MFPS 2006

Towards a calculus for Wireless Systems

Nicola Mezzetti and Davide Sangiorgi[1]

University of Bologna, Italy

Abstract

In wireless systems, the communication mechanism combines features of broadcast, synchrony, and asynchrony. We develop an operational semantics for a calculus of wireless systems. We present a Reduction Semantics and a Labelled Transition Semantics and prove a correspondence result between them. We first consider a core calculus, essentially with only the primitives for communication, and then a few extensions.

A major goal of the semantics is to describe the forms of interferences among the activities of processes that are peculiar of wireless systems. Such interferences occur when a location is simultaneously reached by two transmissions.

Key words: Wireless systems, Broadcast, Reduction Semantics, Labelled Transition Semantics, Interferences

---

1 Motivation of the work

Wireless technology is becoming everyday more widespread; its applications span from user applications, such as personal area networks, ambient intelligence, and wireless local area networks, to real-time applications, such as cellular networks for mobile telephony or sensor networks.

The goal of this work is to define a basic calculus of wireless systems, and its associated operational semantics. The most important contribution of this work has been the semantics. Here we had to deal with the peculiarities of wireless systems, which combine features of broadcast, synchrony, and asynchrony, as we outline below.

Wireless devices communicate by *broadcast* of messages. This is however quite different from the more conventional wired-based broadcast that we find in networks with Ethernet and that, from a semantic point of view, is well-understood [Pra95a,Pra96]. First, in Ethernet-like systems broadcasting is

---

[1] Supported by the EC project "SENSORIA" and Italian MIUR Project n. 2005015785 "Logical Foundations of Distributed Systems and Mobile Code".

This paper is electronically published in
Electronic Notes in Theoretical Computer Science
URL: `www.elsevier.nl/locate/entcs`

global, i.e., the messages transmitted reach all nodes of the system. By contrast, in wireless system broadcasting is local, i.e., a transmission spans over a limited area, called *cell*, and therefore reaches only a – possibly empty – subset of the devices in the system. Second, the communication channel for an Ethernet device is full-duplex; that is, a node can transmit and receive at the same time. As a consequence of full-duplex channels and global broadcasting, interferences between two simultaneous transmissions are immediately detected by transmitters. Thus transmitters know that they have to retransmit their messages, and they do so after a randomly-chosen period of time. This means that interferences in Ethernet-like systems are easy to repair. In a model of these systems it is therefore reasonable to abstract from interferences, i.e., to assume that they do not exist. By contrast, in wireless systems channels are *half-duplex*: on a given channel, a device can either transmit or receive, but cannot do both at the same time. Hence, an interference between two transmissions is only possibly detected by receivers located in the intersection of the cells of the two transmitters. Interference is thus a delicate aspect of wireless systems that is handled by means of ad-hoc protocols (e.g., CDMA/CA). Interference is also one of the key aspects to be described by a model of these systems. A third, but semantically less relevant, difference between Ethernet and wireless broadcast is that only the latter uses explicit communication channels.

Wireless systems have also features of *synchrony* that remind us of synchronous languages (e.g., Esterel [BG92], Statecharts [Har87], SCCS [Mil83]). Indeed, in a single time unit of a wireless system multiple events can happen; such a simultaneous execution of the events is different from an interleaving of them: only in the former case, for instance, interferences can appear.

Wireless devices locally try to remain synchronized on a common clock so that, for instance, transmissions start at the same time. This is due to the fact that each device incorporates a hardware clock that is employed for physical transmission; however, each hardware clock is generally characterized by a different clock drift with respect to the nominal frequency. Hence, to reduce the effect of the clock drift, each device re-synchronizes its own clock on the transmissions which are performed from the other devices. However, the physical distance among the devices and the partial area covered by each transmission may prevent an exact global agreement on clocks. As a consequence, the clocks of distant devices usually do not coincide. This introduces an additional element of *asynchrony* that affects the way interferences appear and are detected, as we discuss below.

Another – though less important – feature of wireless systems (that a semantics should show) is the fact that a transmitter, before initiating a transmission, checks that, locally, the communication channel (i.e. a specific frequency band) is not presently employed for performing other transmissions. This is imposed in order to reduce the possibilities of interferences.

Our calculus is called *Calculus of Wireless Systems* (CWS). It has nodes, which represent the devices of the system, that can be composed in parallel. Inside a node there is a sequential process, which models the behaviour of that device. Each node has a location and a radius that define the cell over which that node can transmit. When developing the semantics for CWS we tried hard to adhere to the standard operational semantics of concurrent systems (broadcast systems as in CBS [Pra95b,Pra96,OPT02], point-to-point systems as CCS [Mil89] or CSP [BHR84]) in which each transmission of a message (or communication) is represented precisely by a single transition (or reduction) in the semantic. For instance, in the labelled transition semantics of CCS an event $P \xrightarrow{\overline{a}v} P'$ means that an output process in P transmits value v on channel a and then the process evolves into P'. Only one among the input processes executing in parallel with P receives that value. Similarly, in CBS an event $P \xrightarrow{!v} P'$ means that an output process in P transmits value v and then the process evolves into P'. All the input processes executing in parallel with P catch that transmission.

Our attempt at following this approach was unsuccessful due to the physical aspects of wireless systems outlined above. In particular, we could not maintain transmission atomicity and, at the same time, express all the possible modes in which interferences may occur.

A natural alternative would have been to follow the approach of synchronous languages. In these languages an event in the semantics represents a set of transmissions, namely all those that occurred during the same time unit. Unfortunately, we failed also with this approach. The reason is that – as explained earlier – the synchrony among the devices of a wireless system is not perfect: the transmission of two distant devices may span time units that are only partially overlapping, with strong consequences on the interference that these devices can cause with each other and with other devices.

We thus decided to refine the view on transmissions and observe, for each node, the change of state between transmission and reception (and vice versa), rather then single transmissions. Further, in accordance with physical wireless devices, we assume that when a device is not performing a transmission its antenna is in reception mode. We call *event* the state change that occurs in the network when a device changes the function of its antenna. Specifically, we call *begin transmission* the event which corresponds to a device which initiates a transmission; and, we call *end transmission* the event which corresponds to a transmitter which finishes its transmission. This choice has also physical justifications: wireless communication protocols require different signal sequences for indicating begin and end of transmissions.

In concurrency theory, *Labelled Transition Systems* (LTS) are the most widely adopted means of giving operational semantics. An LTS can be a basis for defining powerful proof techniques: both inductive techniques, because, for instance, the derivation of a transition in the LTS is driven by the structure of a term; and co-inductive techniques, because, for instance, the transitions of an LTS expose the full behaviour of a system (its internal activities as well as the interactions with the environment) which is needed for defining behavioural equivalences according to the notion of bisimulation. However, sometimes the rules of the LTS can be hard to understand. This happens, for instance, in calculi for mobility such as the π-calculus or in higher-order calculi such as Ambients. Thus, a different form of operational semantics, called *Reduction Semantics* (RS), has been introduced [Mil91]. An RS only gives meaning to the internal activities of a system. For this, the structural operation semantics style is combined with an auxiliary relation of structural congruence that allows manipulation of the term structure so to bring potential interacting components in contiguous positions. An RS can make the meaning of the a calculus or language easier to grasp; and it can be used to check the correctness of an LTS, by proving a consistency with respect to such LTS. On the other hand, the LTS remain superior for defining reasoning techniques on the processes.

For reasons similar to those above, we define both RS and LTS semantics for CWS. Besides the different flavor (the fact of being RS or LTS), the two semantics also differ in the approach we followed to check the interferences. In the LTS, the derivation of a transition takes a set of active transmitters (i.e., a set of devices that are currently engaged in a transmission) as a parameter; then, the various possibilities of interferences are checked against such a parameter. In the RS, by contrast, such parameter is absent; the checks for interferences are "hardwired" into the rules of the semantics itself. As a consequence, however, the derivation of a reduction has to be decomposed into three separate sub-derivations, which are defined using some auxiliary relations on an extension of the calculus. Each one of such auxiliary relations implements a logical distinct sub-component of the mechanism with which transmissions are performed in wireless systems (e.g., individuation of the transmitter; individuation of its cell; for each receiver in the cell, the individuation of possible interferences). The main extensions of the calculus are given by markers that are placed on the network nodes and that represent a partial state of the node within the whole reduction.

Our main technical result is the equivalence between the two semantics. The proof differs from existing results of this kind in the literature both because of the differences between standard LTSs and RSs and ours, and because of the different approaches used in our LTS and RS concerning the identification of the active nodes.

At present, CWS does not model the movement of devices. This choice simplifies the syntax and the semantics of the calculus. Further, movement is not relevant in important classes of wireless systems, most notably sensor networks (not all sensor networks are static, but the static case is predominant). A sensor network ([ASSC02,AK04]) is a computer network of many, spatially distributed devices using sensors to monitor conditions (e.g., temperature, sound, vibration, pressure, motion) at different locations. Each device is equipped with a radio transceiver, a small micro-controller, and an energy source, usually a battery. Sensors communicate employing a broadcast-based wireless data-link protocol (i.e., 802.15.4) to collaboratively transport, and possibly aggregate, the collected information to a monitoring computer. Sensor networks are applied in a wide variety of areas: traffic monitoring and video surveillance [IBK+05,Est01,MCP+02,SPMC04], disaster recovery [WLLP01,CDM+03,Nik06], home monitoring [Kev05a], manufacturing and industrial automation [inc04,Goo04,Kev05b,Sys04,SNL05], etc..

We discuss the addition of movement to CWS in Section 7.

Related Work

Probabilistic simulation is currently the main validation technique used for verification of wireless systems: given a system, pseudo-random number generators are employed to select a finite subset of the possible execution traces and then statistical analysis techniques are applied to obtain probabilistic information on that system.

We discuss below the works on process calculi that are most closely related to ours. Broadcast has been first analysed by Prasad [Pra95b,Pra96,OPT02] who models Ethernet-like communications. Recently, CBS# ([NH04,NH05]) has been proposed as a calculus for mobile wireless networks. The major difference with our work is that CBS# abstracts from interferences; that is, a value transmitted is correctly received by all receivers in the transmission cell of the sending node. Moreover, in CBS# each event represents a complete transmission (rather its boundary as in our case). Such abstractions can make CBS# interesting for reasoning on high level properties and applications. We chose to model a wireless system at a lower level, name that of data-link, which is the lowest level at which wireless systems can be programmed. At this level interferences are an essential aspect (indeed, the presence of collisions non resolvable by hardware is one of the most original features of wireless networking). Another difference between our work and CBS# is that in the latter, in order to determine the recipients of a transmission, a graph representation of node localities is employed. While this approach is more flexible, ours (based on locations and radius that define transmission cells and distances) enables a more compact representation of connectivity.

Finally, the calculus for Mobile Broadcasting Systems, MBS [Pra05] is currently being developed. MBS aims at providing a communication model which implements the "globally asynchronous, locally synchronous" communication model which is proper of wireless communication systems. Channels are employed as sealed rooms, preventing a message sent within a room to being captured by processes in other rooms. As in CBS#, interferences between two simultaneous transmissions is not taken into account.

P	$\overset{\text{def}}{=}$	$\mathtt{out}\langle e\rangle.P$	*output*
	$\mid$	$\langle v\rangle.P$	*active output*
	$\mid$	$\mathtt{in}(x).P$	*input*
	$\mid$	$(x).P$	*active input*
	$\mid$	$\mathbf{0}$	*inactive process*
N	$\overset{\text{def}}{=}$	$n\,[P]^{c}_{l,r}$	*node (or device)*
	$\mid$	$N \vert N$	*parallel composition*
	$\mid$	$\mathbf{0}$	*empty network*

Table 1
Language for the description of wireless networks

2 The Language

We present the (core of the) Calculus of Wireless Systems, CWS. The language has two parts: the process part, which describes the possible state transitions inside a node, and the network part. For now, we introduce only those primitives that are necessary for communication. Some extensions are presented in Section 6.

Processes are sequential and live within the nodes, which are the basic network elements, and represent single devices. Each node has a location and a transmission radius; they are employed to define the cell over which that node can transmit. Nodes cannot be created or destroyed. We write $n\,[P]^{c}_{l,r}$ for a node named n, executing process P, synchronized on channel c, and which can transmit over a cell centered in l (i.e., its location) with radius r. The node identifier n represents a logical location – the device network addresses. By contrast, location l represents a physical location and, together with the radius, is employed for deriving information about the network connectivity.

We do not indicate how locations should be specified; for instance, they could be given by means of a coordinate system. The only assumption we make is the possibility of comparing locations, so to determine whether a node lies or not within the transmission cell of another node. We do so by means of a function d which takes two locations and returns their distance.

Nodes transmit values by broadcasting them via channels. The details about how transmissions work are given in the next section. Two important features of transmissions that are visible in the choice of operators in CWS are the following:

- As motivated in Section 1, in order to identify whether (and where) two transmissions interfere with each other, in CWS a transmission is not atomic, but is modelled by its two boundary events: *begin-transmission* and *end-transmission*. Physically, these events represent a modification of the antenna communication mode (from reception to transmission mode, and opposite).

- In a wireless system, a transmission is performed in a bit-per-bit fashion. If the transmission aborts, the partial bit-stream is discarded. Abstractly, this corresponds to keeping the channel allocated for the transmission time and revealing the value just before concluding the transmission. We hence assume that a transmitted value becomes visible only once it is completely transmitted; that is, we make it visible to the recipients with the occurrence of the end-transmission event.

Table 1 defines the syntax of CWS. We use $a..d$ for channels; $m..o$ for identifiers; $x..z$ for variables; $u..v$ for the values that can be transmitted in a communication over a channel; these include node identifiers, but not channels. Finally, $e..f$ are value expressions, which include values and variables; in the operational semantics, we use also the special symbol $\bot$, which indicates detection of an interference and cannot be transmitted. We do not provide a grammar for value and boolean expressions. They represent orthogonal choices with respect to the process and network constructs. We assume the existence of an evaluation function $[\![.]\!]$, which returns the value of a closed expression.

In the process constructs, an output $\mathtt{out}\langle e \rangle.P$ is a process willing to initiate the transmission of value $v = [\![e]\!]$; as a result, the process will evolve to $\langle v \rangle.P$. This represents a process that is currently transmitting v; when the transmission is terminated, this process becomes P. An input process $\mathtt{in}(x).P$ represents a process that is willing to receive; when the begin of a transmission is detected in clear the process becomes the active input process $(x).P$, representing a process that is currently receiving. This reception succeeds if no collision is detected until the end of the transmission; in this case, the received value is bound to x. Otherwise $\bot$, which indicates a failed reception, is bound to x. The inactive process represents a terminated process. In the syntax for networks, parallel composition represents two networks that execute in parallel. The empty network is a network without any node. In $\mathtt{in}(x).P$ and $(x).P$ the displayed occurrence of x is binding with scope P.

We assume that in any network each node identifier is unique, and two different nodes cannot share the same location. Further, no active input or active output can appear underneath a process prefix. We call such networks *well-formed*; in the remainder of the paper all networks are well formed. Since new nodes cannot be created, the well-formedness of a network is maintained as the network evolves.

For power saving purposes, when a device is not performing any transmission, it maintains the antenna in reception mode. Hence, in the reminder of this work we call *transmitter* a node whose process begins with an output or with an active output prefix; in the latter case, we say that the node is an *active transmitter* because it is presently performing a transmission (in practice, this is the only case in which a wireless device employs its antenna in transmit mode).

3 Reduction Semantics

We present in this section a Reduction Semantics (RS) for CWS. In the literature, RS has been mainly employed for describing processes with handshake communications (essentially mobile and higher calculi such as the π-calculus). In such systems, a reduction is an interaction between an emitter and a receiver. In our case, an internal activity is a broadcast which, as discussed in Sections 2 and 1, is modelled by the two events begin-transmission and end-transmission. Thus, in our RS for core CWS, a reduction represents a begin-transmission or an end-transmission.

To understand the formalisation of reduction, we have to bear in mind those communication aspects that are specific of wireless communications, as discussed in Section 1, notably:

C1: a node cannot freely initiate a transmission, but can only do so if it is not presently reached by a transmission over the same channel from another node;

C2: a transmission cannot reach arbitrary modes, but may affect all and only the nodes in the transmission cell of the transmitter;

C3: a transmission that is received by a node is not always readable, but it is so only if no other transmission reaches that node at that time.

A reduction is derived in three steps, each of which is formalised by means of specific relations and rules. For writing such rules, we extend the network terms of Table 1; we call *tagged terms* the extended terms and we use E to range over them:

$$E \overset{\text{def}}{=} N \quad \Big| \quad n\,[P]^{c}_{l,r}\,\{N\} \quad \Big| \quad \langle\!| N |\!\rangle \quad \Big| \quad \langle\!| N |\!\rangle_v \quad \Big| \quad E|E$$

where N is as by Table 1. In a term $n\,[P]^{c}_{l,r}\,\{N\}$, node $n\,[P]^{c}_{l,r}$ will always be a transmitter, and $\{N\}$ will represent a bag of receivers that falls within the cell of the transmitter. Terms $\langle\!| N |\!\rangle$ and $\langle\!| N |\!\rangle_v$ represent sets of receivers that have been tagged. We descrive the three steps used to derive a reduction:

(Event Selection): the transmitter node (i.e., the node that causes the reduction) is selected. In this step the transmitter is tagged by creating an empty bag for it (as for instance in $n\,[P]^{c}_{l,r}\,\{\mathbf{0}\}$), and condition C1 is checked, if necessary;

(Event Firing): the set of receivers that fall within the transmission cell of the node previously selected is isolated, by checking condition C2 and inserting them into the transmitter bag; then the transmission event is performed and, in doing so, the receivers are tagged.

(Normalization): for each receiver tagged in the previous step, condition C3 is checked so to determine its correct evolution and remove the tag; finally, the bag of the transmitter is dismissed.

Formally, a reduction step $N \longrightarrow N'$ occurs if one of the following holds:

(Begin Transmission) there are $c,\ l,\ r,\ E,\ E',\ l'$ such that
$N >^c_l E \longrightarrow^c_{l,r} E' \succ^* \ \succ^{c\,*}_{l'} \ \succeq N'$

(End Transmission) there are $c,\ l,\ r,\ E,\ E',\ l'$ such that
$N > E \longrightarrow^c_{l,r} E' \succ^* \ \succ^{c\,*}_{l'} \ \succeq N'$

where $\succ^*$ and $\succ^{c\,*}_{l'}$ are the transitive closures of $\succ$ and $\succ^c_{l'}$. The actual rules for the above relations are presented in Appendix A together with some notations that they use. The rules also use (standard) structural congruence, written $\equiv$ (the smallest congruence that makes networks, with $\mid$ and $\mathbf{0}$, an abelian monoid), and two extension of it. We recall that $d(l, l')$ gives the distance between locations l and l'.

The Event Selection step is formalised by means of relations $>^c_l$ (for a begin-transmission event) and $>$ (for an end-transmission). In rules RS-Comm$_{begin}$, and RS-Comm$_{end}$, the empty bag $\{\mathbf{0}\}$ marks the node selected for the transmission event. The choice of the selected node is then propagated through the system; in case of begin-transmission, condition C1 has to be checked (rule RS-Comp$_{begin}$).

The Event Firing step is formalised by relation $\longrightarrow^c_{l,r}$. The key rules are RS-Out$_{begin}$ and RS-Out$_{end}$, for begin- transmission and end-transmission events, respectively. In both cases, the transmission does not have an effect on the receivers in the bag, which are simply tagged (for an end-transmission, the tag includes the value that has been communicated in the transmission). Rule RS-Comp$_{event}$, for parallel composition, checks that the rest of the system does not contain receivers which should have been placed in the transmitter bag. This check guarantees that the bags employed in rules RS-Out$_{begin}$ and RS-Out$_{end}$ are indeed complete. In rule RS-SC-TR, the two extended structural congruence are used: the first, $\equiv_T$, allows us to bring receivers in the bag; the second, $\equiv_R$, allows us to decompose a tagged set of receivers into its primitive components.

Finally, the Normalization step is formalised by relations $\succ$, $\succ^c_l$, and $\succeq$. Relations $\succ$ and $\succ^c_l$ are used for the same reason, namely removing the tags in the receivers produced in the previous step. The reason for having two relations is that in the case of $\succ^c_l$ a check on l and c might need to be performed. Relation $\succ$ is employed to in three cases. The first two regard receivers that are reached by a begin-transmission event (i.e., receivers tagged $\llparenthesis - \rrparenthesis$). Such event has no effect if the receiver is not active and is presently reached by another transmission (rule RS-Ready$_{noise}$). By contrast, the event produces interference if the receiver is active, i.e., it is already listening to another transmission (rule RS-Reading$_{begin}$). The structural congruence of rule RS-SC$_{norm}$ allows us to obtain a conflict in rule RS-Ready$_{noise}$. The last case regards receivers that are reached by an end-transmission event (i.e., receivers tagged $\llparenthesis - \rrparenthesis_v$): such event has no effect if the receiver is not active, and therefore is waiting for a begin-transmission (rule RS-Ready$_{end}$).

Relation $\succ_l^c$ is employed in two cases: an input process becoming an active receiver (rule RS-Ready$_{begin}$), and an active receiver successfully terminating its reception (rule RS-Reading$_{end}$). The absence of interferences is checked in condition $E_2 \not\downarrow (l, c)$ of rule RS-Comp$_{labnorm}$: the location with the tagged receiver should not be reached by the transmission coming from an untagged active transmitter (that is, a transmitter different from the one that produced the tag). Finally, the relation $\succeq$ is employed to close the bag of the transmitter and to enable further reductions.

Note that normalisation must be applied to all tagged receivers in the systems, using relation $\succ$ or $\succ_l^c$; otherwise, the final term still contains tags and therefore does not belong to the grammar of Table 1.

Example: a communication

This example shows a communication between a transmitter $S \overset{\text{def}}{=} n\,[\mathtt{out}\langle v\rangle.\mathbf{0}]^c_{l,\rho}$ and a receiver $R \overset{\text{def}}{=} m\,[\mathtt{in}(x).P]^c_{l',\rho'}$ that is placed within the transmitter cell (i.e., $d(l, l') \leq \rho$).

$$\begin{array}{rcl}
n\,[\mathtt{out}\langle v\rangle.\mathbf{0}]^c_{l,\rho} \big|\, m\,[\mathtt{in}(x).P]^c_{l',\rho'} & >^c_l & n\,[\mathtt{out}\langle v\rangle.\mathbf{0}]^c_{l,\rho} \left\{ m\,[\mathtt{in}(x).P]^c_{l',\rho'} \right\} \\
 & \longrightarrow^c_{l,\rho} & n\,[\langle v\rangle.\mathbf{0}]^c_{l,\rho} \{\mathbf{0}\} \big|\, \llparenthesis m\,[\mathtt{in}(x).P]^c_{l',\rho'} \rrparenthesis \\
 & \succ^c_{l'} & n\,[\langle v\rangle.\mathbf{0}]^c_{l,\rho} \{\mathbf{0}\} \big|\, m\,[(x).P]^c_{l',\rho'} \\
 & \succeq & n\,[\langle v\rangle.\mathbf{0}]^c_{l,\rho} \big|\, m\,[(x).P]^c_{l',\rho'} \\
 & > & n\,[\langle v\rangle.\mathbf{0}]^c_{l,\rho} \left\{ m\,[(x).P]^c_{l',\rho'} \right\} \\
 & \longrightarrow^c_{l,\rho} & n\,[\mathbf{0}]^c_{l,\rho} \{\mathbf{0}\} \big|\, \llparenthesis m\,[(x).P]^c_{l',\rho'} \rrparenthesis_v \\
 & \succ^c_{l'} & n\,[\mathbf{0}]^c_{l,\rho} \{\mathbf{0}\} \big|\, m\,[\{v/x\}P]^c_{l',\rho'} \\
 & \succeq & n\,[\mathbf{0}]^c_{l,\rho} \big|\, m\,[\{v/x\}P]^c_{l',\rho'}
\end{array}$$

At the beginning of the execution, the only possible transition for the network is produced by the transmitter S, which initiates a transmission since there are no active transmitters in the network which interfere; thus the transmission bag for S is created and the receivers, R, are put within the bag (rules RS-StructCong and RS-BagReady). Then, the transmission is performed (rule RS-Out$_{begin}$) and R is tagged with a begin-transmission tag. Finally, the receiver is normalized (rule RS-Ready$_{begin}$) and the transmission bag closed. The end-transmission happens in a similar manner and R obtains the value transmitted by S.

Example: an interference

This example shows the behaviour of a receiver $R \stackrel{\text{def}}{=} m\,[\mathtt{in}(x).P]^c_{l',\rho'}$ and two transmitters $S_1 \stackrel{\text{def}}{=} n_1\,[\mathtt{out}\langle v_1\rangle.\mathbf{0}]^c_{l_1,\rho}$ and $S_2 \stackrel{\text{def}}{=} n_2\,[\mathtt{out}\langle v_2\rangle.\mathbf{0}]^c_{l_2,\rho}$ when transmissions interfere with each other at the receiver site (i.e., the relations $d(l_1,l_2) \geq \rho$, $d(l_1,l') \leq \rho$, and $d(l_2,l') \leq \rho$ hold). We have:

$$
\begin{array}{rcl}
N & \stackrel{\text{def}}{=} & n_1\,[\mathtt{out}\langle v_1\rangle.\mathbf{0}]^c_{l_1,\rho} \Big| \, n_2\,[\mathtt{out}\langle v_2\rangle.\mathbf{0}]^c_{l_2,\rho} \Big| \, m\,[\mathtt{in}(x).P]^c_{l',\rho'} \\
& >^c_{l_2} & n_2\,[\mathtt{out}\langle v_2\rangle.\mathbf{0}]^c_{l_2,\rho} \Big\{ m\,[\mathtt{in}(x).P]^c_{l',\rho'} \Big\} \Big| \, n_1\,[\mathtt{out}\langle v_1\rangle.\mathbf{0}]^c_{l_1,\rho} \\
& \longrightarrow^c_{l_2,\rho} & n_2\,[\langle v_2\rangle.\mathbf{0}]^c_{l_2,\rho}\,\{\mathbf{0}\} \Big| \, \langle\!| m\,[\mathtt{in}(x).P]^c_{l',\rho'} \,|\!\rangle \Big| \, n_1\,[\mathtt{out}\langle v_1\rangle.\mathbf{0}]^c_{l_1,\rho} \\
& \succ^c_{l'} & n_2\,[\langle v_2\rangle.\mathbf{0}]^c_{l_2,\rho}\,\{\mathbf{0}\} \Big| \, m\,[(x).P]^c_{l',\rho'} \Big| \, n_1\,[\mathtt{out}\langle v_1\rangle.\mathbf{0}]^c_{l_1,\rho} \\
& \succeq & n_2\,[\langle v_2\rangle.\mathbf{0}]^c_{l_2,\rho} \Big| \, m\,[(x).P]^c_{l',\rho'} \Big| \, n_1\,[\mathtt{out}\langle v_1\rangle.\mathbf{0}]^c_{l_1,\rho} \\
& >^c_{l_1} & n_1\,[\mathtt{out}\langle v_1\rangle.\mathbf{0}]^c_{l_1,\rho} \Big\{ m\,[(x).P]^c_{l',\rho'} \Big\} \Big| \, n_2\,[\langle v_2\rangle.\mathbf{0}]^c_{l_2,\rho} \\
& \longrightarrow^c_{l_1,\rho} & n_1\,[\langle v_1\rangle.\mathbf{0}]^c_{l_1,\rho}\,\{\mathbf{0}\} \Big| \, \langle\!| m\,[(x).P]^c_{l',\rho'} \,|\!\rangle \Big| \, n_2\,[\langle v_2\rangle.\mathbf{0}]^c_{l_2,\rho} \\
& \succ & n_1\,[\langle v_1\rangle.\mathbf{0}]^c_{l_1,\rho}\,\{\mathbf{0}\} \Big| \, m\,[\{\bot/x\}P]^c_{l',\rho'} \Big| \, n_2\,[\langle v_2\rangle.\mathbf{0}]^c_{l_2,\rho} \\
& \succeq & n_1\,[\langle v_1\rangle.\mathbf{0}]^c_{l_1,\rho} \Big| \, m\,[\{\bot/x\}P]^c_{l',\rho'} \Big| \, n_2\,[\langle v_2\rangle.\mathbf{0}]^c_{l_2,\rho} \\
& > & n_2\,[\langle v_2\rangle.\mathbf{0}]^c_{l_2,\rho}\,\{\mathbf{0}\} \Big| \, m\,[\{\bot/x\}P]^c_{l',\rho'} \Big| \, n_1\,[\langle v_1\rangle.\mathbf{0}]^c_{l_1,\rho} \\
& \longrightarrow^c_{l_2,\rho} & n_2\,[\mathbf{0}]^c_{l_2,\rho}\,\{\mathbf{0}\} \Big| \, m\,[\{\bot/x\}P]^c_{l',\rho'} \Big| \, n_1\,[\langle v_1\rangle.\mathbf{0}]^c_{l_1,\rho} \\
& \succeq & n_2\,[\mathbf{0}]^c_{l_2,\rho} \Big| \, m\,[\{\bot/x\}P]^c_{l',\rho'} \Big| \, n_1\,[\langle v_1\rangle.\mathbf{0}]^c_{l_1,\rho} \\
& > & n_1\,[\langle v_1\rangle.\mathbf{0}]^c_{l_1,\rho}\,\{\mathbf{0}\} \Big| \, m\,[\{\bot/x\}P]^c_{l',\rho'} \Big| \, n_2\,[\mathbf{0}]^c_{l_2,\rho} \\
& \longrightarrow^c_{l_1,\rho} & n_1\,[\mathbf{0}]^c_{l_1,\rho}\,\{\mathbf{0}\} \Big| \, m\,[\{\bot/x\}P]^c_{l',\rho'} \Big| \, n_2\,[\mathbf{0}]^c_{l_2,\rho} \\
& \succeq & n_1\,[\mathbf{0}]^c_{l_1,\rho} \Big| \, m\,[\{\bot/x\}P]^c_{l',\rho'} \Big| \, n_2\,[\mathbf{0}]^c_{l_2,\rho}
\end{array}
$$

At the beginning, both transmitters can begin a transmission. We assume that S_2 initiates its transmission; consequently, the receiver R detects the transmission and begins receiving. Then, transmitter S_1, which cannot detect the transmission from T_2, initiates its own, provoking an interference on R; this interference is detected by R, which stores $\bot$ and then continues its execution.

4 Labelled Transition Semantics

Reflecting the structure of the language syntax, the Labelled Transition Semantics (LTS) has two sets of rules: those for the processes and those for the networks. Table 2 presents the LTS for processes. Transitions are of the form $P \xrightarrow{\alpha} P$, where the grammar for α is

$$\alpha ::= ! \;\Big|\; ? \;\Big|\; !v \;\Big|\; ?v$$

Labels ! and $!v$ represent begin-transmission and end-transmission events emanating from P; and ?, $?v$ represent begin-transmission and end-transmission events that reach P. We comment the rules. In PS-OUT$_{begin}$, the process evaluates expression e and initiates the transmission of the resulting value; in PS-OUT$_{end}$, the process terminates transmission of v; in PS-IN$_{begin}$, the process captures a begin-transmission; in PS-WAIT$_{begin}$, the input process does not react to an end-transmission event because it is waiting for a begin-transmission event; in PS-IN$_{end1}$, the process receives v, and then continues its execution; in PS-IN$_{end2}$, the process receives noise (i.e., there is interference); rule PS-NOTIN shows that non-input processes never react to reception of events.

$$\frac{[\![e]\!] = v}{\mathtt{out}\langle e\rangle.P \xrightarrow{!} \langle v\rangle.P}\ [\text{PS-OUT}_{begin}] \qquad \langle v\rangle.P \xrightarrow{!v} P\ \ [\text{PS-OUT}_{end}]$$

$$\mathtt{in}(x).P \xrightarrow{?} (x).P\ \ [\text{PS-IN}_{begin}] \qquad \mathtt{in}(x).P \xrightarrow{?v} \mathtt{in}(x).P\ \ [\text{PS-WAIT}_{begin}]$$

$$(x).P \xrightarrow{?v} \{v/x\}P\ \ [\text{PS-IN1}_{end}]$$

$$(x).P \xrightarrow{?\perp} \{\perp/x\}P\ \ [\text{PS-IN2}_{end}]$$

$$\frac{\alpha \in \{?, ?v\} \quad P \notin IP}{P \xrightarrow{\alpha} P}\ [\text{PS-NOIN}]$$

where IP is the set of processes of the form $\mathtt{in}(x).P_1$ or $(x).P_1$

Table 2
Labelled transitions for processes

In the network semantics, a transition has the form

$$T \rhd N \xrightarrow{\mu} N'$$

and reads *given the set T of active transmitters, at the occurrence of event μ, network N becomes network N'*. Each active transmitter in T is represented by a triple (l, r, c) which represent, respectively, the transmitter location, radius, and communication channel. The set T and the set of active transmitters in N need not coincide; this can be useful, for instance to analyse the evolution of a system in presence of external transmissions. The two sets should however coincide to know the evolution of a system in isolation. The semantics does not automatically updates the set T after a transition. Thus, when a transition is performed, the new T to be used for the next transition has to be manually computed. We have preferred to keep in the rules only the information necessary for deriving the transitions; it is easy to modify the rules so that they also produce the new T.

There are four possible kinds of transitions :

(i) $T \triangleright N \xrightarrow{c![l,r]} N'$: the node in N at location l becomes active and initiates a transmission over channel c with radius r;

(ii) $T \triangleright N \xrightarrow{c!v[l,r]} N'$: analogous to the previous one, for an end transmission;

(iii) $T \triangleright N \xrightarrow{c?[l,r]} N'$: the begin of a transmission over channel c and with radius r, produced by a node that is external to N and located at l, is propagated into N;

(iv) $T \triangleright N \xrightarrow{c?v[l,r]} N'$: analogous to the previous one, for an end transmission event.

Accordingly, the following well-formedness condition is imposed: in transitions of the form (1) or (3), that is, begin-transmissions, (l, r, c) should not be in T; conversely, for transitions of the form (2) or (4), that is, end-transmissions, (l, r, c) must be in T.

Table 3 presents the LTS for networks. In the rules, the metavariable θ is a notational convention used to have fewer rules. Thus $?\theta$ ranges over $?$ and $?v$; similarly, $!\theta$ ranges over $!$ and $!v$. Also, we write $T|_{c,l}$ for the subset of the active transmitters T whose transmission reach a node located in l and synchronized on channel c. Formally,

$$T|_{c,l} \; = \; \{(l', r', c') \mid d(l', l) \leq r' \wedge c' = c\}$$

In the LTS for networks, the key rules regard events reaching a single node (i.e., NS-In$_1$, NS-In$_2$, and NS-In$_3$) and nodes emitting events (i.e., NS-Out$_{begin}$ and NS-Out$_{end}$). Rule NS-In$_1$ shows the behaviour of a node that receives an event in clear. Condition $d(l, l') \leq r'$ ensures that the node receives the event in the label, whereas $T|_{l,c} \subseteq \{l', r', c\}$ ensures that the node is not exposed to other transmissions. Rule NS-In$_2$ shows that a node is not affected by events that do not reach the node. Rule NS-In$_3$ describes a node that detects colliding transmissions; conditions $d(l, l') \leq r'$ and $T|_{l,c} \nsubseteq \{l', r', c\}$ ensure that at least two transmissions reach the node. Rule NS-Out$_{begin}$ shows that a transmission can be initiated by a node only when that node is not presently reached by another transmission. In rule NS-Out$_{end}$, the

$$\frac{P \xrightarrow{!} P' \quad T|_{l,c} = \emptyset}{T \rhd n\,[P]^c_{l,r} \xrightarrow{c![l,r]} n\,[P']^c_{l,r}}\ [\text{NS-}\textsc{Out}_{begin}]$$

$$\frac{P \xrightarrow{!v} P'}{T \rhd n\,[P]^c_{l,r} \xrightarrow{c!v[l,r]} n\,[P']^c_{l,r}}\ [\text{NS-}\textsc{Out}_{end}]$$

$$\frac{P \xrightarrow{?\theta} P' \quad d(l,l') \leq r' \quad T|_{l,c} \subseteq \{l',r',c\}}{T \rhd n\,[P]^c_{l,r} \xrightarrow{c?\theta[l',r']} n\,[P']^c_{l,r}}\ [\text{NS-}\textsc{In}_1]$$

$$\frac{d(l,l') > r' \vee c \neq c'}{T \rhd n\,[P]^c_{l,r} \xrightarrow{c'?\theta[l',r']} n\,[P]^c_{l,r}}\ [\text{NS-}\textsc{In}_2]$$

$$\frac{P \xrightarrow{?\perp} P' \quad d(l,l') \leq r' \quad T|_{l,c} \not\subseteq \{l',r',c\}}{T \rhd n\,[P]^c_{l,r} \xrightarrow{c?\theta[l',r']} n\,[P']^c_{l,r}}\ [\text{NS-}\textsc{In}_3]$$

$$\frac{T \rhd N_1 \xrightarrow{c?\theta[l,r]} N_1' \quad T \rhd N_2 \xrightarrow{c!\theta[l,r]} N_2'}{T \rhd N_1|N_2 \xrightarrow{c!\theta[l,r]} N_1'|N_2'}\ [\text{NS-}\textsc{Com}_r]$$

$$\frac{T \rhd N_1 \xrightarrow{c!\theta[l,r]} N_1' \quad T \rhd N_2 \xrightarrow{c?\theta[l,r]} N_2'}{T \rhd N_1|N_2 \xrightarrow{c!\theta[l,r]} N_1'|N_2'}\ [\text{NS-}\textsc{Com}_l]$$

$$\frac{T \rhd N_1 \xrightarrow{c?\theta[l,r]} N_1' \quad T \rhd N_2 \xrightarrow{c?\theta[l,r]} N_2'}{T \rhd N_1|N_2 \xrightarrow{c?\theta[l,r]} N_1'|N_2'}\ [\text{NS-R}\textsc{Com}_{in}]$$

Table 3
Labelled transitions for networks

transmission of a node terminates.

The propagation of events through the network is described by rules NS-$\textsc{Com}_r$, NS-$\textsc{Com}_l$, and NS-$\textsc{Com}_{in}$. In NS-$\textsc{Com}_r$ (and NS-$\textsc{Com}_l$), an event generated in a network is propagated to another network that executes in parallel (NS-$\textsc{Com}_l$ is symmetrical to NS-$\textsc{Com}_r$). Finally, NS-$\textsc{Com}_{in}$ shows two networks in parallel that detect the same event.

Examples

We redo the examples at the end of Section 3 using the LTS. We do not repeat the descriptions of the systems. First the communication example. Let

$$N \stackrel{\text{def}}{=} n\,[\mathtt{out}\langle v\rangle.\mathbf{0}]^{c}_{l,\rho}\Big|\, m\,[\mathtt{in}(x).P]^{c}_{l',\rho'}$$

We have:

$$\begin{aligned}
\emptyset \rhd N &\xrightarrow{c![l,\rho]} n\,[\langle v\rangle.\mathbf{0}]^{c}_{l,\rho}\Big|\, m\,[(x).P]^{c}_{l',\rho'} \stackrel{\text{def}}{=} N_1\\
\{(l,\rho,c)\} \rhd N_1 &\xrightarrow{c!v[l,\rho]} n\,[\mathbf{0}]^{c}_{l,\rho}\Big|\, m\,[\{v/x\}P]^{c}_{l',\rho'}
\end{aligned}$$

The first event is the begin-transmission from the transmitter S; this can happen since there are no active transmissions reaching S (i.e., $\emptyset|_{l,c} = \emptyset$). The receiver R, which was not previously reached by any transmission (i.e., $\emptyset|_{l',c} = \emptyset$) sees the transmission event and begins receiving that transmission. In the second event, the triple (l, r, c), representing S, is included in the set of active transmissions. As pointed out earlier, with the present semantics the set of active transmitters has to be computed manually at each transition. In the final step, the transmitter completes the transmission of value v and R terminates the reception.

Now the interference example. Let

$$N \stackrel{\text{def}}{=} n_1\,[\mathtt{out}\langle v_1\rangle.\mathbf{0}]^{c}_{l_1,\rho}\Big|\, n_2\,[\mathtt{out}\langle v_2\rangle.\mathbf{0}]^{c}_{l_2,\rho}\Big|\, m\,[\mathtt{in}(x).P]^{c}_{l',\rho'}\,.$$

We have:

$$\begin{aligned}
\emptyset \rhd N &\xrightarrow{c![l_2,\rho]} n_1\,[\mathtt{out}\langle v_1\rangle.\mathbf{0}]^{c}_{l_1,\rho}\Big|\, n_2\,[\langle v_2\rangle.\mathbf{0}]^{c}_{l_2,\rho}\Big|\, m\,[(x).P]^{c}_{l',\rho'}\\
&\quad \stackrel{\text{def}}{=} N_1\\
\{(l_2,\rho,c)\} \rhd N_1 &\xrightarrow{c![l_2,\rho]} n_1\,[\langle v_1\rangle.\mathbf{0}]^{c}_{l_1,\rho}\Big|\, n_2\,[\langle v_2\rangle.\mathbf{0}]^{c}_{l_2,\rho}\Big|\, m\,[\{\bot/x\}P]^{c}_{l',\rho'}\\
&\quad \stackrel{\text{def}}{=} N_2\\
\{(l_2,\rho,c),(l_1,\rho,c)\} \rhd N_2 &\xrightarrow{c!v_2[l_2,\rho]} n_1\,[\langle v_1\rangle.\mathbf{0}]^{c}_{l_1,\rho}\Big|\, n_2\,[\mathbf{0}]^{c}_{l_2,\rho}\Big|\, m\,[\{\bot/x\}P]^{c}_{l',\rho'}\\
&\quad \stackrel{\text{def}}{=} N_3\\
\{(l_1,\rho,c)\} \rhd N_3 &\xrightarrow{c!v_1[l_1,\rho]} n_1\,[\mathbf{0}]^{c}_{l_1,\rho}\Big|\, n_2\,[\mathbf{0}]^{c}_{l_2,\rho}\Big|\, m\,[\{\bot/x\}P]^{c}_{l',\rho'}
\end{aligned}$$

First, S_2 initiates its transmission; R detects the transmission and begin receiving. Then, S_1, which cannot detect S_2's transmission (since it holds that $\{(l_2,\rho,c)\}|_{l,c} = \emptyset$) initiates its own transmission, causing an interference at R. As a consequence, R records $\bot$ as a result.

Note that for the above interference to occur, condition $d(l_1, l_2) \geq \rho$ is necessary. A process begins transmitting only if its sees the communication channel free.

5 The Harmony Theorem

The Harmony Theorem proves the correspondence between the two semantics. The theorem has two parts. The first shows that the LTS is compatible with structural congruence; that is, application of structural congruence does not affect the possible transitions. The second shows that the RS behaves as the LTS; i.e., each reduction in the RS has a corresponding transition in the LTS and vice versa, and the resulting networks are structurally congruent.

Theorem 5.1 (Harmony Theorem) *Let N be network, and T the set of active transmitters in N.*

(i) *If $T \triangleright N \xrightarrow{\mu} N'$ and $N \equiv N_1$ then also $T \triangleright N_1 \xrightarrow{\mu} N_1' \equiv N'$.*

(ii) *If $N \longrightarrow N'$ then either*

  (a) *there are l, r, c such that $T \triangleright N \xrightarrow{c![l,r]}\equiv N'$;*

  (b) *or there are l, r, c and v such that $T \triangleright N \xrightarrow{c!v[l,r]}\equiv N'$.*

(iii) *The converse, i.e., if $T \triangleright N \xrightarrow{c![l,r]} N'$, for some l, r, c, or $T \triangleright N \xrightarrow{c!v[l,r]}\equiv N'$, for some l, r, c, v, then also $N \longrightarrow\equiv N'$.*

The structure of the proof of the theorem is the same as as that of analogous results for point-to-point message-passing process calculi [Mil91,SW01]. The details, however, are quite different, because of the peculiarities of our LTS and RS (and the use of tagged terms and six auxiliary relations in the RS). We first prove a few auxiliary lemmas for the RS that describe how the structure of a network evolves across the reduction sub-steps, when a transmission begins and when a transmission ends.

6 Language Extensions

In the previous sections we have considered the subset of CWS with only the operators necessary for communication. We now briefly discuss the full calculus. While the new extensions add a considerable amount of details to the rules for the RS and LTS and to the Harmony Theorem, they do not introduce new technical difficulties. Because of this, and of lack of space, we only give an informal account of them.

The first addition is both an extension and a modification of the core calculus: the input construct is replaced by the *handled input* construct $\mathtt{in}(x.)P\lceil Q$. This process can either receive a transmission, or trigger the handler Q. In the first case, when the reception begins the process becomes an active receiver $(x).P$. In the second case, the process evolves to process Q. The handled input enables the modelling of scenarios in which a device, due to the verification of internal conditions (e.g., timeouts, interrupts, exceptions) stops waiting for incoming messages. This situation is common in modern wireless systems (for instance, in communication protocols that implement techniques for preventing starvation).

The other additions to the process syntax are: a construct `switch` $c.P$, representing a process that decides to change its communication channel, synchronizing on channel c, and then continues as P; the construct `if` e `then` P_1 `else` P_2, representing a process that behaves according to the truth value $[\![e]\!]$; recursion, to model devices with an infinite behaviour. Finally, the network language is augmented with the restriction construct $(\nu c)\ N$, that creates a new channel c with scope N; the new channel can be used only by the nodes in N.

A consequence of these additions in the operational semantics is the introduction of transitions representing activities internal to the nodes (the analogous of the τ-steps in languages such as CCS), both in the RS and in the LTS. Examples of internal activities are the switch of a communication channel and the activation of an input handler.

7 Conclusions and Future Work

In this paper we have developed an operational semantics for a calculus of wireless systems. We have presented a RS and a LTS and proved a correspondence result between them. We have first presented a core calculus, essentially containing the primitives for communication, and we have then extended it to a more complete calculus.

Technical difficulties and the peculiarities of wireless systems lead us to separate, in both semantics, the begin and the end of a transmission. By contrast, the two semantics follow a different approach for the check of interferences: only the LTS takes a set of active transmitters as a parameter for the rules. It would have been possible to design the RS employing the same parameter as for the LTS, and the correspondence between the two semantics would have been easier to find. We have chosen not to do so in order to compare the two approaches; at present, we do not see one that is obviously better than the other.

In the definition of the operational semantics we encountered problems due to the physical peculiarities of wireless systems, discussed in Section 1. For this reason, we decided to maintain the calculus "small" and to focus on its operational semantics only.

A number of developments are possible. First, we need to evaluate the expressive power of the operators in CWS. Some examples, such as a version of the alternating bit protocol for wireless systems, are presented in [Mez06], but more experiments are needed. Second, one could try to define appropriate notions of behavioural equivalence and/or logics. Standard methods for defining these notions on top of an operational semantics appears difficult to pursue. For instance, a behavioural equivalence is usually defined with respect to an observer that interacts with the systems. In wireless systems, each device – and therefore presumably also the observer – has a locations and a transmission cell, and it is not clear how to take them into account.

In CWS, devices are static – they cannot move. Movement could be added as a separate feature, orthogonal to communication. Indeed, with present technologies the communication times are negligible with respect to times for mechanical movements. For instance, consider a transmitter that moves at 5 meters and that transmits a 2000-byte frame over a channel having a 2 megabit-per-second bandwidth. The actual transmission would take about 0.008 seconds; during that time, the transmitter moves only about 4 centimetres away. In other words, a viable approach to movement could consist in simply adding primitives for expressing movement of devices (the calculus could even be parametric with respect to to this); during communication, however, the network would be considered static, and therefore communication could be described using the machinery developed in this paper.

Another possible extension is to enable channel transmission and dynamic creation of new channels. It should be possible to formalise such an extension borrowing ideas from calculi for point-to-point mobile processes such as the π-calculus. Similarly one could proceed for the addition of primitives for cryptography.

One may also want to relax the assumption that nodes within a certain radius can always receive transmissions, because, for instance, blockage caused by obstacles, features of the antenna, etc..

Finally, it could be useful to add an explicit notion of time to the calculus. For instance, this would allow us a more precise formalization of the timeout handler for the input operation (Section 6), so to obtain a deterministic semantics for handler activation.

Acknowledges

We would like to thank Luciano Bononi for many interesting discussions, and the anonymous referees for some useful suggestions.

References

[AK04] Ian F. Akyildiz and Ismail H. Kasimoglu. Wireless sensor and actor networks: research challenges. *Ad Hoc Networks*, 2(4):351–367, 2004.

[ASSC02] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):10214, August 2002.

[BG92] Gerard Berry and Georges Gonthier. The esterel synchronous programming language: Design, semantics, implementation. *Science of Computer Programming*, 19(2):87–152, 1992.

[BHR84] S. D. Brookes, C. A. R. Hoare, and A. W. Roscoe. A theory of communicating sequential processes. *Journal of the Association for Computing Machinery*, 31(3):560–599, 1984.

[CDM+03] Ioannis Chatzigiannakis, Tassos Dimitriou, Marios Mavronicolas, Sotiris Nikoletseas, and Paul Spirakis. A comparative study of protocols for efficient data propagation in smart dust networks. *Parallel Processing Letters*, 13(4):615–627, 2003.

[Est01] Deborah Estrin. Wireless sensor networks: application driver for low power distributed systems. In *ISLPED '01: Proceedings of the 2001 international symposium on Low power electronics and design*, pages 194–194, New York, NY, USA, 2001. ACM Press.

[Goo04] B. G. Goode. Reliable enough for a nuclear plant? *Wireless for Industry*, 3:7–9, 2004.

[Har87] David Harel. Statecharts: A visual formalism for complex systems. *Science of Computer Programming*, 8(3):231–274, June 1987.

[IBK+05] Diane Ingraham, Rod Beresford, Kadambari Kaluri, Moïse Ndoh, and Kannan Srinivasan. Wireless sensors: Oyster habitat monitoring in the bras d'or lakes. In *Proceedings od the International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 399–400. Springer-Verlag, 2005.

[inc04] Microstrain inc. Wireless sensors monitor critical civil infrastructure. *Wireless for Industry*, 3:6, 2004.

[Kev05a] T. Kevan. Coming soon to your neighborhood. *Wireless Sensors*, 3:5–8, 2005.

[Kev05b] T. Kevan. Updating a steel mill wirelessly. *Wireless Sensors*, 4:4–5, 2005.

[MCP+02] Alan Mainwaring, David Culler, Joseph Polastre, Robert Szewczyk, and John Anderson. Wireless sensor networks for habitat monitoring. In *Proceedings of the First ACM International Workshop on Wireless Sensor Networks and Applications (WSNA-02)*, pages 88–97, New York, September 28 2002. ACM Press.

[Mez06] N. Mezzetti. *Trust and wireless communication.* PhD thesis, University of Bologna, 2006.

[Mil83] R. Milner. Calculi for synchrony and asynchrony. *Theoretical Computer Science*, 25(3):267–310, 1983.

[Mil89] R. Milner. *Communication and Concurrency.* Prentice Hall, 1989.

[Mil91] R. Milner. The polyadic π-calculus: a tutorial. Technical Report ECS–LFCS–91–180, LFCS, 1991. Also in *Logic and Algebra of Specification*, ed. F.L. Bauer, W. Brauer and H. Schwichtenberg, Springer Verlag, 1993.

[NH04] Sebastian Nanz and Chris Hankin. Formal security analysis for ad-hoc networks. In *Proceedings of the 2004 Workshop on Views on Designing Complex Architectures (VODCA'04)*, pages 195–213. Elsevier, 2004.

[NH05] Sebastian Nanz and Chris Hankin. A framework for security analysis of mobile wireless networks. Technical Report 2005/4, Imperial College London, 2005. `http://www.doc.ic.ac.uk/research/technicalreports/`.

[Nik06] Sotiris E. Nikoletseas. Models and algorithms for wireless sensor networks (smart dust). In *Proceedings of the 32nd Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM 2006)*, pages 64–83. Springer-Verlag, 2006.

[OPT02] Karol Ostrovsky, K. V. S. Prasad, and Walid Taha. Towards a primitive higher order calculus of broadcasting systems. In *Proceedings of the 4th International Conference on Principles and Practice of Declarative Programming*, pages 2–13. ACM Press, 2002.

[Pra95a] K. V. S. Prasad. A calculus of broadcasting systems. *Sci. Comput. Program.*, 25(2-3):285–327, 1995.

[Pra95b] K. V. S. Prasad. A calculus of broadcasting systems. *SCIPROG: Science of Computer Programming*, 25(2-3):285–327, 1995.

[Pra96] K. V. S. Prasad. Broadcasting in time. In Paolo Ciancarini and Chris Hankin, editors, *COORDINATION*, volume 1061 of *Lecture Notes in Computer Science*, pages 321–338. Springer, 1996.

[Pra05] K. V. S. Prasad. A prospectus for mobile broadcasting systems. In Luca Aceto and Andrew Gordon, editors, *Proceedings of the Workshop on Algebraic Process Calculi: The First Twenty Five Years and Beyond, (PA '05)*, pages 209–212. BRICS Press, June 2005.

[SNL05] D. Senders, S. Neely, and J. Lewis. Manufacturing probe needles with vision. *Wireless Sensors*, 3:25–28, 2005.

[SPMC04] Robert Szewczyk, Joseph Polastre, Alan M. Mainwaring, and David Culler. Lessons from a sensor network expedition. In *2nd European Workshop on Wireless Sensor Networks*, pages 307–322. ACM Press, 2004.

[SW01] D. Sangiorgi and D. Walker. *The π-calculus: a Theory of Mobile Processes.* Cambridge University Press, 2001.

[Sys04] Sensicast Systems. Sensicast mesh network monitors nuclear plant motors. *Wireless for Industry*, 3:6, 2004.

[WLLP01] Brett Warneke, Matt Last, Brian Liebowitz, and Kristofer S. J. Pister. Smart dust: Communicating with a cubic-millimeter computer. *Computer*, 34(1):44–51, 2001.

A Reduction rules

Notations

- $E \not\downarrow (l, c)$ holds if network E contains no active transmitters $n\,[\langle v\rangle.P]^{c}_{l',r}$, for which $d(l, l') \leq r$ holds (i.e., no transmissions on c emanating from E presently reach l).
- $(l, r, c) \not\Downarrow E$ holds if network E contains no receivers $n\,[\mathtt{in}(x).P]^{c}_{l',r'}$ or $n\,[(x).P]^{c}_{l',r'}$ for which $d(l, l') \leq r$ holds (i.e., a transmission on c from l with radius r presently reaches no receivers in E).

Transmitter structural congruence, $\equiv_T$ (add the rules below to those for $\equiv$)

$$\frac{d(l, l') \leq r}{n\,[P]^{c}_{l,r}\,\{N\}\,|m\,[\mathtt{in}(x).Q]^{c}_{l',r'} \equiv_T n\,[P]^{c}_{l,r}\left\{N|m\,[\mathtt{in}(x).Q]^{c}_{l',r'}\right\}}\ \text{[SC-BagReady]}$$

$$\frac{d(l, l') \leq r}{n\,[P]^{c}_{l,r}\,\{N\}\,|m\,[(x).Q]^{c}_{l',r'} \equiv_T n\,[P]^{c}_{l,r}\left\{N|m\,[(x).Q]^{c}_{l',r'}\right\}}\ \text{[SC-BagReading]}$$

Receiver structural congruence, $\equiv_R$ (add the rules below to those for $\equiv$)

$$\langle\!| N_1|N_2 |\!\rangle \equiv_R \langle\!| N_1 |\!\rangle | \langle\!| N_2 |\!\rangle \quad \text{[SC-begin]}$$

$$\langle\!| N_1|N_2 |\!\rangle_v \equiv_R \langle\!| N_1 |\!\rangle_v | \langle\!| N_2 |\!\rangle_v \quad \text{[SC-end]}$$

Rules for $>^{c}_{l}$

$$\frac{\llbracket e \rrbracket = v}{n\,[\mathtt{out}\langle e\rangle.P]^{c}_{l,r} >^{c}_{l} n\,[\mathtt{out}\langle v\rangle.P]^{c}_{l,r}\,\{\mathbf{0}\}}\ \text{[RS-Comm}_{begin}\text{]}$$

$$\frac{N_1 >^{c}_{l} E \quad N_2 \not\downarrow (l, c)}{N_1 \,|\, N_2 >^{c}_{l} E \,|\, N_2}\ \text{[RS-Comp}_{begin}\text{]}$$

Rules for $>$

$$n\,[\langle v\rangle.P]^{c}_{l,r} > n\,[\langle v\rangle.P]^{c}_{l,r}\,\{\mathbf{0}\} \quad \text{[RS-Comm}_{end}\text{]}$$

$$\frac{N_1 > E}{N_1 \,|\, N_2 > E \,|\, N_2}\ \text{[RS-Comp}_{end}\text{]}$$

Rules for $\longrightarrow^{c}_{l,r}$

$$n\,[\mathtt{out}\langle v\rangle.P]^{c}_{l,r}\,\{N\} \longrightarrow^{c}_{l,r} n\,[\langle v\rangle.P]^{c}_{l,r}\,\{\mathbf{0}\} \mid \llparenthesis N \rrparenthesis \quad [\text{RS-Out}_{begin}]$$

$$n\,[\langle v\rangle.P]^{c}_{l,r}\,\{N\} \longrightarrow^{c}_{l,r} n\,[P]^{c}_{l,r}\,\{\mathbf{0}\} \mid \llparenthesis N \rrparenthesis_{v} \quad [\text{RS-Out}_{end}]$$

$$\frac{E_1 \longrightarrow^{c}_{l,r} E_1' \quad (l,r,c) \not\Downarrow E_2}{E_1 \mid E_2 \longrightarrow^{c}_{l,r} E_1' \mid E_2} \quad [\text{RS-Comp}_{event}]$$

$$\frac{E \equiv_T E' \longrightarrow^{c}_{l,r} E'' \equiv_R E'''}{E \longrightarrow^{c}_{l,r} E'''} \quad [\text{RS-SC-TR}]$$

Rules for $\succ^{c}_{l}$

$$\llparenthesis n\,[\mathtt{in}(x).P]^{c}_{l,r} \rrparenthesis \succ^{c}_{l} n\,[(x).P]^{c}_{l,r} \quad [\text{RS-Ready}_{begin}]$$

$$\llparenthesis n\,[(x).P]^{c}_{l,r} \rrparenthesis_{v} \succ^{c}_{l} n\,[\{v/x\}P]^{c}_{l,r} \quad [\text{RS-Reading}_{end}]$$

$$\frac{E_1 \succ^{c}_{l} E_1' \quad E_2 \not\downarrow (l,c)}{E_1 \mid E_2 \succ^{c}_{l} E_1' \mid E_2} \quad [\text{RS-Comp}_{labnorm}]$$

$$\frac{E \equiv E' \succ^{c}_{l} E'' \equiv E'''}{E \succ^{c}_{l} E'''} \quad [\text{RS-SC}_{\succ^{c}_{l}}]$$

Rules for $\succ$

$$\frac{d(l,l_m) \leq r_m}{m\,[\langle v_m\rangle.P_m]^{c}_{l_m,r_m} \mid \llparenthesis n\,[\mathtt{in}(x).P]^{c}_{l,r} \rrparenthesis \succ m\,[\langle v_m\rangle.P_m]^{c}_{l_m,r_m} \mid n\,[\mathtt{in}(x).P]^{c}_{l,r}}$$

$$[\text{RS-Ready}_{noise}]$$

$$\llparenthesis n\,[(x).P]^{c}_{l,r} \rrparenthesis \succ n\,[\{\bot/x\}P]^{c}_{l,r} \quad [\text{RS-Reading}_{begin}]$$

$$\llparenthesis n\,[\mathtt{in}(x).P]^{c}_{l,r} \rrparenthesis_{v} \succ n\,[\mathtt{in}(x).P]^{c}_{l,r} \quad [\text{RS-Ready}_{end}]$$

$$\frac{E_1 \succ E_1'}{E_1 \mid E_2 \succ E_1' \mid E_2} \quad [\text{RS-Comp}_{norm}]$$

$$\frac{E \equiv E' \succ E'' \equiv E'''}{E \succ E'''} \quad [\text{RS-SC}_{norm}]$$

Rules for $\succeq$

$$n\,[P]^{c}_{l,r}\,\{\mathbf{0}\} \succeq n\,[P]^{c}_{l,r} \quad [\text{RS-}\textsc{Transm}]$$

$$\frac{E_1 \succeq N_1}{E_1 \mid N_2 \succeq N_1 \mid N_2} \quad [\text{RS-}\textsc{Comp}_{\succeq}]$$

$$\frac{E \equiv E' \succeq E'' \equiv E'''}{E \succeq E'''} \quad [\text{RS-SC}_{\succeq}]$$