, is actually also a τ step in the extensional LTS, $\Gamma \triangleright W \stackrel{\tau}{\longmapsto} \Gamma' \triangleright W'$, from which the result follows.

Recall from Definition 2.5 that there are three possible ways to infer the reduction step $\Gamma \triangleright W \rightarrow\!\triangleright\ \Gamma' \triangleright W'$. If it is either (Internal), i.e. $\Gamma \triangleright W \stackrel{\tau}{\longrightarrow} W'$, or a (Transmission), i.e. $\Gamma \triangleright W \stackrel{c!v}{\longrightarrow} W'$, then by definition $\Gamma \triangleright W \stackrel{\tau}{\longmapsto} \Gamma' \triangleright W'$ follows. Condition (ii), (Time), is not possible because in the original sequence (5) above the testing component $eureka!\langle ok\rangle$ can not make a σ move. □

Next we show that reduction barbed congruence is preserved by σ-actions.

Proposition 4.17 [Preserving extensional σs] Suppose $\Gamma_1 \triangleright W_1 \simeq \Gamma_2 \triangleright W_2$. Then $\Gamma_1 \triangleright W_1 \stackrel{\sigma}{\longmapsto} \Gamma_1' \triangleright W_1'$ implies $\Gamma_2 \triangleright W_2 \stackrel{\sigma}{\Longmapsto} \Gamma_2' \triangleright W_2'$ such that $\Gamma_1' \triangleright W_1' \simeq \Gamma_2' \triangleright W_2'$.

Proof We will use a testing context:

$$T = \sigma.(\tau.eureka!\langle ok\rangle + fail!\langle no\rangle)$$

where *eureka* and *fail* are fresh channels. Let $\Gamma_1 \triangleright W_1 \mid T \rightarrow\!\triangleright^*\ \Gamma_1' \triangleright W_1' \mid eureka!\langle ok\rangle$ $(= C_1)$. Since $\Gamma_1 \triangleright W_1 \simeq \Gamma_2 \triangleright W_2$ we must have a series of reduction steps

$$\Gamma_2 \triangleright W_2 \mid T \rightarrow\!\triangleright^*\ C_2 \tag{6}$$

such that $C_1 \simeq C_2$. Because $C_1 \Downarrow_{eureka}$ and $C_1 \not\Downarrow_{fail}$ (recall that *fail* is fresh) the same must be true of C_2. As $\Gamma_1' \vdash eureka : \mathbf{idle}$, it follows that C_2 must take the form $\Gamma_2' \triangleright W_2' \mid eureka!\langle ok\rangle$. By Lemma 4.15 we have that $\Gamma_1' \triangleright W_1' \simeq \Gamma_2' \triangleright W_2'$. It remains to establish that $\Gamma_2 \triangleright W_2 \stackrel{\sigma}{\Longmapsto} \Gamma_2' \triangleright W_2'$.

We proceed as in the previous proposition, by extracting out of (6) the contributions from $\Gamma_2 \triangleright W_2$; we know that because of the presence of the time delay in T, and by maximal progress (Proposition 2.2), exactly one of the reduction steps involves the passage of time. So (6) actually takes the form

$$\Gamma_2 \triangleright W_2 \mid T_2 \rightarrow\!\triangleright_i^*\ \Gamma' \triangleright W' \mid \ldots \rightarrow\!\triangleright_\sigma\ \Gamma'' \triangleright W'' \mid \ldots \rightarrow\!\triangleright_i^*\ \Gamma_2' \triangleright W_2' \mid eureka!\langle ok\rangle$$

Each individual reduction step can now be projected on to the first component, giving the required

$$\Gamma_2 \triangleright W_2 \Longmapsto \Gamma \triangleright W \stackrel{\sigma}{\longmapsto} \Gamma' \triangleright W' \Longmapsto \Gamma_2' \triangleright W_2'$$

□

The cases for the actions $\alpha \in \{c?v, \iota(c), \gamma(c,v)\}$ are more complicated to analyse; as we will see, the distinguishing contexts for such actions will require several instants of time to detect whether the action has been performed by the tested configuration or not. To solve this problem, we equip the distinguishing context T_α with a fresh channel *halt*, different from *eureka* and *fail*, whose role is that of preventing time to pass when testing if a configuration has performed the (weak) extensional α-action.

As an example we consider in detail a testing context $T_{\gamma(c,v)}$ for the extensional action $\gamma(c,v)$,

$$vd{:}(0,\cdot).(c[x].([x{=}v]d!\langle ok\rangle, \mathsf{nil}) + fail!\langle no\rangle \mid \sigma^2.[\exp(d)]eureka!\langle ok\rangle, \mathsf{nil} \mid \sigma.halt!\langle ok\rangle)$$

where *eureka*, *halt*, *fail* are fresh channels and *ok* is a message such that $\delta_{ok} = 1$. This is designed to detect whether a configuration $\Gamma \triangleright W$ has performed a weak $\gamma(c, v)$-action. Let us discuss informally how the testing context $T_{\gamma(c,v)}$ operates. Each of the three fresh channels *eureka*, *halt*, *fail* plays a different role: *fail* ensures that the reception along channel c has finished; *eureka* guarantees that the received values is actually v; *halt* serves to stop the computation after one time unit.

We provide a possible evolution of the testing contexts $T_\gamma(c, v)$ when running in a channel environment Γ such that $\Gamma(c) = (1, v)$, and then we discuss how it works.

$$
\begin{array}{lll}
 & \Gamma_1 \triangleright T_{\gamma(c,v)} & \\
\rightarrow\!\!\triangleright_\sigma & \Gamma_1' \triangleright T_1 & = \Gamma_1' \triangleright \nu d{:}(0,\cdot).(([v{=}v]d!\langle ok\rangle, \mathsf{nil}) + \mathit{fail}!\langle no\rangle \mid \\
 & & \quad \mid \sigma.[\exp(d)]\mathit{eureka}!\langle ok\rangle, \mathsf{nil} \mid \mathit{halt}!\langle ok\rangle) \\
\rightarrow\!\!\triangleright_i & \Gamma_1' \triangleright T_2 & = \Gamma_1' \triangleright \nu d{:}(0,\cdot).(\sigma.d!\langle ok\rangle \mid \sigma.[\exp(d)]\mathit{eureka}!\langle ok\rangle, \mathsf{nil} \mid \mathit{halt}!\langle ok\rangle) \\
\rightarrow\!\!\triangleright_i & \Gamma_1'' \triangleright T_3 & = \Gamma_1'' \triangleright \nu d{:}(0,\cdot).(\sigma.d!\langle ok\rangle \mid \sigma.[\exp(d)]\mathit{eureka}!\langle ok\rangle, \mathsf{nil} \mid \sigma) \\
\rightarrow\!\!\triangleright_\sigma & \Gamma_1''' \triangleright T_4 & = \Gamma_1''' \triangleright \nu d{:}(0,\cdot).(d!\langle ok\rangle \mid [\exp(d)]\mathit{eureka}!\langle ok\rangle, \mathsf{nil} \mid \mathsf{nil}) \\
\rightarrow\!\!\triangleright_i & \Gamma_1'''' \triangleright T_5 & = \Gamma_1'''' \triangleright \nu d{:}(1, ok).(\sigma \mid [\exp(d)]\mathit{eureka}!\langle ok\rangle, \mathsf{nil} \mid \mathsf{nil}) \\
\rightarrow\!\!\triangleright_i & \Gamma_1''''' \triangleright T_6 & = \Gamma_1''''' \triangleright \nu d{:}(1, ok).(\sigma \mid \sigma.\mathit{eureka}!\langle ok\rangle \mid \mathsf{nil}) \\
\rightarrow\!\!\triangleright_\sigma & \Gamma_1{}'''''' \triangleright T_7 & = \Gamma_1'''''' \triangleright \nu d{:}(0,\cdot).(\mathsf{nil} \mid \mathit{eureka}!\langle ok\rangle \mid \mathsf{nil})
\end{array}
$$

Initially a configuration of the form $\Gamma \triangleright W \mid T_{\gamma(c,v)}$ has a weak barb at channels *halt* and *fail*. Further, the testing component has an active receiver over channel c; note that the configuration $\Gamma \triangleright W \mid T_{\gamma(c,v)}$ is well-formed only if $\Gamma \vdash c : \mathbf{exp}$. When channel c is released, the testing component checks if the received value is v, in which case the barb along channel *fail* disappears and a broadcast along a restricted channel d is fired one time instant after channel c has been released, and only in the case that the received value along channel c matches value v. Note that in the same instant of time the barb at *halt* disappears.

A second parallel component waits two time instants before checking if channel d is exposed; in this case it broadcasts over channel *eureka*. Note that the exposure check is passed only if a broadcast along channel d has been fired in the third time instant of a computation of $\Gamma \triangleright W \mid T_{\gamma(c,v)}$; this is possible only if channel c has been released in the second instant of time of such a computation, and the received value is v; that is, $\Gamma(c) = (1, v)$.

Proposition 4.18 [Preserving $\gamma(c, v)$-actions] Let $\Gamma_1 \triangleright W_1, \Gamma_2 \triangleright W_2$ be two configurations such that $\Gamma_1 \triangleright W_1 \simeq \Gamma_2 \triangleright W_2$ and $\Gamma_1 \triangleright W_1 \overset{\gamma(c,v)}{\longmapsto} \Gamma_1' \triangleright W_1'$. Then $\Gamma_2 \triangleright W_2 \overset{\gamma(c,v)}{\Longmapsto} \Gamma_2' \triangleright W_2'$ such that $\Gamma_1' \triangleright W_1' \simeq \Gamma_2' \triangleright W_2'$.

Proof First note that the extensional transition $\Gamma_1 \triangleright W_1 \overset{\gamma(c,v)}{\longmapsto} \Gamma_1' \triangleright W_1'$ can be derived in the extensional semantics only by using rule (Deliver); this implies that $\Gamma_1 \triangleright W_1 \xrightarrow{\sigma} W_1'$ and $\Gamma_1(c) = (1, v)$.

Now consider the configurations $C_1 = \Gamma_1 \triangleright W_1 \mid T_{\gamma(c,v)}$, $C_2 = \Gamma_2 \triangleright W_2 \mid T_{\gamma(c,v)}$ where the testing context $T_{\gamma(c,v)}$ is defined above. As $\Gamma_1 \triangleright W_1 \simeq \Gamma_2 \triangleright W_2$, by definition of barbed congruence it follows that $C_1 \simeq C_2$. It is easy to check that C_1 has the following reductions:

$$
\begin{array}{lll}
C_1 & \rightarrow\!\!\triangleright_\sigma & \Gamma_1' \triangleright W_1' \mid T_1 \\
 & \rightarrow\!\!\triangleright_i & \Gamma_1' \triangleright W_1' \mid T_2
\end{array}
$$

Moreover, $\Gamma_1' \triangleright W_1' \mid T_2 \rightarrow^* \Gamma_1'''' \triangleright W_1'''' \mid T_7 \Downarrow_{eureka}$, as can be seen from the following reduction sequence:

$$\begin{array}{rcl} \Gamma_1' \triangleright W_1' \mid T_2 & \rightarrow_i & \Gamma_1'' \triangleright W_1' \mid T_3 \\ & \rightarrow_\sigma & \Gamma_1''' \triangleright W_1'' \mid T_4 \\ & \rightarrow_i & \Gamma_1''' \triangleright W_1'' \mid T_5 \\ & \rightarrow_i & \Gamma_1''' \triangleright W_1'' \mid T_6 \\ & \rightarrow_\sigma & \Gamma_1'''' \triangleright W_1'''' \mid T_7 \end{array}$$

Thus, $\Gamma_1' \triangleright W_1' \mid T_2 \Downarrow_{eureka}$, $\Gamma_1' \triangleright W_1' \mid T_2 \Downarrow_{halt}$ and $\Gamma_1' \triangleright W_1' \mid T_2 \not\Downarrow_{fail}$; this last statement is true because *fail* is fresh in $\Gamma_1 \triangleright W_1$.

Since $C_1 \simeq C_2$, it follows that $C_2 \rightarrow^* C_2'$ for some C_2' such that $\Gamma_1' \triangleright W_1' \mid T_2 \simeq C_2'$. By definition of reduction barbed congruence it holds that $C_2' \Downarrow_{eureka}$, $C_2' \Downarrow_{halt}$ and $C_2' \not\Downarrow_{fail}$. Note that $\Gamma_1' \vdash halt : \mathbf{idle}$, hence by Proposition 4.14 it follows that $C_2' = \Gamma_2' \triangleright W$ for some Γ_2', W such that $\Gamma_2' \vdash halt : \mathbf{idle}$. Note also that $\Gamma_2' \triangleright T_2$ has a weak barb on both barbs *eureka* and *halt*, while it lacks a barb on *fail*. The reader can convince herself that this is possible only in the case that $C_2' = \Gamma_2' \triangleright W_2' \mid T_2$ for some system term W_2'.

The weak reduction $C_2 \ (\rightarrow_i)^* \ C_2'$ can be unfolded as follows:

$$\begin{array}{rcl} C_2 \ (\rightarrow_i)^* & \Gamma_2'' \triangleright W_2'' \mid T_{\gamma(c,v)} \\ \rightarrow_\sigma & \Gamma_2''' \triangleright W_2''' \mid T_1 \\ (\rightarrow_i)^* & \Gamma_2^{iv} \triangleright W_2^{iv} \mid T_1 \\ \rightarrow_i & \Gamma_2^{iv} \triangleright W_2^{iv} \mid T_2 \\ (\rightarrow_i)^* & \Gamma_2' \triangleright W_2' \mid T_2 \end{array} \tag{7}$$

Note that in the reduction sequence (7) it is crucial that in the testing component T_1 both the broadcast along channel *halt* and the exposure check of the restricted channel d are performed in the same time instant. Since the testing components in the reduction sequence (7) above never contain a sender along a non-fresh channel, nor they are waiting to receive a value, it is straightforward to note that the sequence of reductions above induces the sequence

$$\Gamma_2 \triangleright W_2 \rightarrow_i^* \rightarrow_\sigma \rightarrow_i^* \Gamma_2' \triangleright W_2' \ .$$

By definition of our reduction semantics we have that $\Gamma_2 \triangleright W_2 \stackrel{\sigma}{\Longrightarrow} \Gamma_2' \triangleright W_2'$.

It remains to show that $\Gamma_2(c) = (1, v)$. Since we assumed that $\Gamma_1(c) = (1, v)$ and $\Gamma_1 \triangleright W_1 \simeq \Gamma_2 \triangleright W_2$, it follows from Proposition 4.14 that $\Gamma_2 \vdash c : \mathbf{exp}$. Further, by Lemma 4.1 it cannot be $\Gamma_2 \vdash_t c : n$ for some $n \geq 2$, for otherwise it would not be possible to derive the reduction (7) above. Hence $\Gamma_2 \vdash_t c : 1$. Similarly, we have that $\Gamma_2 \vdash_v c : v$, for again it would not be possible to derive the reduction (7) above.

Now since $\Gamma_2 \triangleright W_2 \stackrel{\sigma}{\Longrightarrow} \Gamma_2' \triangleright W_2'$ and $\Gamma_2' = (1, v)$ we have a required matching move $\Gamma_2 \triangleright W_2 \stackrel{\gamma(c,v)}{\Longrightarrow} \Gamma_2' \triangleright W_2'$. This move is indeed matching because we already know that $\Gamma_1' \triangleright W_1' \mid T_2 \simeq \Gamma_2' \triangleright W_2' \mid T_2$, and from a straightforward generalisation of Proposition 4.15 it follows that $\Gamma_1' \triangleright W_1' \simeq \Gamma_2' \triangleright W_2'$. □

Proposition 4.19 [Preserving Input Actions] Suppose $\Gamma_1 \triangleright W_1 \simeq \Gamma_2 \triangleright W_2$, where both configurations are well-formed. Then $\Gamma_1 \triangleright W_1 \stackrel{c?v}{\longmapsto} \Gamma_1' \triangleright W_1'$ implies $\Gamma_2 \triangleright W_2 \stackrel{c?v}{\Longmapsto} \Gamma_2' \triangleright W_2'$ such that $\Gamma_1' \triangleright W_1' \simeq \Gamma_2' \triangleright W_2'$.

Proof [Outline] The proof of this statement is similar in style to that of Proposition 4.18; in this case it is necessary to exhibit a distinguishing context to detect whether a configuration has performed a (weak) extensional input action. The system term

$$T_{c?v} \stackrel{\text{def}}{=} (c\,!\langle v\rangle.\mathit{eureka}!\langle \mathit{ok}\rangle + \mathit{fail}!\langle \mathit{no}\rangle) \mid \mathit{halt}!\langle \mathit{ok}\rangle$$

serves this purpose. Note that the assumption that $\Gamma_1 \triangleright W_1$ is well-formed is necessary to ensure that, if $\Gamma_1 \triangleright W_1 \stackrel{c?v}{\longmapsto}$, then $\Gamma_1 \triangleright W_1 \mid T_{c?v}$ has a weak barb on channel *eureka*. At least intuitively, such a barb is enabled in $\Gamma_1 \triangleright W_1 \mid T_{c?v}$ only after one instant of time; therefore, we need to ensure that $\Gamma_1 \triangleright W_1$ allows time to pass. □

Proposition 4.20 [Preserving ι-actions] Suppose $\Gamma_1 \triangleright W_1 \simeq \Gamma_2 \triangleright W_2$, where again both are well-formed configurations. Then $\Gamma_1 \triangleright W_1 \stackrel{\iota(c)}{\longmapsto} \Gamma_1' \triangleright W_1'$ implies $\Gamma_2 \triangleright W_2 \stackrel{\iota(c)}{\Longmapsto} \Gamma_2' \triangleright W_2'$ such that $\Gamma_1' \triangleright W_1' \simeq \Gamma_2' \triangleright W_2'$.

Proof [Outline] We proceed again as in Proposition 4.18, this time using the system term $T_{\iota(c)} = ([\exp(c)]\mathsf{nil}, \mathit{eureka}!\langle \mathit{ok}\rangle) + \mathit{fail}!\langle \mathit{no}\rangle \mid \mathit{halt}!\langle \mathit{ok}\rangle$. Again, well-formedness of W_1 is needed to ensure that if $\Gamma_1 \triangleright W_1 \stackrel{\iota(c)}{\longmapsto}$ then $\Gamma_1 \triangleright W_1 \mid T_{\iota(c)}$ has a weak barb on channel *eureka*, for the broadcast along such a channel takes place after one time instants from the beginning of the computation of $\Gamma_1 \triangleright W_1 \mid T_{\iota(c)}$. □

We are now ready to prove the main result of this Section.

Theorem 4.21 [Completeness] On well-defined configurations, reduction barbed congruence implies bisimilarity.

Proof It is sufficient to show that the relation

$$\mathcal{S} \stackrel{\text{def}}{=} \{(\Gamma_1 \triangleright W_1\,,\, \Gamma_2 \triangleright W_2) :\ \ \Gamma_1 \triangleright W_1 \simeq \Gamma_2 \triangleright W_2\}$$

is a bisimulation. To do so we show that for each extensional action α, the relation $\mathcal{S}$ satisfies the corresponding transfer property given in Definition 3.1. The proof of this statement is a simple consequence of the series of propositions we have just developed.

For 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 need to show that $\Gamma_2 \triangleright W_2 \stackrel{\gamma(c,v)}{\Longmapsto} \Gamma_2' \triangleright W_2'$ for some Γ_2', W_2' such that $(\Gamma_1' \triangleright W_1', \Gamma_2' \triangleright W_2') \in \mathcal{S}$, i.e. $\Gamma_1' \triangleright W_1' \simeq \Gamma_2' \triangleright W_2'$. However, this statement is a direct consequence of Proposition 4.18.

For each possible extensional action α we have proven similar propositions which give the required matching move. □

5 Applications

In this section, we show how our calculus CCCP can be used to model different interesting behaviours which arise at the MAC sub-layer [22] of wireless networks. Then, we exploit our bisimulation proof technique to provide examples of behaviourally equivalent networks. In particular we give some examples comparing the behaviour of routing protocols and *Time Division Multiplexing*.

We start with some simple examples. The first show that stations which do not transmit on unrestricted channels can not be detected. To this end we use $\mathrm{fsn}(W)$ to denote the set of unrestricted channel names in the code W which have transmission occurrences, that is occurrences of an unrestricted channel name c the form $c\,!\langle u\rangle.\ldots.$.

Example 5.1 [Unobservable systems] Consider a wireless system in which no station can broadcast on any free channel. Intuitively none of its behaviour should be observable. In CCCP this means that the system should be behaviourally equivalent to the *empty* system nil.

Formally consider the configuration $\Gamma \triangleright \mathsf{nil}$ where Γ is an arbitrary channel environment. This configuration has non-trivial extensional behaviour. For example it is input enabled, and so can perform all extensional actions of the form $c?v$. It can also perform σ actions, indicating the passage of time.

Now let W be arbitrary station code such that $\mathrm{fsn}(W) = 0$, that is it can not broadcast on any free channel. The configuration $\Gamma \triangleright W$ has similar behaviour. Indeed let $\mathcal{S}$ be the relation

$$\{(\Gamma \triangleright W,\ \Gamma \triangleright \mathsf{nil}) \ \mid \ \mathrm{fsn}(W) = 0\}$$

Then it is straightforward to show that $\mathcal{S}$ is a strong bisimulation in the extensional LTS. Our soundness result therefore ensures that

$$\Gamma \triangleright W \ \simeq \ \Gamma \triangleright \mathsf{nil}$$

whenever $\mathrm{fsn}(W) = 0$. □

Next we consider what happens when a channel becomes permanently exposed. . This situation can be modelled by using two stations s_0, s_1 which repeatedly send a value along channel c; each broadcast performed by s_1 takes place before the transmission of s_0 ends, and vice versa. In this case we say that the channel c is *corrupted*. Clearly, if a system transmits only on corrupted channels; then it cannot be detected. Let us see how this scenario is reflected in our behavioural theory.

Example 5.2 [Noise obfuscates transmissions] Let v be a value such that $\delta_v = 2$ and let $\mathrm{Snd}(c)$ denote the code $\mathsf{fix}\, X.c\,!\langle v\rangle.X$, which continually broadcasts v along c. To model the two stations s_0 and s_1 discussed informally above we use the code $\mathrm{Noise}(c) = \mathrm{Snd}(c) \mid \sigma.\mathrm{Snd}(c)$.

Then, consider a configuration $\Gamma \triangleright W$ such that $\mathrm{fsn}(W) \subseteq \{c\}$; that is does not transmit on free channels different from c. Then

$$\Gamma \triangleright W \mid \mathrm{Noise(c)} \ \simeq \ \Gamma \triangleright \mathrm{Noise(c)}$$

To prove this, it is sufficient to exhibit bisimulation containing the pair of configurations ($\Gamma \triangleright W \mid$ Noise(c), $\Gamma \triangleright$ Noise(c)).

We use the following abbreviations:

$$\begin{aligned}\text{Noise}'(c) &= \sigma^2.\text{Snd}(c) \mid \sigma.\text{Snd}(c)\\ \text{Noise}''(c) &= \sigma.\text{Snd}(c) \mid \text{Snd}(c)\\ \text{Noise}'''(c) &= \sigma.\text{Snd}(c) \mid \sigma^2.\text{Snd}(c)\end{aligned}$$

Then let $\mathcal{S}$ denote the following set of pairs of configurations:

$$\begin{array}{l}\{(\Delta \triangleright W \mid \text{Noise(c)} \quad , \quad \Delta' \triangleright \text{Noise(c)}),\\ (\Delta \triangleright W \mid \text{Noise}'(c) \quad , \quad \Delta' \triangleright \text{Noise}'(c)),\\ (\Delta \triangleright W \mid \text{Noise''}(c) \quad , \quad \Delta' \triangleright \text{Noise}''(c)),\\ (\Delta \triangleright W \mid \text{Noise'''}(c) \quad , \quad \Delta' \triangleright \text{Noise}'''(c)) \quad \mid\\ \qquad \Delta, \Delta' \vdash c : \mathbf{exp}, \text{fsn}(W) \subseteq \{c\} \ \}\end{array}$$

Then it is possible to check that $\mathcal{S}$ is a weak bisimulation in the extensional LTS. At least intuitively, this is because in the extensional LTS all outputs fired along the obfuscated channel c corresponds to internal actions; further, in the configurations included in $\mathcal{S}$ channel c is never released, so that neither $\iota(c)$-actions nor $\gamma(c, v)$-actions can be performed by any configuration included in $\mathcal{S}$. □

The *Carrier Sense Multiple Access* (CSMA) scheme [20] is a widely used MAC-layer protocol in which a device senses the channel (*physical carrier sense*) before transmitting. More precisely, if the channel is sensed free the sender starts transmitting immediately, that is in the next instant of time[3]; if the channel is busy, that is some other station is transmitting, the device keeps listening to the channel until it becomes idle and then starts transmitting immediately. This strategy is called *1-persistent* CSMA and can be easily expressed in our calculus in terms of the following process:

$$c!!\langle v\rangle.P = \mathsf{fix}\, X.[\text{exp}(c)]X, c\,!\langle v\rangle.P$$

So, by definition CSMA transmissions are delayed whenever the channel is busy.

In the next example we prove a natural property of CSMA transmissions.

Example 5.3 [Delay in CSMA broadcast] Suppose $\Gamma \vdash_t c : n$ for some $n > 0$. Then, for any $k \leq n + 1$

$$\Gamma \triangleright c!!\langle v\rangle.P \ \simeq \ \sigma^k.c!!\langle v\rangle.P \tag{8}$$

Intuitively, since $\Gamma \vdash_t = n$, the transmission of value v in $\Gamma \triangleright c!!\langle v\rangle.P$ can take place only after at least n instants of time. The same happens in $\Gamma \triangleright \sigma^k.c!!\langle v\rangle.P$.

[3]Recall that in wireless systems channels are half-duplex.

Table 10 A simple topology for a network

.

To prove (8) formally we need to exhibit a bisimulation $\mathcal{S}$ relation in the extensional LTS which contains all pairs of the form $(\Gamma \triangleright c!!\langle v\rangle.P,\ \sigma^k.c!!\langle v\rangle.P)$, where Γ is such that $\Gamma \vdash_t : n > 0$ for some n satisfying $k \leq (n+1)$. One possible $\mathcal{S}$ takes the form $\mathcal{R} \cup \mathcal{I}d$ where $\mathcal{I}d$ is the identity relation over configurations and $\mathcal{R}$ is given by:

$$\mathcal{R} = \quad \{(\Delta_n \triangleright c!!\langle v\rangle.P, \Delta_n \triangleright \sigma^h.c!!\langle v\rangle.P) \mid \quad \mid \Delta_n \vdash_t c : n, h \leq n\}$$

□

In our calculus the network topology is *assumed to be* flat. However, we can exploit the presence of multiple channels to model networks with a more complicated topological structure. The idea is to associate a particular channel with a collection of stations which are in the same neighbourhood.

Example 5.4 [Network Topology] Suppose that we want to model a network with two stations s, r with the following features:

- the range of transmission of s is too short to reach external agents,
- the station r is in the range of transmission of s,
- the range of transmission of r is long enough to also reach external agents.

A graphical representation of the network we want to model is given as $\mathcal{N}_0$ of Table 10. We can model this network topology by using a specific restricted channel, say d, for the local communication between stations s and r. In CCCP a wireless system running on $\mathcal{N}_0$ would therefore take the form

$$C_0 = \Gamma \triangleright \nu d : (0, \cdot).(S \mid R)$$

where

- S represents the code running at station s; it can therefore only broadcast and receive along the restricted channel d (recall that we do not want station s to be able to communicate directly with the external environment)

- R represents the code running at station r; it can only receive values along the restricted channel d (since in $\mathcal{N}_0$ station r can receive messages broadcast by station r, but not by the external environment), while it is free to broadcast on other channels (since station r is able to broadcast messages to the external environment)

As a specific example we could let S denote the single broadcast $d!\langle v\rangle$, and $R = \mathsf{fix}\, X.\lfloor d?(x).c!\langle x\rangle\rfloor X$. Then in the configuration C_0 the station s broadcasts as a value and station r acts as a forwarder; this behaviour is reminiscent of range repeaters in wireless terminology.

Suppose now that we want to add a second station e to the above network topology, so that

- broadcasts from e can be detected by r; this can be accomplished by allowing the process used to model station e to broadcasts along a restricted channel d.
- broadcasts from e can not reach s, nor the external environment. For this to be true, it is sufficient to require that the process which models the behaviour of station e can broadcast values only along the restricted channel d; further, in order for ensuring that the station e cannot detect values broadcast by s, we require that the process used to represent station e does not use receivers along channel d.

The network topology we wish to model is depicted as $\mathcal{N}_1$ in Table 10 and so a wireless system running on this network takes the form

$$C_1 = \nu d{:}(0,\cdot).(S \mid R \mid E)$$

where E is the code running at station e. As an example we could take E to be the faulty code $d!\langle v\rangle + \tau.\mathsf{nil}$.

Then in C_1 station r still acts as a forwarder for station s; however station e can non-deterministically decide whether to corrupt the transmission from node s to r, causing a collision.

Let us assume that the transmission time of the value used in these networks, v, satisfies $\delta_v = \delta_{\mathsf{err}}$. Then we can show

$$C_0 \simeq \Gamma \triangleright \sigma^{\delta_v}.c!\langle v\rangle$$
$$C_1 \simeq \Gamma \triangleright \tau.\sigma^{\delta_v}.c!\langle v\rangle + \tau.\sigma^{\delta_v}.c!\langle \mathsf{err}\rangle$$

Intuitively the reasons for these equivalences are obvious. The transmission along channel d is restricted in C_0, so it cannot be observed by the external environment. The only activity which can be observed is the broadcast of value v along channel c, which takes place after δ_v instants of time. For C_1, a collision can happen along channel d, which is again restricted; the only activity that can be detected by the external environment is a transmission which takes place after δ_v instants of time. Such a transmission will contain either the value v or an error message of length δ_v.

The formal proof of these identities involves exhibiting two bisimulations, containing the relevant pairs of configurations. Here we exhibit a bisimulation for showing that $C_1 \simeq \Gamma \triangleright \tau.\sigma^{\delta_v}.c!\langle v\rangle$. For the sake of simplicity, let $\delta_{\mathsf{err}} = \delta_v = 1$ and define the system terms

$$\begin{array}{lcl} W & = & \nu d : (0,\cdot).(S \mid E \mid R) \\ W_e & = & \nu d : (1,\mathsf{err}).(S \mid \sigma \mid c[x].c!\langle x\rangle) \\ W'' & = & \nu d : (1,\mathsf{err}).(\sigma \mid \sigma \mid c[x].c!\langle x\rangle) \\ W_{\mathsf{err}} & = & \nu d : (0,\cdot).(\mathsf{nil} \mid \mathsf{nil} \mid c!\langle \mathsf{err}\rangle) \end{array}$$

$$\begin{array}{lcl} W_s & = & \nu d : (1,v).(\sigma \mid E \mid c[x].c!\langle x\rangle) \\ W' & = & \nu d : (0,\cdot).(S \mid \mathsf{nil} \mid R) \\ W_{\mathrm{ok}} & = & \nu d : (0,\cdot).(\mathsf{nil} \mid \mathsf{nil} \mid c!\langle v\rangle) \\ W_c & = & \nu d : (0,\cdot).(\mathsf{nil} \mid \mathsf{nil} \mid \sigma) \end{array}$$

Then it is easy to show that the relation

$$\begin{array}{rcll}
\mathcal{S} & = & \{ & (\Delta \triangleright W \quad , \quad \Delta \triangleright \tau.\sigma.c!\langle v\rangle + \tau.\sigma.c!\langle \mathsf{err}\rangle) \quad , \\
& & & (\Delta \triangleright W_s \quad , \quad \Delta \triangleright \tau.\sigma.c!\langle v\rangle + \tau.\sigma.c!\langle \mathsf{err}\rangle) \quad , \\
& & & (\Delta \triangleright W_e \quad , \quad \Delta \triangleright \sigma.c!\langle \mathsf{err}\rangle) \quad , \\
& & & (\Delta \triangleright W' \quad , \quad \Delta \triangleright \sigma.c!\langle v\rangle) \quad , \\
& & & (\Delta \triangleright W'' \quad , \quad \Delta \triangleright \sigma.c!\langle \mathsf{err}\rangle) \quad , \\
& & & (\Delta \triangleright W_{\mathrm{ok}} \quad , \quad \Delta \triangleright c!\langle v\rangle) \quad , \\
& & & (\Delta \triangleright W_{\mathsf{err}} \quad , \quad \Delta \triangleright c!\langle \mathsf{err}\rangle) \quad , \\
& & & (\Delta \triangleright W_c \quad , \quad \Delta \triangleright \sigma) \quad | \\
& & | & \Delta \quad \text{arbitrary channel environment} \quad \}
\end{array}$$

is a weak bisimulation. □

As a final example let us consider how the TDMA modulation technique [45] 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 his 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.

As a simple example let us describe how two messages v_0 and v_1 can be delivered in TDMA style; for simplicity, we assume $\delta_{v_0} = \delta_{v_1} = 2$. The main idea here is to split each of these values 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 time instant to be transmitted, and a binary operator $\circ$ for composing messages such that

$$\begin{array}{rcl}
v_0^0 \circ v_0^1 & = & v_0 \\
v_1^0 \circ v_1^1 & = & v_1 \\
v \circ \mathsf{err} & = & \mathsf{err} \circ v = \mathsf{err}
\end{array}$$

where v is an arbitrary value; in this case we assume that $\delta_{\mathsf{err}} = 2$.

More specifically, for this example we assume four different 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. The network we consider for modelling the TDMA transmission is then given by

$$C_0 = \Gamma \triangleright \nu d{:}(0,\cdot)(\hat{S}_0 \mid \hat{S}_1 \mid \hat{R}_0 \mid \hat{R}_1)$$

where

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

Table 11 Two transmitting stations using different time slots to broadcast values

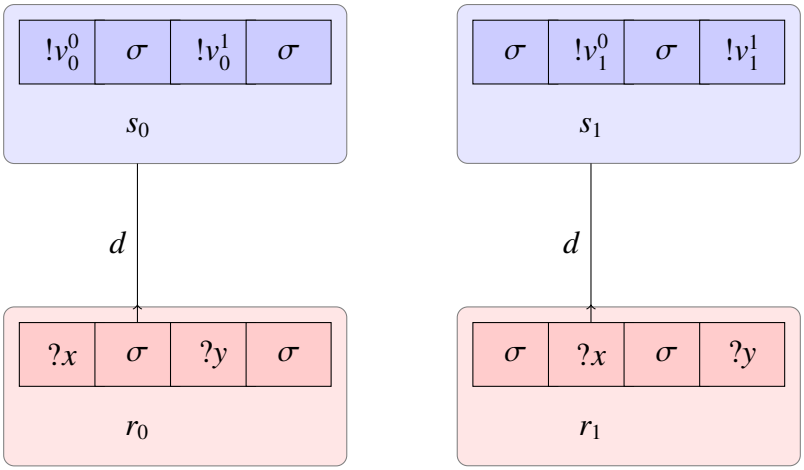
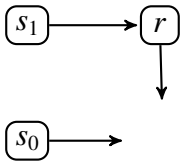


Table 12 Forwarding two messages to the external environment



The intuitive behaviour of this network is depicted in Table 11. Station s_0 wishes to broadcast value v_0, while s_1 wishes to broadcast value v_1. They both use the same (restricted) channel d to broadcast their respective values; however, both stations split the value to be broadcast in two packets. Value v_0 is split in v_0^0 and v_0^1, while v_1 is split in v_1^0 and v_1^1.

The two stations run a TDMA protocol with a time frame of length two. Station s_0 takes control of the first time frame, hence transmitting its two packets v_0^0 and v_0^1 in the first and the third time slot, respectively. Station s_1 takes control of the second time frame; hence the two packets v_1^0 and v_1^1 are broadcast in the second and fourth time slot, respectively.

Stations r_0 and r_1 wait to collect the values broadcast along channel d. However, the former is interested only in packets sent in the first time frame, while the latter detects only values sent in the second time frame. At the end of their associated time frame the stations r_0 and r_1 have received two packets which 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.

As an alternative to TDMA, the two values v_0, v_1 can be also be delivered to the external environment by means of a simple routing, along the lines suggested in Example 5.4. Here we consider the configuration

$$C_1 = \Gamma \triangleright \nu d{:}(0,\cdot).(S_0 \mid S_1 \mid R)$$

where

$$\begin{aligned} S_0 &= \sigma^4.c!\langle v_0\rangle \\ S_1 &= \sigma^4.d!\langle v_1\rangle \\ R &= d?(x).c!\langle x\rangle \end{aligned}$$

Intuitively, the configuration C_1 models three wireless stations s_0, s_1, r, running the code S_0, S_1, R, respectively, and connected as in Table 12. Station s_0 waits four instants of time, then it broadcasts value v_0 directly to the external environment via the free channel c. Similarly, after four instants of time the station s_1 broadcasts value v_1 to station r via the restricted channel d. Finally, r forwards the message to the external environment via the free channel c.

From the point of view of the external environment the configuration C_1 performs the following activities:

- it remains idle for the first four instants of time
- it transmits value v_0 in the fifth and sixth time instants
- it transmits value v_1 in the seventh and eighth time instants.

In this manner, at least informally the observable behaviour of C_1, which uses direct routing, is the same as that of C_0, which uses TDMA.

Formally, we can prove

$$C_0 \simeq C_1 \tag{9}$$

However, instead of proving this by giving a bisimulation containing this pair of configurations, instead we prove them individually bisimilar to a simpler specification. Let $\mathcal{S}_1$ denote the configuration $\Gamma \triangleright S_1$ where S_1 is the code

$$\sigma^4.c\,!\langle v_0\rangle.c\,!\langle v_1\rangle.$$

Then we can show that $C_0 \approx \mathcal{S}_1$ and $C_1 \approx \mathcal{S}_1$, from which (9) follows by soundness. Below we provide a bisimulation for showing that $C_1 \simeq \mathcal{S}_1$; for the sake of simplicity, define the following terms:

$$\begin{aligned} S_0^n &= \sigma^n.c!\langle v_0\rangle & \qquad S_1^n &= \sigma^n.d!\langle v_1\rangle \\ R' &= d[x].c!\langle x\rangle & \qquad W_n &= \sigma^n.c\,!\langle v_0\rangle.c!\langle v_1\rangle \end{aligned}$$

for any $n \in \mathbb{N}$. Then the relation

$$\begin{aligned} \mathcal{R} = \{\ & (\Delta \triangleright \nu d : (0,\cdot).(S_0^n \mid S_1^n \mid R) && , && \Delta \triangleright W_n) && , \\ & (\Delta \triangleright \nu d : (0,\cdot).(\sigma^2 \mid d!\langle v_1\rangle \mid R) && , && \Delta \triangleright \sigma^2.c!\langle v_1\rangle) && , \\ & (\Delta \triangleright \nu d : (2,v_1).(c!\langle v_0\rangle \mid \sigma^2 \mid R') && , && \Delta \triangleright c\,!\langle v_0\rangle.c!\langle v_1\rangle) && , \\ & (\Delta \triangleright \nu d : (2,v_1).(\sigma^2 \mid \sigma^2 \mid R') && , && \Delta \triangleright \sigma^2.c!\langle v_1\rangle) && , \\ & (\Delta \triangleright \nu d : (1,v_1).(\sigma \mid \sigma \mid R') && , && \Delta \triangleright \sigma.c!\langle v_1\rangle) && , \\ & (\Delta \triangleright \nu d : (0,\cdot).(\mathsf{nil} \mid \mathsf{nil} \mid c!\langle v_1\rangle) && , && \Delta \triangleright c!\langle v_1\rangle) && \mid \\ \mid\ & \Delta \text{ arbitrary channel environment} && && && \} \end{aligned}$$

is a relation which contains the most relevant couples needed for showing that $C_1 \approx S_1$.

As a final example we can modify the behaviour of the two configurations C_0 and C_1 by adding the possibility of getting a *collision* when delivering values v_0, v_1 to the external environment. In the routing case, this is accomplished by requiring that both stations s_0, s_1 can either broadcast their value directly to the external environment or to the forwarder node r, while in the TDMA case it is sufficient to allow both the stations s_0, s_1 to non-deterministically choose the time slot to be used to broadcast packets.

To this end, let

$$\begin{array}{rcl} S_0^{\mathsf{c}} & = & \tau.\sigma^4.c!\langle v_0\rangle + \tau.\sigma^4.d!\langle v_0\rangle \\ S_1^{\mathsf{c}} & = & \tau.\sigma^4.c!\langle v_1\rangle + \tau.\sigma^4.d!\langle v_1\rangle \\ \hat{S}_0^{\mathsf{c}} & = & d\,!\langle v_0^0\rangle.\sigma.d!\langle v_0^1\rangle + \tau.\sigma.d\,!\langle v_0^0\rangle.\sigma.d!\langle v_0^1\rangle \\ \hat{S}_1^{\mathsf{c}} & = & d\,!\langle v_1^0\rangle.\sigma.d!\langle v_1^1\rangle + \tau.\sigma.d\,!\langle v_1^0\rangle.\sigma.d!\langle v_1^1\rangle \end{array}$$

and consider the configurations

$$\begin{array}{rcl} C_1^{\mathsf{c}} & = & \Gamma \triangleright \nu d{:}(0,\cdot).(S_0^{\mathsf{c}} \mid S_1^{\mathsf{c}} \mid R) \\ C_0^{\mathsf{c}} & = & \Gamma \triangleright \nu d{:}(0,\cdot).(\hat{S}_0^{\mathsf{c}} \mid \hat{S}_1^{\mathsf{c}} \mid \hat{R}_0 \mid \hat{R}_1) \end{array}$$

It is not difficult to see informally that the observable behaviour of these two configurations is the same. Specifically

- either value v_0 is broadcast in the fifth and sixth time slots and v_1 is broadcast in the seventh and eighth instants of time slots, or
- value v_1 is broadcast in the fifth and sixth time slots, while value v_0 is broadcast in the seventh and eighth time slots, or
- a collision occur in the fifth and sixth time slots, or
- a collision occur in the seventh and eighth time slots.

This informal behaviour can be described by the term

$$\begin{array}{rcll} S_2 & = & \tau.\sigma^4.c\,!\langle v_0\rangle.c!\langle v_1\rangle & + \\ & & \tau.\sigma^4.c\,!\langle v_1\rangle.c!\langle v_0\rangle & + \\ & & \tau.\sigma^4.c!\langle \mathsf{err}\rangle & + \\ & & \tau.\sigma^6.c!\langle \mathsf{err}\rangle & \end{array}$$

and once more we can exhibit bisimulations to establish $\Gamma \triangleright S_2 \approx C_0^{\mathsf{c}}$ and $\Gamma \triangleright S_2 \approx C_1^{\mathsf{c}}$. Then soundness again ensures that

$$C_0^{\mathsf{c}} \simeq C_1^{\mathsf{c}}$$

6 Conclusions

In this paper we have given a behavioural theory of wireless systems at the MAC level. In our framework individual wireless stations broadcast information to their neighbours along virtual channels. These broadcasts take a certain amount of time to complete, and are subject to collisions. If a broadcast is successful a recipient may choose to ignore the information it contains, or may act on it, in turn generating further broadcasts. We believe that our reduction semantics, given in Section 2, captures much of the subtlety of intensional MAC-level behaviour of wireless systems.

Then based on this reduction semantics we defined a natural contextual equivalence between wireless systems which captures the intuitive idea that one system can be replaced by another in a larger network without affecting the observable behaviour of the original network. In the main result of the paper, we then gave a sound and complete characterisation of this behavioural equivalence in terms of *extensional actions*. This characterisation is important for two reasons. Firstly it gives an understanding of which aspects of the intensional behaviour is important from the point of view of external users of these wireless systems. Secondly it gives a powerful sound and complete co-inductive proof method for demonstrating that two systems are behaviourally equivalent. We have also demonstrated the viability of this proof methodology by a series of examples.

Let us now examine some relevant related work. We start with the literature on process calculi for wireless systems. Nanz and Hankin [33] 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 [28] has proposed an untimed process calculus for mobile ad-hoc networks with a labelled characterisation of reduction barbed congruence, while [13] contains a calculus called CMAN, also with mobile ad-hoc networks in mind. This latter paper also gives a characterisation of reduction barbed congruence, this time in terms of a contextual bisimulation. It also contains a formalisation of an attack on the cryptographic routing protocol ARAN.

Singh, Ramakrishnan and Smolka [44] 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. The authors provide a labelled transition semantics and a bisimulation in *open* style. The ω-calculus is then used for modelling the AODV routing protocol. Another extension of the π-calculus for modelling, called LUNAR, may be found in [7] which is used to model ad-hoc routing protocols.

In [8] 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. [11] 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. They propose a notion of bisimulation for networks parametrised on a set of topological invariants that must be respected by equivalent networks. This work in then refined in [12] where the authors propose an equational theory for an extension of RBPT. Godskesen and Nanz [14] have proposed a simple timed calculus for wireless systems to express a wide range of mobility models. Kouzapas and Philippou [23] 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.

All the calculi mentioned up to now abstract away from the from the possibility of interference between broadcasts. Lanese and Sangiorgi [24] have instead proposed the CWS calculus, a lower level untimed calculus to describe interferences in wireless systems. In their operational semantics there is a separation between the beginning and ending of a broadcast, so there is some implicit representation of the passage of time. A more explicit timed generalisation of CWS is given [29] to express MAC-layer protocols such as CSMA/CA, where the authors propose a bisimilarity which is proved to be sound but not complete with respect to a notion of reduction barbed congruence. We view the current paper as a simplification and generalisation of [29].

The research we have mentioned so far has been focused on formalising various aspects of ad-hoc networks. However other than [14, 29], these various calculi abstract away from time. Nevertheless there is an extensive literature on timed process algebras, which we now briefly review. From a purely syntactic point of view, the earliest proposals are extensions of the three main process algebras, ACP, CSP and CCS. For example, [2] presents a real-time extension of ACP, [40] contains a denotational model for a timed extension of CSP, while CCS is the starting point for [32]. In [2] and [40] time is real-valued, and at least semantically, associated directly with actions. The other major approach to representing time is to introduce a special action to model the passage of time, and to assume that all other actions are instantaneous. This approach is advocated in [15, 5, 32, 35] and [46, 47], although the basis for this approach may be found in [6]. The current paper shares many of the assumptions of the languages presented in these papers; in particular we have been influenced by [18] which contains a timed version of CCS enjoying time determinism, maximal progress and patience. All the just mentioned papers assume that actions are instantaneous and only the extension of ACP presented in [15] does not incorporate time determinism; however maximal progress is less popular and patience is even rarer.

From this early work on timed process calculi a flourishing literature has emerged. Here we briefly mention some highlights of this research. Prasad [37] has proposed a timed variant of his CBS [36], called TCBS. In TCBS a timeout can force a process wishing to speak to remain idle for a specific interval of time; this corresponds to have a priority. TCBS also assumes time determinism and maximal progress. Corradini et al. [9] deal with *durational actions* proposing a framework relying on the notions of reduction and observability to naturally incorporate timing information in terms of process interaction. Our definition of timed reduction barbed congruence takes inspiration from theirs. Corradini and Pistore [10] have studied durational actions to describe and reason about the performance of systems. Actions have lower and upper time bounds, specifying their possible different durations. Their *time equivalence* refines the untimed one. Baeten and Middelburg [3] consider a range timed process algebras within a common framework, related by embeddings and conservative extensions relations. These process algebras, $\mathrm{ACP}^{\mathrm{sat}}$, $\mathrm{ACP}^{\mathrm{srt}}$, $\mathrm{ACP}^{\mathrm{dat}}$ and $\mathrm{ACP}^{\mathrm{drt}}$, allow the execution of two or more actions consecutively at the same point in time, separate the execution of actions from the passage of time, and consider actions to have no duration. The process algebra $\mathrm{ACP}^{\mathrm{sat}}$ is a real-time process algebra with absolute time, $\mathrm{ACP}^{\mathrm{srt}}$ is a real-time process algebra with relative time. Similarly, $\mathrm{ACP}^{\mathrm{dat}}$ and $\mathrm{ACP}^{\mathrm{drt}}$ are discrete-time process algebras with absolute time and relative time, respectively. In these process algebra the focus is on

unsuccessful termination or deadlock. In [4] Baeten and Reniers extend the framework of [3] to model successful termination for the relative-time case. Laneve and Zavattaro [25] have proposed a timed extension of π-calculus where time proceeds asynchronously at the network level, while it is constrained by the local urgency at the process level. They propose a timed bisimilarity whose discriminating is weaker when local urgency is dropped.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks (Amsterdam, Netherlands: 1999)*, 38(4):393–422, March 2002.

[2] J. Baeten and J. Bergstra. Real Time Process Algebra. *Formal Aspects of Computing*, 3(2):142–188, 1991.

[3] J. Baeten and C. Middelburg. *Process Algebra with Timing*. EATCS Series. Springer-Verlag, 2002.

[4] J. C. M. Baeten and M. A. Reniers. Timed Process Algebra (With a Focus on Explicit Termination and Relative-Timing). In *SFM*, volume 3185 of *Lecture Notes in Computer Science*, pages 59–97. Springer-Verlag, 2004.

[5] J.C.M. Baeten and J.A. Bergstra. Discrete time process algebra. *Formal Aspects of Computing*, 8(2):188–208, 1996.

[6] G. Berry and L. Cosserat. The ESTEREL Synchronous Programming Language and its Mathematical Semantics. Technical Report 842, INRIA, Sophia-Antipolis, 1988.

[7] Johannes Borgström, Shuqin Huang, Magnus Johansson, Palle Raabjerg, Björn Victor, Johannes Åman Pohjola, and Joachim Parrow. Broadcast psi-calculi with an application to wireless protocols. In *SEFM*, volume 7041 of *Lecture Notes in Computer Science*, pages 74–89. Springer, 2011.

[8] Andrea Cerone and Matthew Hennessy. Modelling probabilistic wireless networks (extended abstract). In Holger Giese and Grigore Rosu, editors, *Proceedings of the 14th Annual International Conference on Formal Methods for Open Object-Based Distributed System FMOODS 2012, and the 32th Annual International Conference on Formal Techniques for Networked and Distributed Systems FORTE 2012, Stockholm, Sweden, June 13-16 2012*. Springer, 2012.

[9] F. Corradini, G. Ferrari, and M. Pistore. On the semantics of durational actions. *Theoretical Computer Science*, 269(1-2):47–82, 2001.

[10] F. Corradini and M. Pistore. Closed interval process algebra versus interval process algebra. *Acta Informatica*, 37(7):467–509, 2001.

[11] F. Ghassemi, W. Fokkink, and A. Movaghar. Restricted Broadcast Process Theory. In *SEFM*, pages 345–354. IEEE Computer Society, 2008.

[12] F. Ghassemi, W. Fokkink, and A. Movaghar. Equational Reasoning on Ad Hoc networks. In *FSEN*, volume 5961 of *Lecture Notes in Computer Science*, pages 113–128. Springer, 2009.

[13] J.C. Godskesen. A Calculus for Mobile Ad Hoc Networks. In *COORDINATION*, volume 4467 of *Lecture Notes in Computer Science*, pages 132–150. Springer Verlag, 2007.

[14] Jens Chr. Godskesen and Sebastian Nanz. Mobility Models and Behavioural Equivalence for Wireless Networks. In *COORDINATION*, volume 5521 of *Lecture Notes in Computer Science*, pages 106–122. Springer, 2009.

[15] J.F. Groote. Specification and Verification of Real Time Systems in acp. In *PSTV*, pages 261–274. North-Holland, 1990.

[16] Hennessy and Rathke. Bisimulations for a calculus of broadcasting systems. *TCS: Theoretical Computer Science*, 200(1–2):225–260, 1998.

[17] Matthew Hennessy. *A distributed Pi-calculus*. Cambridge University Press, 2007.

[18] Matthew Hennessy and Tim Regan. A process algebra for timed systems. *Information and Computation*, 117(2):221–239, March 1995.

[19] K. Honda and N. Yoshida. On reduction-based process semantics. *Theoretical Computer Science*, 152(2):437–486, 1995.

[20] IEEE 802.11 WG. ANSI/IEEE standard 802.11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications. IEEE Computer Society, 2007.

[21] A. Jeffrey and J. Rathke. Contextual equivalence for higher-order pi-calculus revisited. *Logical Methods in Computer Science*, 1(1:4), 2005.

[22] 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.

[23] Dimitrios Kouzapas and Anna Philippou. A process calculus for dynamic networks. In *FMOODS/FORTE*, volume 6722 of *Lecture Notes in Computer Science*, pages 213–227. Springer, 2011.

[24] Ivan Lanese and Davide Sangiorgi. An operational semantics for a calculus for wireless systems. *Theor. Comput. Sci*, 411(19):1928–1948, 2010.

[25] C. Laneve and G. Zavattaro. Foundations of web transactions. In *FoSSaCS*, volume 3441 of *Lecture Notes in Computer Science*, pages 282–298. Springer, 2005.

[26] Nancy A. Lynch. Input/output automata: Basic, timed, hybrid, probabilistic, dynamic, .. In Roberto M. Amadio and Denis Lugiez, editors, *CONCUR*, volume 2761 of *Lecture Notes in Computer Science*, pages 187–188. Springer, 2003.

[27] Damiano Macedonio and Massimo Merro. A semantic analysis of wireless network security protocols. In Alwyn Goodloe and Suzette Person, editors, *NASA Formal Methods*, volume 7226 of *Lecture Notes in Computer Science*, pages 403–417. Springer, 2012.

[28] M. Merro. An Observational Theory for Mobile Ad Hoc Networks (full paper). *Information and Computation*, 207(2):194–208, 2009.

[29] Massimo Merro, Francesco Ballardin, and Eleonora Sibilio. A timed calculus for wireless systems. *Theor. Comput. Sci.*, 412(47):6585–6611, 2011.

[30] Milner and Sangiorgi. Barbed bisimulation. In *ICALP: Annual International Colloquium on Automata, Languages and Programming*, 1992.

[31] R. Milner. *Communication and Concurrency*. Prentice-Hall, 1989.

[32] F. Moller and C. Tofts. A Temporal Calculus of Communicating Systems. In *CONCUR*, volume 458 of *Lecture Notes in Computer Science*, pages 401–415. Springer Verlag, 1990.

[33] S. Nanz and C. Hankin. A Framework for Security Analysis of Mobile Wireless Networks. *Theoretical Computer Science*, 367(1-2):203–227, 2006.

[34] Sebastian Nanz and Chris Hankin. Static analysis of routing protocols for ad-hoc networks, March 25 2004.

[35] Xavier Nicollin and Joseph Sifakis. The algebra of timed processes, atp: Theory and application. *Inf. Comput.*, 114(1):131–178, 1994.

[36] K. V. S. Prasad. A calculus of broadcasting systems. *Science of Computer Programming*, 25(2–3):285–327, December 1995. ESOP '94 (Edinburgh, 1994).

[37] K.V.S. Prasad. Broadcasting in Time. In *COORDINATION*, volume 1061 of *Lecture Notes in Computer Science*, pages 321–338. Springer Verlag, 1996.

[38] Theodore S. Rappaport. *Wireless communications - principles and practice*. Prentice Hall, 1996.

[39] Julian Rathke and Pawel Sobocinski. Deconstructing behavioural theories of mobility. In *Proc. Fifth IFIP International Conference On Theoretical Computer Science (TCS)*, volume 273 of *IFIP*, pages 507–520. Springer, 2008.

[40] G.M. Reed. A Hierarchy of Domains for Real-Time Distributed Computing. Technical Report, Oxford, 1988.

[41] D. Sangiorgi, N. Kobayashi, and E. Sumii. Environmental bisimulations for higher-order languages. In *Proceedings of the 22nd IEEE Symposium on Logic in Computer Science*, pages 293–302. IEEE Computer Society, 2007.

[42] D. Sangiorgi and D. Walker. *The π-calculus: a Theory of Mobile Processes*. Cambridge University Press, 2001.

[43] Davide Sangiorgi and David Walker. *The Pi-Calculus — A Theory of Mobile Processes*. Cambridge University Press, 2001.

[44] A. Singh, C.R. Ramakrishnan, and S.A. Smolka. A Process Calculus for Mobile Ad Hoc Networks. In *COORDINATION*, volume 5052 of *Lecture Notes in Computer Science*, pages 296–314, 2008.

[45] Andrew S. Tanenbaum. *Computer Networks, 4th ed.* Prentice-Hall International, Inc., 2003.

[46] W. Yi. Real-Time Behaviour of Asynchronous Agents. In *CONCUR*, volume 458 of *Lecture Notes in Computer Science*, pages 502–520. Springer Verlag, 1990.

[47] W. Yi. *A Calculus of Real Time Systems*. Ph.D Thesis, Chalmers University, 1991.